Абай атындағы Қазақ ұлттық педагогикалық университеті

ӘӨЖ 378.016:004.89 Қолжазба құқығында

ЖАКСЫЛЫКОВ АСЕТ ЕГНАЕВИЧ

ЖОҒАРЫ ОҚУ ОРНЫ СТУДЕНТТЕРІНІҢ ПРОГРАММАЛАУДАН БІЛІМІН БАҚЫЛАУДЫҢ ИНТЕЛЛЕКТУАЛДЫҚ МОДЕЛІН ЖАСАУ ЖӘНЕ ІСКЕ АСЫРУ

6D011100 – Информатика

философия докторы (PhD)
алу үшін дайындалған диссертация

Отандық ғылыми кеңесші:
педагогика ғылымдарының докторы,
профессор Сагимбаева А.Е.

Шетелдік ғылыми кеңесші
педагогика ғылымдарының докторы,
профессор Гриншкун В.В.

Қазақстан Республикасы
Алматы 2026

МАЗМҰНЫ

НОРМАТИВТІК СІЛТЕМЕЛЕР..... 3
АНЫҚТАМАЛАР..... 4
БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР..... 6
КІРІСПЕ..... 7
1 ЖОҒАРЫ ОҚУ ОРНЫ СТУДЕНТТЕРІНІҢ ПРОГРАММАЛАУДАН БІЛІМІН ИНТЕЛЛЕКТУАЛДЫҚ МОДЕЛЬ НЕГІЗІНДЕ БАҚЫЛАУДЫҢ ТЕОРИЯЛЫҚ НЕГІЗДЕРІ..... 15
1.1 Программалаудан студенттердің білімін бақылаудың қазіргі жағдайы.... 15
1.2 Интеллектуалдық оқу құралдарын қолданып программалаудан білімді бақылаудың ерекшеліктері..... 29
1.3 Программалаудан студенттердің білімін интеллектуалдық бақылау қағидалары және бағалау критерийлері..... 40
1-тарау бойынша қорытынды..... 53
2 ПРОГРАММАЛАУДАН СТУДЕНТТЕРДІҢ БІЛІМІН БАҚЫЛАУДЫҢ ИНТЕЛЛЕКТУАЛДЫҚ МОДЕЛІН ЖАСАУ ЖӘНЕ ІСКЕ АСЫРУ..... 55
2.1 Программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі..... 55
2.2 Интеллектуалдық модель негізінде программалаудан студенттердің білімін бақылауды ұйымдастыру әдістемесі..... 65
2.3 Программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін эксперименттік тексеру және оның тиімділігін бағалау..... 97
2-тарау бойынша қорытынды..... 114
ҚОРЫТЫНДЫ..... 116
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ..... 118
ҚОСЫМША А – Студенттерге арналған сауалнама..... 129
ҚОСЫМША Ә – Оқытушыларға арналған сауалнама..... 131
ҚОСЫМША Б – Студенттерге, профессор-оқытушыларға және IT-сарапшыларға арналған сауалнама..... 133
ҚОСЫМША В – Сараптамалық-негіздеуші кезеңде жасалған жобалық программалау тапсырмалары..... 140
ҚОСЫМША Г – Эксперименттік тексеруде қолданылған программалау тапсырмалары..... 143
ҚОСЫМША Ғ – Эксперименттік тексеру барысында студенттердің программалау әрекеті туралы деректерді тіркеу..... 144

НОРМАТИВТІК СІЛТЕМЕЛЕР

1. «Білім туралы» Қазақстан Республикасының Заңы (27.07.2007 ж. № 319-III; 01.03.2025 ж.) https://adilet.zan.kz/kaz/docs/Z070000319

2. «Жоғары және жоғары оқу орнынан кейінгі білім берудің мемлекеттік жалпыға міндетті стандарттарын бекіту туралы», ҚР Ғылым және жоғары білім министрінің 2022 жылғы 20 шілдедегі №2 бұйрығы https://adilet.zan.kz/kaz/docs/V2200028916

3. «Жоғары және (немесе) жоғары оқу орнынан кейінгі білім беру ұйымдарында оқытудың кредиттік технологиясы бойынша оқу процесін ұйымдастыру қағидаларын бекіту туралы», ҚР БҒМ 2011 жылғы 20 сәуірдегі №152 бұйрығы https://adilet.zan.kz/kaz/docs/V2300033173

4. «Жоғары және (немесе) жоғары оқу орнынан кейінгі білім беру ұйымдары қызметінің үлгілік қағидаларын бекіту туралы», ҚР БҒМ 2018 жылғы 30 қазандағы №595 бұйрығы https://adilet.zan.kz/kaz/docs/V1800017657

5. Қазақстан Республикасы Үкіметінің қаулысы «Қазақстан Республикасында 2023-2029 жылдарға арналған жоғары білім мен ғылымды дамыту тұжырымдамасын бекіту туралы» (28.03.2023 ж. № 248). https://adilet.zan.kz/kaz/docs/P2300000248

6. ҚР «Жасанды интеллект туралы» Заңы, 2025 жылғы 17 қарашадағы №230-VIII ҚРЗ https://adilet.zan.kz/kaz/docs/Z2500000230

7. «Жасанды интеллектті дамытудың 2024–2029 жылдарға арналған тұжырымдамасын бекіту туралы», ҚР Үкіметінің 2024 жылғы 24 шілдедегі №592 қаулысы https://adilet.zan.kz/kaz/docs/P2400000592

АНЫҚТАМАЛАР

Диссертациялық жұмыста төмендегі терминдер мен анықтамалар қолданылады:

Автоматтандырылған бақылау – студенттің программалық шешімін алдын ала белгіленген тестілік жағдайлар, эталондық шешімдер және тексеру ережелері негізінде автоматты түрде тексеріп, оның дұрыстығын бағалауды және нәтижелер бойынша кері байланыс қалыптастыруды қамтамасыз ететін білімді бақылау түрі.

Ақпараттық белгі – студенттің программалау әрекеті барысында алынған бастапқы деректерді (программалық код, тестілеу нәтижелері, қателер, шешімнің түсіндірмесі және т.б.) талдау арқылы анықталатын, бағалау көрсеткіштерін қалыптастыруға негіз болатын және студент білімінің тікелей бағасы болып табылмайтын дерек.

Бағалау критерийі – студенттің оқу нәтижелерінің белгіленген талаптарға сәйкестігін анықтауға негіз болатын, бағаланатын білімнің, дағдының немесе әрекет нәтижесінің маңызды сапалық сипаттарын айқындайтын өлшем.

Бақылаудың валидтілігі – бақылау нәтижелері негізінде студенттің білімі мен дағдылары туралы жасалатын қорытындылардың оқу мақсаттарына, бағаланатын білім мазмұнына және белгіленген бағалау критерийлеріне сәйкестігі мен негізділігін сипаттайтын қасиет.

Генеративті жасанды интеллект – үлкен көлемдегі деректер негізінде оқытылған модельдерді пайдалана отырып, пайдаланушының сұранысына сәйкес мәтін, программалық код, кескін және басқа да жаңа мазмұн түрлерін генерациялауға қабілетті жасанды интеллект технологиясы.

Деңгейлік дескриптор – белгілі бір бағалау критерийі бойынша студенттің оқу жетістігінің деңгейін анықтайтын және оны басқа деңгейлерден ажыратуға мүмкіндік беретін нақты белгілердің сипаттамасы.

Динамикалық талдау – программаны іске қосу және тестілеу арқылы оның берілген кіріс деректер жағдайындағы жұмыс істеу ерекшеліктерін, алынған нәтижелерді және туындайтын қателерді анықтауға бағытталған программалық кодты талдау тәсілі.

Жіберілімдер тарихы – студенттің белгілі бір программалау тапсырмасы бойынша тексеру жүйесіне ұсынған программалық шешімдерінің, оларды тексеру нәтижелерінің және жіберілген уақыттарының хронологиялық ретпен сақталған, тапсырманы орындау үдерісін талдауға негіз болатын деректер жиынтығы.

Интеллектуалдық бақылау – студенттің программалау тапсырмасын орындау нәтижесі мен үдерісі туралы әртүрлі деректерді жинау, біріктіру және интеллектуалдық талдау негізінде оның білімін белгіленген критерийлер мен көрсеткіштер бойынша бағалауға, сондай-ақ оқытушының педагогикалық шешім қабылдауын қолдауға бағытталған бақылау үдерісі.

Бақылаудың интеллектуалдық моделі – педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысы арқылы бақылау мақсатын анықтаудан бастап деректерді жинау, талдау, бағалау, нәтижелерді интерпретациялау және кері байланыс беруге дейінгі бақылау үдерісінің бірізділігін сипаттайтын құрылымдық-функционалдық модель.

Интеллектуалдық оқу құралдары – жасанды интеллект технологиялары негізінде білім алушының оқу әрекетін талдауға, оқу қажеттіліктерін ескеруге және бейімделген кері байланыс пен оқу қолдауын ұсынуға мүмкіндік беретін цифрлық құралдар.

Код сапасы – программалық кодтың функционалдық дұрыстығынан бөлек, оның құрылымдылығын, оқылымдылығын, кодтау стилі мен атау беру ережелерінің сақталуын, артық қайталанудың болмауын және сүйемелдеуге қолайлылығын сипаттайтын бағалау критерийі.

Қателерді іздеу және түзету – программаның күтілетін нәтижеден ауытқуының себебін анықтау, қатенің орнын табу, оны түзету және түзетілген шешімнің дұрыстығын қайта тексеру әрекеттерінің жиынтығы.

Программалау әрекеті – студенттің есептің мазмұнын түсінуі, шешімді жоспарлауы, алгоритм құруы, оны программалық кодта іске асыруы, тестілеуі, қателерді анықтап түзетуі және қабылданған шешімді негіздеуі сияқты өзара байланысты әрекеттерінің жиынтығы.

Референттік баға – зерттеуде үлкен тілдік модельдер қалыптастырған бағаларды салыстыру үшін эталон ретінде қабылданған, оқытушының студенттік жұмысқа 0–100 балдық шкала бойынша қойған тұтас бағасы.

Статикалық талдау – программаны орындамай, оның бастапқы кодының құрылымын, синтаксисін, стилін және эталондық немесе басқа дұрыс шешімдермен ұқсастығын зерттейтін программалық кодты талдау тәсілі.

Үлкен тілдік модель – өте үлкен көлемдегі мәтіндер мен программалық кодтарда оқытылған, табиғи тілдегі нұсқауларды түсініп, мәтін мен программалық кодты генерациялауға, талдауға және бағалауға қабілетті нейрожелілік модель (мысалы, GPT, Claude, Gemini).

Функционалдық дұрыстық – программалық шешімнің тапсырма талаптарына сәйкестігін, алгоритмінің қисындылығын және тестілік, оның ішінде шекаралық жағдайларда дұрыс нәтиже беруін сипаттайтын интеллектуалдық модельдің бағалау компоненті.

Шешімді түсіндіру сапасы – студенттің алгоритмнің негізгі қадамдарын, таңдалған тәсілдің себептерін және программалық кодтың жұмыс істеу логикасын жазбаша немесе ауызша негіздеу деңгейін сипаттайтын интеллектуалдық модельдің бағалау компоненті.

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР

ҚР	- Қазақстан Республикасы
ҚР ҒЖБМ	- Қазақстан Республикасы Ғылым және жоғары білім министрлігі
ЖОО	- Жоғары оқу орны
ПҚ	- Программалық қамтамасыздандыру
ЖМББС	- Жалпыға міндетті мемлекеттік білім беру стандарты
ЖИ	- Жасанды интеллект
ПОҚ	- Профессор-оқытушылар құрамы
ПӨ	- Программалық өнім
GenAI	- Generative Artificial Intelligence (генеративті жасанды интеллект)
AI	- Artificial Intelligence (жасанды интеллект)
LLM	- Large Language Model (үлкен тілдік модель)
GPT	- Generative Pre-trained Transformer (алдын ала оқытылған генеративті трансформер)
DKT	- Deep Knowledge Tracing (білім деңгейінің өзгерісін терең оқыту әдістері арқылы қадағалау)
MAE	- Mean Absolute Error (орташа абсолюттік қате)
MD	- Mean Difference (орташа мәндер айырмасы)
VPL	- Virtual Programming Lab (Moodle ортасындағы виртуалды программалау зертханасы модулі)
PAAM	- Problem Analysis Algorithmic Model (есепті талдаудың алгоритмдік моделі)

КІРІСПЕ

Зерттеудің өзектілігі. Қазақстан Республикасының «Білім туралы» Заңында жоғары білім беру бағдарламаларының оқыту нәтижелеріне бағдарлануы және білім алушылардың меңгерген білімі, біліктілігі мен дағдыларын бағалау арқылы оларға қол жеткізу деңгейін анықтау көзделген [1]. Осы ережелер Жоғары және жоғары оқу орнынан кейінгі білім берудің мемлекеттік жалпыға міндетті стандартында нақтыланып, оқу қызметінің, бағалау әдістері мен критерийлерінің күтілетін оқыту нәтижелерімен өзара байланысына қойылатын талаптар айқындалған [2]. Аталған талаптарды іске асыру тәртібі оқу процесін ұйымдастыру және білім алушылардың оқу жетістіктерін ағымдық, аралық және қорытынды бақылауды реттейтін қағидаларда айқындалған [3, 4]. Бұл жоғары білім беру үдерісінде білім алушылардың оқу жетістіктерін объективті, негізді және күтілетін оқыту нәтижелеріне сәйкес бағалаудың маңыздылығын көрсетеді.

Қазіргі кезеңде жоғары білім беру саласындағы өзгерістер цифрлық технологиялардың қарқынды дамуымен тығыз байланысты. Қазақстан Республикасында жоғары білімді және ғылымды дамытудың 2023–2029 жылдарға арналған тұжырымдамасында жоғары білім мен ғылым жүйесін трансформациялаудың кезекті кезеңіне өту қажеттілігі айқындалып, дамушы технологиялар мен болашақ экономикасының талаптарына сәйкес заманауи білім беру бағдарламаларын дамытуға басымдық берілген [5].

Жоғары білім берудегі цифрлық трансформацияның маңызды бағыттарының бірі – жасанды интеллект технологияларын білім беру үдерісіне енгізу. Қазақстан Республикасында жасанды интеллектті дамытудың 2024–2029 жылдарға арналған тұжырымдамасында жасанды интеллект саласындағы адами капиталды дамытуға және осы бағыттағы кадрлық әлеуетті арттыруға басымдық берілген [6]. Жасанды интеллект технологияларын білім беру үдерісінде қолдану оқыту мен білім алудың жаңа мүмкіндіктерін кеңейтіп, білім беру үдерісін ұйымдастырудың қалыптасқан тәсілдеріне өзгерістер енгізуде. 2025 жылы қабылданған «Жасанды интеллект туралы» Заң бұл бағыттың құқықтық негізін айқындап, жасанды интеллект жүйелерін пайдаланудағы ашықтық пен жауапкершілік талаптарын белгіледі [7].

Технологиялық ортаның мұндай өзгерісі болашақ информатика мұғалімдері мен IT мамандарын даярлау мазмұнына да тікелей әсер етеді. Олардың кәсіби даярлығында программалау маңызды орын алады, өйткені оны меңгеру алгоритмдік ойлауды дамытуға, есептерді формализациялауға және программалық жолмен шешу дағдыларын қалыптастыруға негіз болады.

Жоғары оқу орнында программалауды оқытудың теориялық және әдістемелік негіздері отандық зерттеушілер Е.Ы. Бидайбеков [8], Ж.К. Нурбекова [9], Г.Д. Жангисина [10], С.А. Нугманова [11], Т.Х. Хакимова [12], С.Т. Кожахметов [13], А.Т. Байбақтина [14], К.М. Байгушева [15], А.А. Бекежанова [16], Г.И. Аймичева [17] және т.б. еңбектерінде зерттелген. Программалауды оқытудың мазмұны, әдістері мен технологияларына қатысты мәселелер ресейлік ғалымдар В.В. Гриншкун [18], Ф.В. Шкарбан [19], И.В. Левченко [20], М.А. Федотенко [21] және т.б., сондай-ақ шетелдік зерттеушілер

G. Rodrigues [22], I. Calderon [23], A. Shoufan [24], A. Luxton-Reilly [25] және т.б. еңбектерінде қарастырылған.

ЖОО-да программалауды оқытудың теориялық және әдістемелік мәселелерімен қатар, студенттердің оны меңгеру үдерісінің өзіндік ерекшеліктері де зерттеушілер назарын аударуда. Программалауды меңгеру студенттен программалау тілінің синтаксисі мен негізгі құрылымдарын білумен қатар, есепті түсіну және формализациялау, шешу алгоритмін құру, программалық кодты әзірлеу, оны тексеру, қателерді анықтау және түзету сияқты өзара байланысты әрекеттерді орындауды талап етеді. Кейінгі зерттеулер студенттердің программалау ұғымдары мен құрылымдарын түсіну, синтаксис пен семантиканы меңгеру, есептерді шешу және программалау дағдыларын қалыптастыруда бірқатар қиындықтарға кездесетінін көрсетеді [26, 27]. Бұл қиындықтар бұрынғы классикалық зерттеулерде де сипатталған. A. Robins және т.б. бастаушы программалаушының тіл конструкцияларын біле тұра, оларды біртұтас шешімге біріктіре алмайтынын көрсетсе [28], M. McCracken бастаған зерттеу тобы бірінші курс студенттерінің едәуір бөлігі күтілген деңгейдегі программалау есебін толық шеше алмағанын анықтаған [29]. Бұл ерекшеліктер программалаудан білімді бақылау мен бағалауда орындалған тапсырманың соңғы нәтижесімен қатар, программаны жазу үдерісіндегі студент әрекеттерін де ескеру қажеттігін көрсетеді.

Программалаудан студенттердің білімін бақылау мен бағалаудың ғылыми-әдістемелік мәселелері отандық зерттеулерде әртүрлі қырынан қарастырылған. Г.М. Абильдинова [30, б. 6] программалаудан білімді бақылауды эксперттік жүйелер теориясы негізінде ұйымдастыруға арналған электрондық құралды жасау және пайдалану әдістемесін ұсынған. Н.К. Токжигитова [31, б. 8] болашақ информатика мамандарының визуалды программалау бойынша оқу жетістіктерін мультикритериалды бағалау әдіснамасын әзірлеген. А.Е. Сағымбаева, Ж.К. Нурбековамен бірлескен зерттеуімізде генеративті жасанды интеллект технологияларын программалаудан білімді бақылауда қолдану, программалық кодты тексеру, кері байланыс ұсыну мүмкіндіктері мен оларды қолдануға байланысты мәселелер қарастырылды [32, б. 379]. Бұл зерттеулер программалаудан білімді бақылаудың мазмұны мен құралдарының технологиялық өзгерістерге сәйкес кеңейіп келе жатқанын айғақтайды.

Программалаудан білімді бақылау құралдарының дамуы программалық шешімдерді автоматтандырылған тексеру мен бағалау жүйелерінің қалыптасуына негіз болды.

K.M. Ala-Mutka программалау тапсырмаларын автоматтандырылған бағалау тәсілдерін қарастырса [33, б. 83], H. Keuning, J. Jeuring және B. Heeren [34] студенттің программалық шешіміне автоматты кері байланыс қалыптастыру тәсілдерін жүйелеген. P. Ihantola және т.б. автоматты бағалау жүйелерінің даму бағыттарына шолу жасаса [35], J.C. Paiva, J.P. Leal және Á. Figueira осы саладағы кейінгі еңбектерді жүйелеп, бағалаудың функционалдық тестілеуден статикалық талдауға және мазмұнды кері байланысқа дейін кеңейгенін көрсетті [36]. S. Combéfis [37] программалық кодты автоматты бағалау жүйелерінде қолданылатын талдау тәсілдері мен кері байланыс түрлерін зерттеген. M. Messer программалауды оқытудағы автоматтандырылған

бағалау мен кері байланыс құралдарына жүргізген жүйелі шолуда мұндай жүйелердің басым бөлігі программалық шешімнің функционалдық дұрыстығын бағалауға бағытталғанын анықтаған [38, б. 13]. Аталған зерттеулер программалаудан білімді автоматтандырылған бақылаудың программалық кодты тексеру, оның дұрыстығын анықтау және жедел кері байланыс беру бағыттарында дамығанын көрсетеді.

Автоматтандырылған бақылаудың бұл мүмкіндіктері LeetCode, Codeforces, HackerRank сияқты онлайн-платформаларда кеңінен қолданылады. Бұл платформаларда программалық шешім алдын ала белгіленген тестілік жағдайлар бойынша орындалып, алынған нәтиже күтілетін нәтижемен салыстырылады. Мұндай тәсіл программалық шешімнің функционалдық дұрыстығын автоматты түрде тексеруге, тапсырманы бірнеше рет орындап көруге және тексеру нәтижесі бойынша жедел кері байланыс алуға мүмкіндік береді.

Алайда автоматтандырылған бақылау жүйелерінің басым бөлігі программалық шешімнің соңғы нәтижесі мен функционалдық дұрыстығын тексеруге бағытталған. Мұндай бақылауда студенттің есепті түсінуі, шешу алгоритмін құруы, программаны жазу барысында жіберген қателерін анықтауы мен түзетуі, қабылдаған шешімдерін түсіндіруі сияқты программалау үдерісін сипаттайтын әрекеттері толық ескеріле бермейді.

Қазіргі жасанды интеллект технологияларының қарқынды дамуы программалауды оқыту мен студенттердің білімін бақылауға да елеулі өзгерістер енгізуде. Генеративті жасанды интеллект құралдары программалық кодты генерациялауға, толықтыруға, қателерді анықтап түзетуге және программалық шешімді түсіндіруге мүмкіндік береді. Мұндай жағдайда студент ұсынған дайын программалық код оның білім деңгейін толық сипаттайтын жалғыз көрсеткіш бола алмайды. Сондықтан программалаудан білімді бақылауда студенттің соңғы нәтижесімен қатар, тапсырманы орындау үдерісі мен генеративті ЖИ-мен өзара әрекеттесу деректерін де ескеру қажеттілігі артады. Осы деректерді өзара байланысты талдау және оларды білімді бағалауда пайдалану программалаудан білімді бақылауды интеллектуалдық негізде ұйымдастыру қажеттігін айқындайды.

Біздің зерттеуіміз тұрғысынан программалаудан білімді интеллектуалдық бақылау – студенттің программалау тапсырмасын орындау нәтижесі мен үдерісі туралы әртүрлі деректерді жинау, біріктіру және интеллектуалдық талдау негізінде оның білімін белгіленген критерийлер мен көрсеткіштер бойынша бағалауға, сондай-ақ оқытушының педагогикалық шешім қабылдауын қолдауға бағытталған бақылау үдерісі.

Осыған байланысты жоғары оқу орнында программалаудан студенттердің білімін объективті және кешенді бақылау қажеттілігі мен студенттің программалау әрекеті туралы әртүрлі деректерді интеллектуалдық талдау негізінде бағалауды қамтамасыз ететін бақылаудың интеллектуалдық моделін құру мәселесінің ғылыми-әдістемелік тұрғыдан толық шешілмеуі арасында **қарама-қайшылық** бар.

Аталған қарама-қайшылық «*Жоғары оқу орнында студенттердің программалаудан білімін объективті және кешенді бақылауды қамтамасыз ететін интеллектуалдық модельді қалай жасауға және іске асыруға болады?*»

деген зерттеу мәселесін айқындап, зерттеу тақырыбын «Жоғары оқу орнында студенттердің программалаудан білімін бақылаудың интеллектуалдық моделін жасау және іске асыру» деп таңдауға негіз болды.

Осы мәселені ғылыми-әдістемелік тұрғыдан шешу қажеттілігі зерттеу тақырыбының **көкейкестілігін** айқындайды.

Зерттеудің мақсаты: жоғары оқу орны студенттерінің программалаудан білімін объективті және кешенді бақылауды қамтамасыз ететін интеллектуалдық модельді ғылыми тұрғыдан негіздеп жасау және оны іске асыру.

Зерттеу нысаны: жоғары оқу орнында студенттерге программалауды оқыту үдерісі.

Зерттеу пәні: жоғары оқу орны студенттерінің программалаудан білімін бақылаудың интеллектуалдық моделін жасау және іске асыру.

Зерттеудің ғылыми болжамы: егер программалаудан студенттердің білімін бақылау педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысы негізінде жүзеге асырылса, онда студенттің программалаудан білім жағдайын неғұрлым толық анықтауға мүмкіндік туады, өйткені интеллектуалдық бақылау студенттің программалау тапсырмасын орындаудың соңғы нәтижесімен ғана шектелмей, оның программалау әрекетін сипаттайтын деректерді бағалау көрсеткіштері негізінде кешенді талдауға және педагогикалық тұрғыдан интерпретациялауға мүмкіндік береді.

Зерттеудің жетекші идеясы: программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі студенттердің программалау әрекетінің нәтижесі мен үдерісін кешенді бағалауға және олардың білім деңгейін объективті анықтауға мүмкіндік береді.

Зерттеудің мақсаты мен ғылыми болжамына сәйкес келесідей зерттеу **міндеттері қойылды**:

- программалаудан студенттердің білімін бақылаудың қазіргі жағдайын талдау және интеллектуалдық оқу құралдарын қолдану жағдайындағы бақылаудың ерекшеліктерін айқындау;
- программалаудан студенттердің білімін интеллектуалдық бақылау қағидаларын анықтау мен бағалау критерийлерін нақтылау;
- программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін жасау;
- программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі негізінде бақылауды ұйымдастыру әдістерін ұсыну;
- программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін эксперименттік тексеру және оның тиімділігін бағалау.

Зерттеу әдістері: зерттеу мәселесіне қатысты философиялық, психологиялық-педагогикалық және ғылыми-әдістемелік әдебиеттерді, нормативтік құжаттарды, программалауды оқыту мен білімді бақылауға арналған отандық және шетелдік зерттеулерді теориялық талдау және жинақтау; бақылау, сауалнама жүргізу, тестілеу, студенттердің программалау әрекетінің нәтижелері мен үдерісін талдау; педагогикалық эксперимент, эксперимент

нәтижелерін сандық және сапалық талдау, математикалық статистика әдістері қолданылды.

Зерттеудің әдіснамалық және теориялық негіздерін педагогикалық бақылау мен бағалау теориясы (П.И. Пидкасистый [39], И.П. Подласый [40], Н.В. Басова [41], P. Black [42], W. Harlen [43], S. Messick [44] және т.б.); оқу мақсаттары мен танымдық үдерістерді жіктеудің Блум–Андерсон–Кратвол таксономиясы (B.S. Bloom, L.W. Anderson, D.R. Krathwohl [45] және т.б.); информатиканы және программалауды оқыту теориясы мен әдістемесі (Е.Ы. Бидайбеков [8], Ж.К. Нұрбекова [9], Г.Д. Жангисина [10], Т.Х. Хакимова [12], С.Т. Кожахметов [13], К.М. Байгушева [15], С.А. Нугманова [11], В.В. Гриншкун [18] және т.б.); программалаудан білімді бақылау мен бағалау (Г.М. Абильдинова [30], Н.К. Токжигитова [31], А.Е. Сағымбаева [46] және т.б.); программалау тапсырмаларын автоматтандырылған бағалау және кері байланыс беру (K.M. Ala-Mutka [33], H. Keuning, J. Jeuring, B. Heeren [34], S. Combéfis [37], M. Messer [38] және т.б.); интеллектуалдық оқыту жүйелері, студентті модельдеу және білім жағдайын қадағалау (A.T. Corbett, J.R. Anderson [47], C. Piech [48], Y. Shi [49] және т.б.); білім беруде және программалауды оқыту мен бағалауда генеративті жасанды интеллект технологияларын қолдану (А.Е. Сағымбаева [50], Ж.К. Нұрбекова [51], D. Kohen-Vacs, M. Usher, M. Jansen [52], Z. Zhang [53], P. Silva, E. Costa [54] және т.б.) салаларындағы отандық және шетелдік ғалымдардың ғылыми тұжырымдары мен зерттеу нәтижелері құрайды.

Зерттеудің ғылыми жаңалығы:

- интеллектуалдық оқу құралдарын қолдану жағдайында программалаудан студенттердің білімін бақылаудың ерекшеліктері айқындалды.
- программалаудан студенттердің білімін интеллектуалдық бақылаудың қағидалары анықталып, студенттің программалау әрекетінің нәтижесі мен үдерісін кешенді бағалауға мүмкіндік беретін бағалау критерийлері нақтыланды.
- педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысына негізделген программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі жасалды;
- программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі негізінде олардың программалау әрекетінің нәтижесі мен үдерісін қысқа диагностикалық тапсырмалар, деңгейлік бақылау және жобалық жұмысты кезеңдік бақылау арқылы бағалауға бағытталған бақылауды ұйымдастыру әдістері анықталды.

Зерттеудің теориялық маңыздылығы: программалаудан студенттердің білімін интеллектуалдық бақылаудың ерекшеліктерінің айқындалуы, интеллектуалдық бақылау қағидаларының анықталып, студенттердің программалау әрекетінің нәтижесі мен үдерісін бағалауға арналған критерийлердің нақтылануы және программалаудан білімді бақылаудың интеллектуалдық моделінің теориялық тұрғыдан негізделе отырып жасалуында.

Зерттеудің практикалық маңыздылығы: программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі негізінде бақылауды ұйымдастыру әдістерінің әзірленуімен, білімді бақылауға арналған «Bilim AI» бағдарламалық өнімінің жасалуымен және оларды білім беру тәжірибесінде қолдану мүмкіндігімен сипатталады. Зерттеу нәтижелерін жоғары оқу орындарында студенттердің программалаудан білімін бақылау мен бағалауда, сондай-ақ педагог кадрлардың біліктілігін арттыру институттарында информатика мұғалімдерінің кәсіби біліктілігін жетілдіру барысында пайдалануға болады.

Қорғауға ұсынылатын негізгі қағидалар:

- программалаудан студенттердің білімін интеллектуалдық бақылау студенттің программалау тапсырмасын орындау нәтижесімен ғана шектелмей, программалау әрекетін сипаттайтын деректерді ескере отырып, белгіленген критерийлер негізінде кешенді бағалауға бағытталады.

- программалаудан білімді интеллектуалдық бақылаудың қағидалары мен бағалау критерийлері студенттің программалау әрекетінің нәтижесі мен тапсырманы орындау барысын өзара байланыста бағалаудың теориялық негізін құрайды.

- программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысына негізделіп, студенттің программалау әрекетінің нәтижесі мен тапсырманы орындау барысын кешенді бағалауды қамтамасыз етеді.

- интеллектуалдық модель негізінде бақылауды ұйымдастыру әдістері студенттердің программалау әрекетінің нәтижесі мен орындалу барысы туралы әртекті деректерді жинақтау мен талдауға, олардың білімін белгіленген критерийлер бойынша кешенді бағалауға және бағалау нәтижелерін педагогикалық тұрғыдан интерпретациялауға мүмкіндік береді.

Зерттеу нәтижелерін сынақтан өткізу. Зерттеудің негізгі қағидалары мен нәтижелері "АКТ жағдайында инновациялық құралдар және оқыту әдістемесі" атты халықаралық ғылыми-практикалық (Ресей, Красноярск, 2015); «Математикалық модельдеу мен ақпараттық технологиялар білімде және ғылымда" атты VII Халықаралық ғылыми-әдістемелік (Алматы, 2015), «Интеллектуалдық ақпараттық және коммуникациялық технологиялар - «Қазақстан-2050» III халықаралық ғылыми – практикалық (Астана 2016); «Заманауи ғылым және білімнің өзекті мәселелері мен инновация тенденциялары» атты халықаралық ғылыми-тәжірибелік (Түркістан 2017); «Математикалық модельдеу мен ақпараттық технологиялар білімде және ғылымда" атты X Халықаралық ғылыми-әдістемелік (Алматы, 2025); «Білім беруді цифрлық трансформациялау» мәселелер мен шешімдер» атты IV ғылыми-практикалық (Болгария, Русе, 2025); Компьютерлік ғылымдар және инженерия бойынша 10-шы халықаралық (UBMK) (Түркия, Стамбұл, 2025); Қазақстан математиктерінің бірінші конгресі (Шымкент, 2026) атты халықаралық ғылыми-практикалық конференцияларында баяндалды.

Диссертацияның мазмұны бойынша 22 ғылыми еңбек жарияланды: оның ішінде Scopus деректер базасында индекстелетін басылымдарда – 3 (ғылыми журналда – 1, халықаралық конференция материалдарында – 2), Қазақстан

Республикасы Ғылым және жоғары білім министрлігінің Ғылым және жоғары білім саласында сапаны қамтамасыз ету комитеті ұсынған басылымдарда – 9, шетелдік ғылыми журналда – 1, халықаралық ғылыми конференциялар мен конгресс материалдарының жинақтарында – 7 (оның ішінде шетелдік конференция материалдарында – 2) жарияланым жарық көріп, авторлық құқықпен қорғалатын объект – 2.

Зерттеу жұмысы Қазақстан Республикасы Ғылым және жоғары білім министрлігі Ғылым комитетінің гранттық қаржыландыруы бойынша AP23490592 «Генеративті жасанды интеллект технологиясының программалаудан ЖОО студенттерінің білімін бақылау әдістемесіне әсерін зерттеу» ғылыми жобасы аясында орындалды.

Зерттеу кезеңдері: Зерттеу үш кезеңде жүргізілді.

Зерттеудің бірінші кезеңі (2014–2017 жж.) докторантурада оқу кезеңін қамтып, білімді компьютерлік бақылаудың теориялық негіздерін зерттеуге, көпдеңгейлі ақпараттық модельді жасауға және оны ақпараттық жүйе негізінде іске асыруға бағытталды. Осы кезеңде зерттеу нәтижелері ғылыми жарияланымдарда апробацияланып, «Многоуровневая семантическая информационная система контроля результатов обучения» ақпараттық жүйесі әзірленіп, оған авторлық құқықпен қорғалатын объектіге куәлік алынды [55].

Зерттеудің екінші кезеңі (2017–2024 жж.) бірінші кезеңде әзірленген «Многоуровневая семантическая информационная система контроля результатов обучения» ақпараттық жүйесі студенттердің программалаудан білімін бақылауда қолданылды. Жүйені қолдану барысында алынған бақылау нәтижелері талданып, педагогикалық тұрғыдан интерпретацияланды.

Зерттеудің үшінші кезеңі (2024–2025 жж.) зерттеу AP23490592 ғылыми жобасы аясында жалғастырылып, программалаудан студенттердің білімін бақылаудың қазіргі жағдайына теориялық талдау жүргізілді, интеллектуалдық оқу құралдарын қолдану жағдайындағы бақылаудың ерекшеліктері айқындалды. Генеративті жасанды интеллект технологияларының программалаудан білімді бақылаудағы мүмкіндіктері мен оларды қолдануға байланысты мәселелер зерттелді. Программалаудан білімді интеллектуалдық бақылаудың қағидалары анықталып, бағалау критерийлері мен көрсеткіштері нақтыланды. Алынған теориялық және эмпирикалық нәтижелер негізінде программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін жасауға ғылыми-әдістемелік негіз қалыптастырылды.

Зерттеудің төртінші кезеңі (2025–2026 жж.) зерттеу жұмыстары AP23490592 ғылыми жобасы аясында жалғастырылды. Алдыңғы кезеңде алынған теориялық және эмпирикалық нәтижелер негізінде программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі жасалып, оның құрылымдық компоненттері мен олардың өзара байланысы айқындалды. Интеллектуалдық модель негізінде программалаудан білімді бақылауды ұйымдастыру әдістері әзірленіп, оларды практикалық іске асыру жолдары анықталды. Зерттеу нәтижелерін практикалық қолдануға арналған «Bilim AI» бағдарламалық өнімі әзірленіп, оған авторлық құқықпен қорғалатын объектіге куәлік алынды [56]. Ұсынылған интеллектуалдық модель эксперименттік тұрғыдан тексеріліп, оның тиімділігі бағаланды. Эксперимент барысында алынған нәтижелерге сандық және сапалық талдау жүргізіліп, зерттеу

нәтижелері жүйеленді. Зерттеудің негізгі теориялық және практикалық нәтижелері ғылыми жарияланымдарда апробацияланып, қорытындылар мен ғылыми-әдістемелік ұсыныстар тұжырымдалды. Диссертацияны өңдеу жұмыстары жүргізіліп, диссертацияның қорғауға даярлығы аяқталды.

Диссертацияның құрылымы. Диссертация кіріспеден, екі бөлімнен, қорытындыдан, пайдаланылған әдебиеттер тізімінен және қосымшалардан тұрады.

Кіріспеде зерттеу мәселесі қойылып, оның өзектілігі негізделді. Зерттеудің мақсаты, нысаны, пәні, ғылыми болжамы мен міндеттері айқындалып, әдіснамалық негіздері мен зерттеу әдістері сипатталды. Зерттеудің ғылыми жаңалығы, теориялық және практикалық маңыздылығы ашылып, қорғауға ұсынылатын қағидалар тұжырымдалды.

Бірінші тарауда программалаудан студенттердің білімін бақылаудың теориялық негіздері зерделеніп, интеллектуалдық оқу құралдарын қолдану жағдайындағы бақылаудың ерекшеліктері айқындалды. Студенттің программалық шешімі мен оны әзірлеу барысын өзара байланыста бағалау қажеттігі негізделді. Осыған сәйкес интеллектуалдық бақылаудың қағидалары анықталып, есептің мазмұнын түсінуді, алгоритмдік шешімді, программаның функционалдық дұрыстығын, код сапасын, қатені анықтау мен түзетуді және шешімді негіздеуді бағалау критерийлері мен көрсеткіштері нақтыланды.

Екінші тарауда программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі жасалып, оның бес компонентінің өзара байланысы негізделді. Модельді іске асыру үшін қысқа диагностикалық тапсырмаларды қолдану, деңгейлік бақылау және жобалық жұмысты кезеңдік бақылау әдістері айқындалды. Тапсырманың нәтижесі мен орындалу барысы туралы деректерді талдау, оларды бағалау көрсеткіштерімен сәйкестендіру және бағалау нәтижелерін педагогикалық тұрғыдан интерпретациялау тәртібі көрсетілді. Зерттеу нәтижелерін практикада қолдануға арналған «Bilim AI» бағдарламалық өнімі әзірленді. Ұсынылған модель эксперименттік тұрғыдан тексеріліп, интеллектуалдық бақылау нәтижелері оқытушының бағалауымен салыстырылды.

Қорытындыда зерттеу міндеттеріне сәйкес алынған теориялық, әдістемелік және эксперименттік нәтижелер жүйеленді. Программалаудан студенттердің білімін интеллектуалдық бақылауды ұйымдастыру жөніндегі тұжырымдар жасалып, модельді әртүрлі программалау тілдері мен оқу курстарына бейімдеу бағыттары көрсетілді.

1 ЖОҒАРЫ ОҚУ ОРНЫ СТУДЕНТТЕРІНІҢ ПРОГРАММАЛАУДАН БІЛІМІН ИНТЕЛЛЕКТУАЛДЫҚ МОДЕЛЬ НЕГІЗІНДЕ БАҚЫЛАУДЫҢ ТЕОРИЯЛЫҚ НЕГІЗДЕРІ

1.1 Программалаудан студенттердің білімін бақылаудың қазіргі жағдайы

Жоғары білім беру жүйесінде бақылау оқу үдерісінің маңызды құрамдас бөлігі болып табылады. Ол білім алушының қол жеткізген оқу нәтижелері туралы ақпарат алуға, оқу әрекетінің барысын қадағалауға және алынған деректер негізінде оқу үдерісін одан әрі ұйымдастыруға мүмкіндік береді. Осы тұрғыдан бақылаудың мәні қол жеткізілген нәтижені тіркеумен ғана шектелмейді, сонымен қатар нақты оқу нәтижелері мен күтілетін нәтижелер арасындағы сәйкестікті анықтауды көздейді. Сондықтан осы зерттеуде бақылау оқу нәтижелері туралы деректерді алу, талдау және педагогикалық тұрғыдан интерпретациялауды қамтитын үдеріс ретінде қарастырылады.

Білімді бақылау білім алушының белгілі бір ақпаратты қайта жаңғырту қабілетімен ғана шектелмей, оны түсіну, қолдану, талдау және соның негізінде тиісті танымдық әрекеттерді жүзеге асыру деңгейін де анықтауды көздейді [45]. Демек, білімді бақылауда білімнің формалды түрде бар немесе жоқ екендігін белгілеуден гөрі, оның игерілу дәрежесі мен нақты әрекетте көріну ерекшеліктерін айқындау маңызды.

Білімді бақылаудың мәнін ашуда «бақылау» және «бағалау» ұғымдарының арақатынасын нақтылау маңызды. Ғылыми-педагогикалық әдебиеттерде білім алушының оқу жетістіктері туралы деректерді алу, оларды интерпретациялау және алынған нәтижелер негізінде қорытынды жасау бағалау үдерісінің өзара байланысты әрекеттері ретінде қарастырылады [42, 43]. P. Black және D. Wiliam білім алушының жетістіктері туралы деректерді алу, интерпретациялау және оларды оқытудың келесі қадамдары туралы шешім қабылдауда пайдалануды қалыптастырушы бағалаудың маңызды үдерістері ретінде қарастырады [42]. W. Harlen бағалауды білім алушының оқу жетістіктері туралы деректерді жинау, интерпретациялау және олардың негізінде пайымдау жасау үдерісі ретінде сипаттайды [43]. Осы зерттеуде аталған тұжырымдарға сүйене отырып, бақылау студенттің оқу нәтижелері туралы деректерді алу мен талдауды қамтитын үдеріс ретінде, ал бағалау осы деректерді белгіленген критерийлер тұрғысынан интерпретациялау және қорытынды нәтиже қалыптастырумен байланысты бақылаудың құрамдас бөлігі ретінде қарастырылады.

Педагогикалық бақылау мен бағалаудың нәтижелілігі тек білім алушының оқу әрекеті туралы деректерді жинаумен емес, сол деректер негізінде оның нақты білім жағдайы туралы негізді қорытынды жасау мүмкіндігімен анықталады. Осы тұрғыдан бағалаудың валидтілігі бақылау нәтижелерінің бағаланатын білім мазмұны мен оқу нәтижелеріне сәйкестігін сипаттайтын маңызды талаптардың бірі болып табылады [44]. Бақылау барысында тіркелген нәтиже студент білімінің тікелей көрінісі ретінде емес, оның белгілі бір білім компонентін меңгеруі туралы қорытынды жасауға мүмкіндік беретін педагогикалық дерек ретінде қарастырылады. Сондықтан бағалау тапсырмалары мен қолданылатын көрсеткіштер студенттің қандай білімін,

білігін немесе әрекетін сипаттайтыны тұрғысынан негізделуі қажет. Бұл талап әсіресе программалаудан білімді бақылауда маңызды, өйткені программаның орындалу нәтижесі студенттің алгоритмді түсінуін, программалық шешімді негіздеуін және қателерді анықтап, түзету қабілетін толық сипаттай алмайды.

Бақылаудың педагогикалық мүмкіндігі оның диагностикалық қызметінен айқын көрінеді. Диагностикалық тұрғыдан бақылау білім алушының қол жеткізген нәтижесін тіркеумен шектелмей, оның білім мен дағдыларындағы игерілген және жеткіліксіз қалыптасқан жақтарды, сондай-ақ оқу әрекетінде кездесетін қиындықтарды анықтауға бағытталады [57]. Мұндай түсінік бақылау нәтижесін қорытынды балл түріндегі интегралдық көрсеткішпен ғана сипаттаудың жеткіліксіздігін көрсетеді және білім алушының білім жағдайы туралы неғұрлым толық ақпарат қалыптастыру қажеттігін алға шығарады.

Бақылаудың өткізілу кезеңі, мақсаты және оқу үдерісіндегі орнына сәйкес педагогикалық әдебиеттерде алдын ала, ағымдық, аралық (кезеңдік) және қорытынды бақылау түрлері ажыратылады [58]. Олардың айырмашылығы бақылаудың оқу үдерісінде жүзеге асырылу кезеңімен ғана емес, алынатын ақпараттың педагогикалық мақсатымен де анықталады: бастапқы білім жағдайын айқындау, меңгеру барысы мен динамикасын қадағалау, белгілі бір оқу кезеңінің нәтижесін анықтау немесе пәнді меңгерудің қорытынды нәтижесін белгілеу. Осыған байланысты бақылаудың әрбір түрі білім алушының білім жағдайы туралы әртүрлі мазмұндағы деректерді қалыптастырады.

Бақылаудың аталған жалпы педагогикалық қағидалары нақты оқу пәнінің мазмұны мен оқу әрекетінің ерекшеліктеріне сәйкес жүзеге асырылады. Осы тұрғыдан информатикадан білімді бақылаудың теориялық және әдістемелік мәселелері отандық зерттеулерде де қарастырылған. Е.Ы. Бидайбеков, Ж.К. Нурбекова, А.Е. Сағымбаева және басқа зерттеушілердің еңбектерінде бақылау нәтижелерінің ақпараттық сипаты, информатиканы оқыту нәтижелерін анықтау мен өлшеу, бақылаудың пән мазмұнына сәйкестігі мәселелері зерттелген [46, 59, 60]. А.Е. Сағымбаеваның зерттеуінде информатикадан білімді бақылауды ұйымдастыруда бақылаудың түрі мен әдістемесін пән мазмұнына сәйкес таңдау қажеттігі негізделіп, бақылаудың диагностикалық және кері байланыс беру мүмкіндіктері қарастырылған [46].

Аталған тұжырымдар жалпы педагогикалық бақылау қағидаларын информатиканы оқытуда пән мазмұны мен оқу әрекетінің ерекшеліктеріне сәйкес нақтылау қажеттігін көрсетеді. Бұл талап программалаудан білімді бақылауда ерекше мәнге ие, өйткені программалауды меңгеру тек теориялық біліммен емес, есепті шешу, алгоритм құру, программалық кодты жазу, тестілеу және қателерді анықтау мен түзету барысында жүзеге асырылатын әрекеттермен де байланысты.

Студент білімінің нақты әрекет барысында көрінуін түсіндіруде іс-әрекеттік тәсілдің қағидалары маңызды. Іс-әрекеттік тәсілдің психологиялық негіздері А.Н. Леонтьевтің еңбектерінде жүйеленіп, іс-әрекет адамның қоршаған ортамен белсенді өзара әрекеттесуінің және қойылған мақсатқа жетуге бағытталған әрекеттер жүйесі ретінде қарастырылады [61]. Іс-әрекеттің құрылымында мақсатқа бағытталған әрекеттер мен олардың нақты жағдайда жүзеге асуын қамтамасыз ететін операциялардың ажыратылуы адамның қол

жеткізген нәтижесін оны орындау үдерісінен бөліп қарастырмауға мүмкіндік береді. Білімді меңгеру білім алушының белгілі ақпаратты меңгеруімен ғана емес, оны нақты міндеттерді шешуге бағытталған әрекеттерде қолдануымен де байланысты.

Іс-әрекеттік тәсілдің бұл қағидалары қазіргі қазақстандық педагогикалық зерттеулерде де жалғасын тапқан. А.К. Айтпаева іс-әрекеттік тәсілді болашақ педагогтерді даярлаудың практикалық бағыттылығын қамтамасыз ететін негіз ретінде қарастырып, білімді меңгеруді білім алушының белсенді оқу әрекетімен байланыстырады [62]. Осы тұрғыдан студенттің білімін оның оқу тапсырмаларын орындау әрекетінен бөліп қарастыруға болмайды. Сондықтан білімді бақылауда студенттің қол жеткізген нәтижесімен қатар, сол нәтижеге жету барысында меңгерген білімін қалай қолданатынын да ескеру қажет [63].

Бұл тұжырым программалаудан білімді бақылау үшін ерекше мәнге ие. Программалауда студенттің білімі программалау тілінің синтаксисі мен семантикасын білуімен ғана шектелмей, оны есепті шешуге бағытталған әрекеттер барысында қолдануынан көрінеді. Сондықтан программалаудан білімді бақылауда студенттің соңғы нәтижесімен қатар, сол нәтижеге жету барысында жүзеге асыратын әрекеттерін де ескеру қажет.

Ж.К. Нурбекованың программалауды оқытудың теориялық-әдіснамалық негіздеріне арналған зерттеуінде программалауды меңгеру дайын программалық конструкцияларды есте сақтаумен шектелмей, есептерді алгоритмдік негізде шешу және оларды программалау құралдары арқылы жүзеге асырумен байланыста қарастырылады [64]. Бұл тұжырым программалаудан білімнің тек теориялық білім арқылы емес, есепті шешуге бағытталған алгоритмдік және программалық әрекеттер арқылы да көрінетінін негіздеуге мүмкіндік береді. Демек, программалаудан білімді бақылауда студенттің теориялық білімін тексерумен қатар, оның программалау тапсырмасын орындау барысында жүзеге асыратын әрекеттерін де ескеру қажет.

Программалауды меңгеруде программалаудан студенттердің білім мен есептерді шешу дағдыларының өзара байланысы шетелдік зерттеулерде де қарастырылған [25]. S.I. Malik және т.б. программалауды жаңадан үйренушілер программалау тілінің синтаксисі мен семантикасын меңгерумен қатар, есептерді шешу дағдыларын дамытуға да назар аударуы қажеттігін көрсетеді [65]. Авторлар программалауға кіріспе курсын оқытуда есеп шартының кіріс деректерін, орындалатын әрекеттерді және шығыс нәтижесін талдауға бағытталған Problem Analysis Algorithmic Model (PAAM) моделін ұсынған. Зерттеу нәтижелері аталған модельді қолдану студенттердің есептерді шешу дағдыларын дамытуға және есеп шартының кіріс деректерін, орындалу үдерісі мен шығыс нәтижесін түсінуіне оң әсер еткенін көрсеткен [65].

Осы зерттеулердегі тұжырымдарды салыстыру программалауды меңгерудің программалау тілі туралы біліммен ғана шектелмейтінін, сол білімді есепті шешуге бағытталған әрекеттерде қолданумен байланысты екенін көрсетеді [28, 29]. Демек, программалаудың пәндік ерекшеліктерінің бірі программалау бойынша білім мен оны есептерді шешуде қолданудың өзара байланысынан көрінеді.

Программалаудың есепті шешумен байланысы программалау тапсырмасын орындау барысында жүзеге асырылатын әрекеттердің құрылымынан да көрінеді.

F.P. Deek, M. Turoff және J.A. McHugh есептерді шешу мен программа жазуды өзара кіріктіре қарастырып, программалауға қажетті когнитивтік білім мен дағдыларды есепке алатын пәндік-бағдарланған модель ұсынған [66]. Модельде программа жазу есепті тұжырымдау (formulating), шешуді жоспарлау (planning), жобалау (designing), программалық іске асыру (translating), тестілеу (testing) және шешімді ұсыну (delivering) кезеңдері арқылы сипатталған [66].

Программалау тапсырмасын орындау барысында жүзеге асырылатын әрекеттердің өзара байланысы олардың қатаң сызықтық реттілікпен орындалмайтындығынан да көрінеді. O. Kurniawan және т.б. бастаушы программалаушыларға арналған PCDIT құрылымын сызықтық емес есептерді шешу құрылымы ретінде қарастырады [67]. Программалау әрекетінің қайталанатын сипаты программалық іске асыру барысында да байқалады. M.C. Jadud бастаушы программалаушылардың compilation behaviour деп анықталған әрекетін зерттеп, оны студенттің программалық кодты бірнеше рет өңдеуі және компиляциялауы барысында жүзеге асатын программалау әрекеті ретінде қарастырады [68]. Автор екі жыл бойы Java тілінде программалауды үйренген бірінші курс студенттерінің бір компиляциядан келесі компиляцияға дейінгі мыңдаған программалық код нұсқаларын жинап, талдаған. Зерттеуде кодты өңдеу–компиляциялау (edit–compile) циклі студенттердің программаны синтаксистік тұрғыдан дұрыс күйге келтіруге бағытталған әрекеттерімен байланыстырылған [68].

Қарастырылған зерттеулерді талдау негізінде 1-суретте программалау тапсырмасын орындау барысында студент жүзеге асыратын негізгі әрекеттер көрсетілді.

Программалау тапсырмасын орындаудағы аталған әрекеттер мен жоғарыда қарастырылған зерттеулердің нәтижелерін жинақтау программалаудың білімді бақылау тұрғысынан маңызды бірқатар пәндік ерекшеліктерін анықтауға мүмкіндік береді.

Біріншіден, программалауды меңгеру программалау тілі туралы білімді есептерді шешуде қолданумен өзара байланысты. Программалау тілінің синтаксисі мен семантикасын білу студенттің программалауды меңгеруін толық сипаттамайды; бұл білім есепті талдау, шешу тәсілін анықтау және оны программалау құралдары арқылы жүзеге асыру барысында қолданылады. Осы себепті бақылау студенттің теориялық білімін есепті шешу әрекетінде қолдануымен байланысты ұйымдастырылуы тиіс.

Екіншіден, программалау есепті шешуге бағытталған бірнеше өзара байланысты әрекеттерді қамтиды. Есепті талдау, алгоритмдік шешімді жобалау, оны программалық іске асыру, тестілеу, қателерді анықтау және түзету сияқты әрекеттер программалау тапсырмасын орындаудың құрамдас бөліктері болып табылады. Бұл программалаудың есепті шешуге бағытталған көпқұрамды әрекеттік сипатымен ерекшеленетінін көрсетеді.



Сурет 1 – Программалау тапсырмасын орындау барысындағы студенттің әрекеттері

Үшіншіден, программалау тапсырмасының соңғы нәтижесі студенттің программалаудан білімін толық сипаттай бермейді. Дұрыс жұмыс істейтін программа студенттің оны қандай жолмен құрғанын және өзі құрған шешімнің құрылымы мен орындалу логикасын қаншалықты түсінетінін толық ашпайды. Сондықтан программалаудың пәндік ерекшеліктерінің бірі студент білімінің тек соңғы программалық нәтижеде ғана емес, оны алу барысында жүзеге асырылатын әрекеттер мен құрылған шешімді түсінуінде де көрінуімен сипатталады.

Төртіншіден, программалау тапсырмасын орындау қатаң сызықтық әрекеттер тізбегімен шектелмейді. Есепті шешу барысында жекелеген әрекеттер қайта қаралып, программалық код бірнеше рет өңделіп және компиляцияланып отыруы мүмкін. Осы тұрғыдан программалаудың пәндік ерекшеліктерінің бірі программалау әрекетінің итерациялық сипатымен анықталады.

Аталған пәндік ерекшеліктер бақылау нысанын соңғы нәтижеден студенттің есепті шешу әрекетіне дейін кеңейтуді талап етеді.

Программалаудың анықталған пәндік ерекшеліктері студенттің білімі туралы әртүрлі сипаттағы ақпарат алуды қажет етеді. Осыған байланысты программалаудан білімді бақылауда қолданылатын тәсілдердің студент білімінің қандай қырларын анықтауға мүмкіндік беретінін талдау маңызды. Қазіргі зерттеулерде программалаудан білімді бақылау теориялық білімді тексеруден бастап программалық тапсырмалардың орындалуын, программалық

кодты, тапсырманы орындау барысында қалыптасатын деректерді және студенттің программалау әрекеттерін талдауға дейінгі әртүрлі тәсілдер арқылы жүзеге асырылады. Осы ерекшеліктер бақылау барысында қандай деректер алынатынын және олар студент білімінің қай қырларын сипаттайтынын жүйелі талдауды қажет етеді

Программалаудан білімді бақылауды жетілдірудің әдістемелік және технологиялық жолдары отандық зерттеулерде де қарастырылған. Г.М. Абильдинова студенттердің программалаудан білімін эксперттік жүйелер теориясы негізінде бақылау мәселесін зерттеген [30]. Автор программалаудан білімді бақылауды студентті тестілеу немесе әзірленген программалық өнімнің соңғы нәтижесін тексерумен шектеу программалау әрекетін толық қамтуға мүмкіндік бермейтінін көрсетеді [30]. Осыған байланысты зерттеуде бақылау студенттің теориялық білімімен қатар, программа жазу, есептерді жобалау және әзірлеу, программалық өнім жасау, зерттелетін объектінің моделін құру сияқты әрекеттерімен байланыста қарастырылған[30].

Бақылауды жүзеге асыру үшін эксперттік жүйелер теориясына негізделген электрондық құрал әзірленген. Оның білім қоры семантикалық желі түрінде құрылымдалып, қорытынды шығару «Егер–онда» ережелері негізінде ұйымдастырылған [30]. Электрондық құралда бақылау сұрақтары мен әртүрлі формадағы тест тапсырмалары қолданылып, бақылау нәтижесіне сәйкес баға мен ұсыным қалыптастырылған. Сонымен қатар есептер модулінде студентке ішінара дайын программалық код ұсынылып, жүйе жіберілген қатенің орнын анықтап, оны синтаксистік немесе семантикалық қате ретінде жіктеген [30]. Бұл зерттеуде программалаудан білімді бақылаудың теориялық білімді тексеру, программалық тапсырмаларды орындау және программалық қателерді анықтау сияқты бірнеше нысаны бір электрондық құрал аясында біріктірілгенін көрсетеді.

Н.К. Тоқжігітова болашақ информатика мамандарының визуалды программалау бойынша оқу нәтижелерін бағалау мәселесін зерттей отырып, программалауды меңгеру нәтижелерінің көпөлшемді сипатына назар аударады. Зерттеуде оқу нәтижелері қалыптасқан құзыреттілікпен және студент әзірлеген программалық өніммен байланыста қарастырылып, оларды объективті бағалау әртүрлі әдістер мен критерийлерді қолдануды қажет ететіні негізделген [31]. Осыған сәйкес автор әрқайсысы болашақ маман даярлығының белгілі бір қырын сипаттайтын бірнеше критерийді бір мезгілде ескеруге негізделген мультикритериалды бағалау моделін ұсынады [31]. Бұл тәсіл программалаудан оқу нәтижесін бір ғана көрсеткішпен сипаттаудың жеткіліксіздігін және бағалауда оның әртүрлі қырларын өзара байланыста ескеру қажеттігін көрсетеді.

Отандық зерттеулерде программалаудан білімді бақылауды жетілдіру эксперттік жүйелерді қолдану, бақылаудың әртүрлі формаларын біріктіру және бірнеше бағалау критерийлерін ескеру бағыттарында қарастырылған. Сонымен қатар қазіргі зерттеулерде бақылау нысанын кеңейтіп, студенттің дайын программалық нәтижесімен қатар, тапсырманы орындау барысында қалыптасатын деректерді, қателерді анықтау және түзету әрекеттерін, сондай-ақ өзі құрған программалық шешімді түсінуін анықтауға бағытталған тәсілдер ұсынылуда.

Программалаудан білімді бақылаудың осындай бағыттарының бірі студенттің тапсырманы орындау барысында қалыптасатын деректерді талдауға негізделеді. M.M. Villamor программалаудағы ілгерілеуді бағалауда студенттердің компиляциялау әрекеттері мен программалық кодтарын зерттеу және өлшеуге негізделген үдеріске бағытталған тәсілді қарастырады [69]. Зерттеуде қателер мен программалау күйінің өзгерістерін сандық тұрғыдан сипаттауға арналған бірқатар метрикалар жүйеленген [69]. Үдеріске бағытталған мұндай тәсілдер соңғы программалық нәтижемен қатар, программаны құру барысында қалыптасатын деректерді талдау арқылы студенттің программалау әрекеті туралы қосымша ақпарат алуға мүмкіндік береді [70].

Программалау үдерісінде қалыптасатын осындай деректердің маңызды бөлігі студенттің жіберген қателері және оларды түзету әрекеттерімен байланысты. S. Yang және т.б. қателерді анықтау және түзетуді оқытуға арналған зерттеулерді жүйелі талдау негізінде оның бірқатар когнитивтік және когнитивтік емес дағдылармен байланысты күрделі әрекет екенін көрсетеді [22]. Зерттеуде қателерді анықтау және түзету дағдыларын бағалау тәсілдері мен олардың нәтижелері қарастырылып, студенттердің жүйелі debugging стратегияларын меңгеруі жеке проблема ретінде белгіленген [71]. Осы тұрғыдан қателерді анықтау және түзетуге бағытталған тапсырмалар студенттің программалау әрекетінің осы қыры туралы диагностикалық ақпарат алуға мүмкіндік береді.

Алайда программалау үдерісі мен қателерді түзету әрекеттері туралы деректер студенттің өзі құрған программалық шешімді қаншалықты түсінетінін тікелей анықтай бермейді. Осы мәселені анықтаудың бір тәсілі студенттің өз программалық шешімін түсіндіруіне негізделеді. T. Lehtinen, A. Lukkarinen және L. Haaranen программалау тапсырмасын орындағаннан кейін студенттерге өздері құрған программаның құрылымы мен орындалуын түсіндіруге бағытталған сұрақтар (questions about learner's code – QLCs) ұсынған [72]. Зерттеу нәтижесінде студенттердің үштен бірінің дұрыс программа тапсырғанына қарамастан, өз программалық кодын түсіндіруде қиындықтарға кездескені анықталған. Сонымен қатар өз коды туралы сұрақтарға дұрыс жауап беру студенттердің оқу жетістігі және курсты жалғастыруымен жай ғана дұрыс программа тапсыруға қарағанда неғұрлым күшті байланыс көрсеткен [72]. Бұл тәсіл дұрыс программалық нәтиженің алынуын тексерумен қатар, студенттің өзі құрған программаның құрылымы мен орындалу логикасын түсінуі туралы қосымша бақылау деректерін алуға мүмкіндік береді.

Осы зерттеулер бақылау нысанының соңғы программалық нәтижеден программалау үдерісі мен шешімді түсіндіруге дейін кеңейгенін көрсетеді

Программалаудан білімді бақылауда студент білімінің әртүрлі қырларын анықтауға бағытталған тест тапсырмалары, практикалық программалау тапсырмалары, алгоритмдік есептер, программалық қателерді анықтау және түзету (debugging) тапсырмалары, сондай-ақ жобалық жұмыстар қолданылады. Олардың әрқайсысы бақылаудың белгілі бір міндетін жүзеге асырады: тест тапсырмалары программалау ұғымдары мен тіл конструкциялары туралы білімді анықтауға, практикалық программалау тапсырмалары мен алгоритмдік есептер білімді есептерді шешуде қолдануды бағалауға, debugging

тапсырмалары программалық қателерді анықтау және түзету әрекеттерін бақылауға, ал жобалық жұмыстар программалау білімін кешенді практикалық тапсырмаларды орындауда қолдануды бағалауға мүмкіндік береді.

Сонымен бірге программалаудың практикалық және әрекеттік сипаты студенттің оқу нәтижесін бір ғана бақылау формасы арқылы анықтаудың жеткіліксіздігін көрсетеді [73]. Мысалы, Y. Wang және т.б. студенттің қорытынды бағасы бір ғана жазбаша емтихан нәтижесіне негізделетін дәстүрлі бағалау тәсілі программалау тілдерін оқытудың жаңа талаптарына сәйкес келмейтінін атап көрсетеді [74]. Осыған байланысты авторлар студенттердің программа жазу, басқа студенттердің программаларын рецензиялау және өз программаларын қайта өңдеу әрекеттерін қамтитын өзара кодты рецензиялауға (peer code review) негізделген бағалау тәсілін ұсынған [74].

Программалаудан білімді бақылауда практикалық тапсырмалардың қолданылуы студенттер ұсынған программалық шешімдерді тексеру қажеттілігін туындатады. Студенттер саны мен орындалатын программалау тапсырмаларының көлемі артқан жағдайда әрбір программалық шешімді оқытушының қолмен тексеруі едәуір уақыт пен еңбек ресурсын талап етеді [75]. Осыған байланысты программалық шешімдерді тексеру мен бағалауды автоматтандыру программалаудан білімді бақылауды дамытудың маңызды бағыттарының біріне айналды [35, 36].

Программалау тапсырмаларын автоматтандырылған бағалаудың мүмкіндіктері K.M. Ala-Mutka зерттеуінде жүйеленген [33]. Автор программалау тапсырмалары формалды құрылымы бар орындалатын программалар түрінде ұсынылатындықтан, олардың кейбір сипаттамаларын автоматты түрде бағалауға болатынын көрсетеді. Автоматтандырылған бағалаудың негізгі шарты ретінде бағалау нысандарының сандық тұрғыдан өлшенуі белгіленеді, ал сандық тұрғыдан тікелей өлшеуге келмейтін сипаттамаларды бағалауда жартылай автоматтандырылған тәсілдерді қолдану мүмкіндігі қарастырылады [33]. Зерттеуде программаларды статикалық және динамикалық бағалаудың әртүрлі тәсілдері талданып, автоматтандырылған құралдардың оқытушыға тапсырмаларды бағалауда көмектесу және студенттің жұмыс үдерісін автоматты кері байланыс арқылы қолдау мүмкіндігі көрсетілген. Автор автоматтандырудың негізгі артықшылықтарына бағалаудың жылдамдығын, қолжетімділігін, бірізділігі мен объективтілігін жатқызады [33]. Сонымен қатар мұндай құралдарды тиімді қолдану программалау тапсырмалары мен оларды бағалау параметрлерін мұқият педагогикалық жобалауды талап ететіні атап көрсетілген [33].

Сонымен қатар бағалауды автоматтандыру оның педагогикалық тұрғыдан толыққанды болуын автоматты түрде қамтамасыз етпейді. Автоматтандырылған жүйенің программаның тестілік жағдайлардан өтуін дәл және бірізді тексеруі бағалаудың техникалық сенімділігін арттырғанымен, алынған нәтиженің студенттің программалаудан білімін қаншалықты толық сипаттайтыны жеке мәселе болып қалады. Егер бақылау тек программаның соңғы нәтижесіне негізделсе, студенттің алгоритмді таңдау негіздемесі, программалық шешімді түсінуі, қателерді анықтау және түзету әрекеті сияқты білім компоненттері бағалаудан тыс қалуы мүмкін. Сондықтан программалаудан білімді бақылауда автоматтандырылған нәтижелерді

студенттің білім жағдайы туралы педагогикалық қорытынды жасауға мүмкіндік беретін басқа да деректермен толықтыру қажеттілігі туындайды.

Программалау тапсырмаларын автоматтандырылған тексеруді практикада іске асырудың кең таралған формаларының бірі – online judge қағидатына негізделген программалау платформаларын пайдалану. Мұндай платформаларда программалық шешім тестілік жағдайлар бойынша автоматты түрде тексеріліп, оның орындалу нәтижесі анықталады. А. Жақсылықов және т.б. LeetCode платформасында программалау есептерін шешу нәтижелерін зерттеу барысында программалық шешімдерді тексерудің қабылданды, компиляция қатесі, қате жауап, уақыт шектеуінен асып кетті, жад шектеуінен асып кетті, орындалу қатесі және басқа да нәтижелерін пайдаланған [76]. Бұл нәтижелер программаның орындалу дұрыстығын, программаны орындау кезінде туындайтын қателердің жекелеген түрлерін, сондай-ақ уақыт пен жады шектеулеріне сәйкестігін анықтауға мүмкіндік береді [76].

Автоматтандырылған бағалау құралдарының кейінгі даму жағдайы M. Messer және т.б. жүргізген жүйелі шолуда қарастырылған [38]. Авторлар 2017–2021 жылдары жарияланған 121 зерттеуді талдап, автоматтандырылған бағалау құралдарын бағаланатын дағдылар, қолданылатын тәсілдер, программалау тілінің парадигмасы, автоматтандыру дәрежесі және бағалау әдістері бойынша жүйелеген [38]. Зерттеу нәтижелері қарастырылған құралдардың басым бөлігі программалау тапсырмаларының дұрыстығын бағалауға бағытталғанын көрсетеді. Бұл ретте динамикалық тәсілдерде негізінен модульдік тестілеу (unit testing) қолданылса, статикалық талдау тәсілдері студенттің программалық шешімін эталондық шешіммен немесе басқа дұрыс шешімдермен салыстыруға негізделеді [38]. Мұндай құралдардың көпшілігі толық автоматтандырылған бағалауды жүзеге асырып, студентке жедел кері байланыс алуға және тапсырманы бірнеше рет қайта тапсыруға мүмкіндік береді. Сонымен қатар берілетін кері байланыс көбіне тестілердің орындалу нәтижесімен, күтілетін және нақты алынған нәтижелермен немесе программалық шешімнің эталондық шешімнен айырмашылығымен шектелетіні анықталған [45, 77]. Кодтың сүйемелдеуге жарамдылығы, оқылымдылығы мен құжатталуын бағалайтын құралдардың саны салыстырмалы түрде аз және мұндай сипаттамаларды бағалауда негізінен код сапасының метрикаларына сүйенетін статикалық талдау тәсілдері қолданылған [38].

Жоғарыда қарастырылған зерттеулер автоматтандырылған бағалау программалау тапсырмаларын тексерудің тиімділігін арттырғанымен, қолданылатын құралдардың басым бөлігі программаның дұрыстығын анықтауға бағытталғанын көрсетеді [33, 38, 78]. Сонымен қатар автоматтандырылған бағалаудың дамуымен программалық код пен студенттің программалау әрекеті туралы алынатын деректерді талдаудың жаңа тәсілдері қалыптасуда.

Осындай бағыттардың бірі қазіргі зерттеулерде қарқынды дамып келе жатқан жасанды интеллект әдістерін автоматтандырылған бағалауда қолданумен байланысты. Жасанды интеллектке негізделген тәсілдер программаның орындалу нәтижесімен қатар, программалық кодтың әртүрлі сипаттамаларын талдау және алынған деректер негізінде бағалау мен кері байланысты қалыптастыру мүмкіндіктерін кеңейтеді.

Жасанды интеллектке негізделген автоматтандырылған бақылаудың бағыттарының бірі машиналық және терең оқыту әдістерін қолданумен байланысты. M. Wu және т.б. программалау тапсырмаларын орындау барысында студенттерге автоматты кері байланыс қалыптастыру үшін адамның қатысуымен жүзеге асырылатын rubric sampling тәсілін ұсынған [79]. Бұл тәсіл студенттердің тапсырманы орындауы туралы жеткілікті тарихи деректер болмаған жағдайда да кері байланыс қалыптастыруға бағытталған. Авторлар ұсынған тәсіл кері байланысты студент шешімінің нақты бөліктерімен байланыстыруға және студенттің қате түсініктерін оқытушы қолданатын тұжырымдар арқылы сипаттауға мүмкіндік беретінін көрсеткен. Зерттеу нәтижесінде rubric sampling тәсілінің үлкен көлемдегі деректерді қажет ететін алгоритмдермен салыстырғанда жоғары дәлдік көрсетіп, адам деңгейіндегі сәйкестікке жақындағаны анықталған [79].

Соңғы жылдары үлкен тілдік модельдер мен генеративті жасанды интеллект технологияларының дамуы программалаудан білімді бақылау мен кері байланысты автоматтандырудың жаңа мүмкіндіктерін қалыптастырды [80]. А.Е. Сағымбаева және т.б. программалаудан ЖОО студенттерінің білімін бақылауда генеративті жасанды интеллект технологияларын қолдану мүмкіндіктерін талдап, олардың программалық кодты тексеру, қателерді анықтау және түзету, программа құрылымын талдау, тапсырмаларды генерациялау және жекелендірілген кері байланыс беру бағыттарында қолданылатынын көрсетеді [50, 81]. Сонымен қатар зерттеуде GenAI құралдарын білімді бақылауда қолданудың бірқатар шектеулері де қарастырылған. Атап айтқанда, генеративті жасанды интеллект ұсынған жауаптар мен ұсыныстардың әрдайым дұрыс болмауы мүмкін екендігі және осыған байланысты оларды оқытушы тарапынан тексеру қажеттілігі атап көрсетілген [50].

GenAI құралдарының программалаудан білімді бақылаудағы мүмкіндіктерінің бірі студентке автоматты және жекелендірілген кері байланыс берумен байланысты. Осы бағытта T. Phung және т.б. программалау тапсырмаларын орындау барысында студенттерге жекелендірілген кеңестер қалыптастыру тәсілін қарастырған [82]. Авторлар GPT-4 моделін оқытушы (tutor model) ретінде, ал GPT-3.5 моделін қалыптастырылған кеңестердің сапасын тексеретін студент моделі (student model) ретінде қолданған. Ұсынылған тәсіл студенттің программалық кодындағы қателерді ескере отырып, оларды түзетуге бағытталған автоматты кеңестер қалыптастыруға мүмкіндік береді [82].

Қарастырылған зерттеулер жасанды интеллектке негізделген тәсілдердің программалық кодты талдау және студентке мазмұнды, жекелендірілген кері байланыс қалыптастыру арқылы автоматтандырылған бақылаудың мүмкіндіктерін кеңейтетінін көрсетеді. Сонымен қатар GenAI қалыптастыратын жауаптар мен ұсыныстарды тексеру қажеттілігі оларды білімді бақылауда қолдануда оқытушының қатысуының маңызын сақтайды.

Жүргізілген талдау программалаудан білімді бақылауда қолданылатын тәсілдердің бақылау нысаны, алынатын деректер және студенттің білімі мен программалау әрекетінің анықталатын аспектілері бойынша ерекшеленетінін көрсетеді. Дәстүрлі бақылау формаларынан автоматтандырылған және жасанды

интеллектке негізделген тәсілдерге дейінгі даму бақылау барысында алынатын деректер мен оларды талдау мүмкіндіктерінің біртіндеп кеңеюімен сипатталады. Осы ерекшеліктерді жүйелеу мақсатында қарастырылған тәсілдердің салыстырмалы сипаттамасы 1-кестеде берілді.

Кесте 1 – Программалаудан білімді бақылаудың қазіргі тәсілдері мен бағыттарының салыстырмалы сипаттамасы

Бақылау тәсілдері мен бағыттары	Бақылау нысаны	Алынатын бақылау деректері	Бақылауда анықталатын білім мен программалау әрекетінің аспектілері
Дәстүрлі бақылау	Теориялық білім, тапсырманы орындау нәтижесі	Жауаптар, балл, орындалған тапсырма	Теориялық білім мен жалпы оқу нәтижесі
Практикалық программалау тапсырмасы	Құрылған программа	Программаның орындалу нәтижесі, код	Программаны құру және іске асыру нәтижесі
Автоматтандырылған бағалау	Программалық код және оның орындалуы	Test case нәтижелері, күтілетін және нақты нәтиже, статикалық көрсеткіштер	Функционалдық дұрыстық және кодтың формалды сипаттамалары
Үдеріске бағытталған бақылау [69]	Тапсырманы орындау үдерісі	Компиляциялар, қателер, код күйінің өзгерістері	Программалау әрекетінің динамикасы
Debugging-ке бағытталған бақылау [71]	Қатені анықтау және түзету әрекеті	Қателерді табу және түзету нәтижелері	Қателермен жұмыс істеу әрекеті
Кодты түсіндіруге бағытталған бақылау [72]	Студенттің өзі құрған программа	Код құрылымы мен орындалу логикасына берілген түсіндірмелер	Өз программалық шешімін түсінуі
Peer code review [74]	Өзінің және басқа студенттің программалық коды	Рецензиялау, ұсыныстар, кодты қайта өңдеу нәтижелері	Кодты талдау, бағалау және жетілдіру әрекеті
ML/AI негізіндегі тәсілдер [79]	Студенттің программалық шешімінің жекелеген бөліктері	Шешімнің нақты бөліктерімен байланыстырылған автоматты кері байланыс және студенттің қате түсініктері туралы ақпарат	Студент шешіміндегі қателер мен қате түсініктерді анықтау және оларға бағытталған кері байланыс
LLM/GenAI негізіндегі тәсілдер [50, 79]	Программалық код және ондағы қателер	Қателерді түсіндіру, ұсыныстар, жекелендірілген кері	Программалық кодты талдау және мазмұнды,

1-кестенің жалғасы

Бақылау тәсілдері мен бағыттары	Бақылау нысаны	Алынатын бақылау деректері	Бақылауда анықталатын білім мен программалау әрекетінің аспектілері
		байланыс	жекелендірілген кері байланыс

Кестеде жүйеленген тәсілдер программалаудан білімді бақылауда алынатын деректердің әртүрлі сипатта екенін көрсетеді. Олардың бір бөлігі программалық нәтижені және кодтың сипаттамаларын анықтауға бағытталса, екіншілері тапсырманы орындау үдерісі, қателерді анықтау мен түзету, программалық шешімді түсіну және кодты талдау әрекеттері туралы ақпарат береді. Демек, қазіргі бақылау тәсілдері студенттің программалаудан білімі мен әрекетінің әртүрлі қырларын анықтауға мүмкіндік береді.

Сонымен бірге студенттің программалаудан білімі туралы әртүрлі бақылау тәсілдері арқылы алынатын ақпарат біртұтас емес. Программаның функционалдық дұрыстығы мен кодтың сипаттамалары программалық нәтижені сипаттаса, тапсырманы орындау барысы, қателерді анықтау және түзету студенттің программалау әрекеті туралы ақпарат береді. Ал студенттің өзі құрған программаның құрылымы мен орындалу логикасын түсіндіруі оның қабылдаған шешімін қаншалықты түсінетінін анықтауға мүмкіндік береді [21, 22, 23, 24, 26, 27, 42, 43, 45]. Осылайша, әрбір бақылау тәсілі студенттің программалаудан білімінің белгілі бір жағын анықтағанымен, алынған нәтижелердің өзара байланысы толық көрінбеуі мүмкін. Бұл программалаудан білімді бақылауда әртүрлі тәсілдер арқылы алынған нәтижелерді өзара байланысты қарастыру мәселесін алға шығарады.

Аталған мәселе программалау тапсырмасының соңғы нәтижесіне бағдарланған бақылауда айқын көрінеді. Программаның дұрыс орындалуы студенттің есепті шешу тәсілін, құрылған алгоритмнің логикасын және оны программалық түрде іске асыруды қаншалықты түсінетінін толық сипаттай бермейді. Бұған жоғарыда қарастырылған T. Lehtinen және т.б. зерттеуінің нәтижелері дәлел бола алады: дұрыс программа тапсырған студенттердің бір бөлігінің өз кодының құрылымы мен орындалуын түсіндіруде қиындықтарға кездесуі программаның функционалдық дұрыстығы мен студенттің программалық шешімді түсіну деңгейі әрдайым сәйкес келе бермейтінін көрсетті [72]. Сондықтан программаның функционалдық дұрыстығын студенттің программалаудан білімін сипаттайтын маңызды нәтиже ретінде қарастырғанымен, оны білімді бақылаудың жалғыз негізі ретінде қабылдау жеткіліксіз.

Соңғы нәтижеге бағдарланумен байланысты келесі мәселе студенттің программалау тапсырмасын орындау барысында жүзеге асыратын әрекеттерінің жеткілікті ескерілмеуінен көрінеді. Программалау барысында студент есепті шешу тәсілін таңдайды, алгоритмді программалық түрде іске асырады, кодты тексереді, анықталған қателерді түзетеді және қажет болған жағдайда алдыңғы әрекеттеріне қайта оралады. Жоғарыда қарастырылған

зерттеулер программалау әрекетінің осындай итерациялық сипатын және тапсырманы орындау барысында қалыптасатын деректердің студенттің программалау әрекеті туралы қосымша ақпарат беретінін көрсетті [69, 71]. Алайда программаның соңғы нәтижесін тексеруге бағытталған бақылауда бұл әрекеттердің қалай жүзеге асырылғаны толық көрінбеуі мүмкін. Сондықтан программалаудан білімді бақылауда соңғы нәтижемен қатар, оған қол жеткізу барысында жүзеге асырылатын программалау әрекетін де ескеру мәселесі туындайды.

Демек, әртекті бақылау деректерін студенттің біртұтас білім жағдайын сипаттайтындай өзара байланыстыру қажеттілігі сақталады

Қазіргі программалаудан білімді бақылау мәселесі генеративті жасанды интеллект құралдарының студенттердің программалау тапсырмаларын орындау үдерісіне енуімен жаңа мазмұнға ие болды [83, 84]. Жасанды интеллект технологияларын білім беру мен ғылымда дамыту бағыттары, сондай-ақ оларды пайдаланудағы ашықтық пен жауапкершілік талаптары Қазақстан Республикасының стратегиялық және нормативтік құжаттарында да айқындалған [5, 6, 7]. GenAI студентке программалық кодты генерациялау, қатені анықтау және түзету, алгоритмдік шешім ұсыну сияқты әрекеттерде көмек көрсете алатындықтан, студент ұсынған программалық нәтиже оның дербес программалау әрекетінің нәтижесі ретінде ғана қарастырыла алмайды. Осы жағдайда бақылаудың маңызды мәселелерінің бірі программалау тапсырмасын орындаудағы студенттің дербес интеллектуалдық үлесін анықтаумен байланысты.

Бұл мәселенің когнитивтік қыры GenAI құралдарын қолдану барысында студенттің интеллектуалдық әрекеттерінің өзгеруімен байланысты. Liu, Fan және Pan [85]. GenAI-ды қолдану бір жағдайда білім алушының ойлау әрекетін қолдайтын cognitive scaffolding қызметін атқарса, екінші жағдайда студент өзі жүзеге асыруы тиіс концептуалдық және бағалау әрекеттерінің бір бөлігін сыртқы интеллектуалдық құралға беретін cognitive offloading құбылысына әкелуі мүмкін екенін көрсетеді. Осы тұрғыдан студенттің тапсырманы GenAI көмегімен нәтижелі орындауы оның тиісті білім мен әрекеттерді жаңа жағдайда өздігінен жүзеге асыра алатынын толық сипаттай бермейді [85].

Осы жағдай программалаудан білімді бағалаудың валидтілігі мәселесін алға шығарады. Егер тапсырманы орындауға қажетті есепті талдау, алгоритмдік шешімді құру, программалық кодты жазу немесе оны тексеру сияқты әрекеттердің бір бөлігін GenAI жүзеге асырса, алынған программалық нәтиженің студенттің дербес білімін қаншалықты дәл сипаттайтыны туралы мәселе туындайды. Аталған мәселе педагогикалық бағалаудың валидтілігі тұрғысынан ерекше мәнге ие. GenAI көмегімен алынған программалық өнімнің сапасы мен студенттің өзінің программалау білімінің деңгейі өзара байланысты болғанымен, оларды бірдей көрсеткіш ретінде қарастыруға болмайды. Сондықтан GenAI қолданылатын жағдайда бақылаудың негізгі міндеті программалық өнімнің дұрыстығын ғана анықтау емес, сол өнім негізінде студенттің қандай білім компоненттері туралы сенімді педагогикалық қорытынды жасауға болатынын белгілеу болып табылады. Бұл программалау нәтижесімен қатар студенттің шешу әрекетін, кодты түсінуін, негіздеуін және қателермен жұмысын сипаттайтын қосымша деректерді ескеруді қажет етеді.

Бұл мәселенің программалауды оқытудағы көрінісі D. Kohen-Vacs, M. Usher және M. Jansen зерттеуінде байқалады [52]. Авторлар студенттердің LLM генерациялаған қате программалық кодты түзету нәтижелерін оқытушы әзірлеген дәстүрлі программалау тапсырмаларын орындау нәтижелерімен салыстырған. Зерттеу студенттердің LLM генерациялаған кодтағы қателерді түзетуде айтарлықтай қиындықтарға кездескенін көрсетті [52]. Бұл нәтиже GenAI көмегімен алынған программалық нәтиже мен студенттің оны өз білімі негізінде талдап, тексере және түзете алу қабілетін ажырата қарастыру қажеттігін көрсетеді.

Осылайша, GenAI құралдарының программалау тапсырмаларын орындауда қолданылуы программалаудан білімді бақылауда студент ұсынған программалық нәтиже мен оның дербес білімі арасындағы байланысты анықтау мәселесін күшейтеді. Бұл жағдайда программаның дұрыс орындалуы студенттің тапсырманы орындауға қажетті әрекеттерді өздігінен жүзеге асырғанын толық көрсете алмайды. Сондықтан GenAI жағдайында программалаудан білімді бақылауда студенттің алған нәтижесімен қатар, есепті шешу барысында жүзеге асырған дербес программалау әрекеттерін, GenAI көмегін пайдалану сипатын және алынған шешімді өз білімі негізінде талдау, тексеру мен негіздеу қабілетін ескеру қажеттілігі туындайды.

Жоғарыда анықталған мәселелерді жинақтау программалаудан білімді бақылаудың қазіргі жағдайына тән бірқатар өзара байланысты қайшылықтарды бөліп көрсетуге мүмкіндік береді.

Біріншіден, программалаудан білім есепті талдау, алгоритмдік шешімді құру, оны программалық іске асыру, тестілеу, қателерді анықтау мен түзету және алынған шешімді түсіндіру сияқты өзара байланысты әрекеттерде көрінетін көпқырлы сипатқа ие. Сонымен бірге қазіргі бақылау тәсілдері осы әрекеттер мен олардың нәтижелерінің жекелеген жақтарын анықтауға бағытталған. Бұл программалаудан білімнің көпқырлы сипаты мен оны жекелеген бақылау тәсілдері арқылы анықтау арасындағы қайшылықты айқындайды.

Екіншіден, программалау тапсырмасының орындалу нәтижесі студенттің программалаудан білімін бақылаудың маңызды көзі болғанымен, оның сол нәтижеге қол жеткізу барысында жүзеге асырған әрекеттерін толық көрсете бермейді. Бұл мәселе GenAI жағдайында одан әрі күрделене түседі, өйткені программалық нәтижені қалыптастыруға студентпен қатар жасанды интеллект құралдары да қатысуы мүмкін. Мұндай жағдайда программалау тапсырмасының орындалу нәтижесін бағалау мен студенттің сол нәтижеге қол жеткізудегі дербес интеллектуалдық үлесін анықтау қажеттілігі арасында қайшылық пайда болады.

Үшіншіден, автоматтандырылған бағалау, машиналық оқыту және GenAI негізіндегі тәсілдердің дамуы программалық код пен студенттің программалау әрекеті туралы әртүрлі деректерді автоматты түрде алу және талдау мүмкіндіктерін кеңейтуде. Сонымен қатар автоматтандыру деңгейінің артуы бақылау нәтижелерінің дұрыстығын, сенімділігін және студенттің нақты білімін қаншалықты көрсететінін қамтамасыз ету қажеттілігін жоймайды. Нәтижесінде программалаудан білімді бақылауды автоматтандыру

мүмкіндіктерінің кеңеюі мен алынған бақылау нәтижелерінің сенімділігі мен валидтілігін қамтамасыз ету қажеттілігі арасындағы қайшылық алға шығады.

Төртіншіден, программалаудан білімді бақылаудың қорытынды нәтижесін бір балл немесе баға түрінде көрсету студенттің жалпы оқу нәтижесін сипаттауға мүмкіндік береді, алайда бірдей қорытынды баға алған студенттердің тапсырманы орындау барысы, жіберген қателері және кездесетін қиындықтары әртүрлі болуы мүмкін. Сондықтан қорытынды баға студенттің программалаудан білімінің қай жақтары жеткілікті меңгерілгенін және қай бағыттарда қиындықтар бар екенін толық көрсете алмайды. Бұл студенттің білімін қорытынды көрсеткіш арқылы сипаттау мен оның программалаудан білімін диагностикалық тұрғыдан неғұрлым толық анықтау қажеттілігі арасындағы қайшылықты көрсетеді.

Анықталған қайшылықтарды жинақтау программалаудан білімді бақылаудың қазіргі тәсілдері студенттің программалау әрекеті мен оның нәтижелері туралы әртүрлі деректер алуға мүмкіндік бергенімен, осы деректер негізінде студенттің программалаудан білімін тұтас бағалау мәселесінің толық шешілмегенін көрсетеді.

1.2 Интеллектуалдық оқу құралдарын қолданып программалаудан білімді бақылаудың ерекшеліктері

Ақпараттық-коммуникациялық технологиялардың білім беру үдерісіне енгізілуі бақылау тапсырмаларын ұсыну, студент жауаптарын тіркеу және нәтижелерді өңдеу мүмкіндіктерін кеңейтті. Автоматтандырылған бақылауда жауаптар алдын ала белгіленген ережелер немесе эталондар бойынша тексеріліп, олардың негізінде балл немесе тапсырманың орындалуы туралы қорытынды қалыптастырылады. Программалаудан білімді бақылауда мұндай тексеру студент кодының берілген тестілік жағдайларда күтілетін нәтиже шығаруын және белгіленген техникалық талаптарды орындауын анықтауға мүмкіндік береді. Алайда программаның дұрыс орындалуы студенттің қандай білім компоненттерін меңгергенін, алгоритмдік шешімнің логикасын қаншалықты түсінетінін және тапсырманы орындау барысында қандай қиындықтарға кездескенін толық сипаттай бермейді. Сондықтан программалаудан білімді бақылауда тек тапсырманың орындалу нәтижесін тіркеумен шектелмей, жинақталған бақылау деректерін студенттің білімді меңгеру жағдайымен байланыстыра талдау қажеттілігі туындайды. Бұл қажеттілік білімді бақылауды интеллектуалдандырудың негізгі алғышарттарының бірі болып табылады.

Білімді бақылауды ақпараттандыру құралдарының мүмкіндіктері біздің алдыңғы зерттеулерімізде де қарастырылды. Білімді бақылауға арналған программалық құралдарды салыстырмалы талдау барысында олардың тест тапсырмаларын құрастыру, нәтижелерді автоматты бағалау, жинақтау және өңдеу мүмкіндіктері анықталды. Сонымен қатар мұндай құралдардың сервистік және функционалдық мүмкіндіктерін пайдалану білімді бақылаудың пәндік ерекшеліктерін ескере отырып, мазмұндық тұрғыдан ұйымдастырылуы қажеттігі көрсетілді [86]. Бұл тұжырым білімді бақылауды ақпараттандыруда

тек техникалық және функционалдық мүмкіндіктермен шектелмей, бақылау нәтижелерін пәндік білім мазмұнымен байланыстыру қажеттігін көрсетті.

Компьютерлік бақылауды ұйымдастыру мәселесі авторлық зерттеулерде оның педагогикалық функциялары тұрғысынан да қарастырылды. Ә.Е. Жақсылықовтың зерттеуінде компьютерлік бақылаудың болжамдық, оқытушылық, диагностикалық және бақылаушылық функциялары ажыратылып, бақылау нәтижелері негізінде білім алушының біліміндегі қателер мен олқылықтарды анықтау, оның дайындық деңгейін белгілеу және кейінгі оқу әрекетін ұйымдастыру мүмкіндіктері қарастырылған [87]. Сонымен қатар білім алушының анықталған дайындық деңгейіне сәйкес әртүрлі күрделіліктегі тапсырмаларды ұсыну мүмкіндігі сипатталған.

Бақылау нәтижелерін білім мазмұны және оны меңгеру деңгейлерімен байланыстыру мәселесі бұған дейінгі авторлық зерттеулерде көпдеңгейлі және семантикалық ақпараттық модельдер негізінде қарастырылған. А.Е. Сағымбаева мен Ә. Жақсылықовтың еңбектерінде компьютерлік бақылауды ұйымдастыруда оқу материалының құрылымы, білім элементтерінің өзара байланысы және оларды меңгеру деңгейлерін ескеру қажеттілігі негізделіп, көпдеңгейлі семантикалық модельді компьютерлік жүйеде іске асыру жолдары ұсынылған [88–91]. Бұл зерттеулер бақылау нәтижесін жеке тапсырманың орындалуымен шектемей, оны меңгерілуі тиіс білім мазмұнымен байланыстыра қарастырудың маңызын көрсетті.

Кейінгі авторлық зерттеуде бұл бағыт одан әрі дамытылып, көпдеңгейлі ақпараттық модельді компьютерлік жүйеде іске асыру жолдары қарастырылды [92]. Модельді компьютерлік іске асыруда пәндік білім мазмұны семантикалық ұғымдар мен олардың арасындағы байланыстар арқылы құрылымдалып, тест тапсырмалары тиісті деңгейлермен сәйкестендірілді. Білім алушының жауабына байланысты келесі тапсырманы таңдау алгоритмінің қолданылуы бақылауды білімді меңгеру деңгейлерімен байланыстыра ұйымдастыруға мүмкіндік берді [55].

Ал бақылау нәтижелерін білім мазмұнымен байланыстыру идеясы қазіргі интеллектуалдық оқу құралдарында білім алушының оқу әрекеттерінен алынатын деректерді талдау және оның білім жағдайын модельдеу негізінде дамуда. Мұндай құралдардың интеллектуалдық мүмкіндіктерінің бірі білім алушы туралы жинақталған деректер негізінде оның білім жағдайын сипаттайтын студент моделін (student model) қалыптастырумен байланысты. Студентті модельдеу (student modeling) тәсілдері интеллектуалдық оқыту жүйелерінде (Intelligent Tutoring Systems – ITS) қолданылып, қазіргі кезде бейімделген және интеллектуалдық білім беру орталарында білім алушының оқу әрекеттерін талдау, білім жағдайын анықтау және оқытуды дербестендіру мақсатында дамып келе жатыр [93–95].

Студент моделін қалыптастыруда оның білім жағдайының уақыт бойынша өзгерісін бағалауға мүмкіндік беретін тәсілдердің бірі – Knowledge Tracing. Мысалы, Corbett және Anderson ұсынған Bayesian Knowledge Tracing моделінде студенттің белгілі бір білім компонентіне қатысты тапсырмаларға дұрыс немесе қате жауап бергені ескеріледі. Осы жауаптар негізінде оның тиісті білімді меңгеру ықтималдығы есептеліп, әрбір жаңа жауаптан кейін бұл баға жаңартылады [47]. Ал Piech және т.б. ұсынған Deep Knowledge Tracing

моделінде студенттің алдыңғы тапсырмаларды қандай ретпен және қандай нәтижемен орындағаны нейрондық желі арқылы талданады. Жинақталған деректер негізінде модель студенттің келесі тапсырмаға дұрыс жауап беру ықтималдығын болжайды [48]. Бұл модельдер бақылау нәтижелерін жеке жауапқа баға қою үшін ғана емес, студенттің білімді меңгеру барысын бағалау үшін пайдалануға мүмкіндік береді. Алайда модельдік болжам студенттің білімін тікелей сипаттамайтындықтан, оны тапсырмаларда тексерілетін білім компоненттері және пайдаланылған деректермен байланыстыра педагогикалық тұрғыдан түсіндіру қажет.

Педагогикалық бақылау тұрғысынан мұндай модельдік талдау студенттің оқу әрекеттерінен алынған деректер негізінде оның білім жағдайын анықтауға бағытталады [96, 97]. Алайда модель есептеген ықтималдық немесе болжам өздігінен педагогикалық бақылаудың нәтижесі болып табылмайды. Оның мәні пайдаланылған деректердің қандай білім компонентін сипаттайтыны және оқу мақсаттарымен қаншалықты байланысты екені арқылы анықталады. Сондықтан интеллектуалдық бақылауда деректерді модельдік өңдеумен қатар, алынған нәтижелерді студенттің білімді меңгеру жағдайы тұрғысынан педагогикалық интерпретациялау маңызды.

Осылайша, автоматтандырылған бақылау мен интеллектуалдық бақылаудың айырмашылығы тек қолданылатын технологияларда емес, бақылау деректерін өңдеу және алынған нәтижелерді түсіндіру сипатында көрінеді. Автоматтандырылған бақылау студенттің жауабын немесе программалық кодын алдын ала белгіленген ережелер мен критерийлер бойынша тексеруге бағытталса, интеллектуалдық бақылау жинақталған деректер арасындағы байланыстарды модельдік талдау арқылы студенттің білім жағдайы туралы диагностикалық ақпарат қалыптастыруға мүмкіндік береді. Бұл ретте интеллектуалдық бақылау автоматтандырылған бақылауды алмастырмайды, керісінше оның нәтижелерін студенттің білім жағдайын анықтауға бағытталған модельдік өңдеу және педагогикалық интерпретациямен толықтырады.

Автоматтандырылған және интеллектуалдық бақылаудың аталған айырмашылықтарын жүйелеу мақсатында олардың негізгі сипаттамалары 2-кестеде салыстырылды.

Кесте 2 – Автоматтандырылған және интеллектуалдық бақылаудың сипаттамалары

Салыстыру негізі	Автоматтандырылған бақылау	Интеллектуалдық бақылау
Негізгі мақсат	Жауаптың немесе орындалған тапсырманың белгіленген талаптарға сәйкестігін автоматты тексеру	Студенттің білімді меңгеру жағдайы туралы диагностикалық ақпарат қалыптастыру
Бастапқы деректер	Студент жауабы, программалық код, тест нәтижелері және басқа тіркелетін көрсеткіштер	Жауаптар, код, әрекеттер тарихы және олардың тапсырма мазмұны мен білім компоненттеріне қатысы
Өңдеу сипаты	Алдын ала белгіленген ереже, алгоритм немесе эталон бойынша тексеру	Деректер арасындағы байланыстарды модель арқылы талдау және нәтижелерді

2-кестенің жалғасы

Салыстыру негізі	Автоматтандырылған бақылау	Интеллектуалдық бақылау
		педагогикалық тұрғыдан түсіндіру
Уақыттық контекст	Жеке нәтижелерді тексеру, бірнеше нәтиженің статистикасын жинақтау	Алдыңғы және кейінгі нәтижелерді байланыстырып, меңгеру жағдайындағы өзгерістерді бағалау
Қалыптасатын нәтиже	Дұрыс немесе қате жауап, балл, тесттердің орындалуы туралы мәлімет	Меңгеру жағдайының модельдік бағасы, қиындықтар туралы ақпарат және кейінгі нәтижеге қатысты болжам
Педагогикалық қолданылуы	Нәтижені тіркеу, жедел тексеру және алдын ала белгіленген кері байланыс беру	Қиындықтарды анықтау, кері байланысты бейімдеу және оқытушының шешімін ақпаратпен қамтамасыз ету
Білім мазмұнымен байланысы	Тапсырма талаптары, эталондар және тексеру ережелері арқылы беріледі	Талдау нәтижелері бағаланатын білім компоненттерімен, меңгеру деңгейлерімен және оқу мақсатымен байланыстырылады

2-кестеде көрсетілгендей, автоматтандырылған бақылауда негізгі назар тапсырма нәтижесін белгіленген ережелер бойынша тексеруге аударылса, интеллектуалдық бақылауда студенттің әртүрлі оқу әрекеттерінен алынған деректерді байланыстыра талдау басым болады. Соның нәтижесінде бақылау нәтижесі дұрыс/қате немесе балл түріндегі көрсеткішпен шектелмей, студенттің білім жағдайын сипаттайтын диагностикалық ақпаратпен толықтырылады.

Жүргізілген талдау негізінде осы зерттеуде интеллектуалдық бақылау ұғымын мынадай мағынада қолдану ұсынылады:

Интеллектуалдық бақылау – білім алушының оқу нәтижелері мен әрекеттерінен алынатын деректерді пәндік білім құрылымымен байланыстыра модельдік өңдеу және талдау, олардың арасындағы мәнді байланыстарды анықтау және осы негізде білім алушының білім жағдайы туралы диагностикалық тұрғыдан интерпретацияланатын ақпарат қалыптастыру үдерісі.

Осы анықтама интеллектуалдық бақылаудың автоматты тексеру нәтижелерін модельдік талдау және педагогикалық интерпретациямен толықтыратынын нақтылайды

Жоғарыда қарастырылғандай, интеллектуалдық бақылау білім алушының оқу нәтижелері мен әрекеттерінен алынатын әртүрлі деректерді оның білім жағдайымен байланыстыра талдауды көздейді. Программалаудан білімді бақылауда мұндай деректердің маңызды бөлігі студенттің программалық тапсырмаларды орындау барысында қалыптасады. Оларға программалық код, тестілік жағдайлардан өту нәтижелері, жіберілген қателер, программаның орындалу уақыты мен жадыны пайдалану көрсеткіштері және басқа да орындау сипаттамалары жатады. Бұл деректерді автоматты түрде жинау мен өңдеу мүмкіндігі программалауға арналған цифрлық платформаларда жүзеге

асырылады. Сондықтан олардың бақылаудың қандай міндеттерін орындайтынын және қалыптастыратын деректерінің студенттің білім жағдайын диагностикалауға қаншалықты жеткілікті екенін анықтау қажет.

Қазіргі программалау практикасында программалық есептерді шешу және олардың нәтижелерін автоматты тексеру үшін LeetCode, Codeforces, HackerRank сияқты цифрлық платформалар қолданылады. Олардың бақылау мүмкіндіктерін анықтау үшін әр платформаның тапсырмаларды ұйымдастыру, программалық шешімді тексеру, орындалу нәтижелерін тіркеу және кері байланыс қалыптастыру ерекшеліктерін қарастырайық.

LeetCode программалық есептерді әртүрлі күрделілік деңгейлері бойынша ұсынуға және олардың шешімдерін тестілік жағдайлар негізінде автоматты тексеруге мүмкіндік беретін платформалардың бірі болып табылады. Біздің алдыңғы зерттеуімізде LeetCode платформасы әртүрлі күрделіліктегі программалау есептерін шешу ортасы ретінде қолданылып, онда генеративті жасанды интеллект модельдерінің программалық шешімдері салыстырмалы түрде талданды [76]. Зерттеу барысында әртүрлі күрделілік деңгейіндегі 100-ге жуық есеп қарастырылып, программалық шешімдердің алгоритмді іске асыру ерекшеліктері, жіберілген қателері және тиімділігі бағаланды.

Платформадағы автоматты тексеру программалық шешімнің тестілік жағдайлардан өту нәтижесін тіркеп, Accepted, Compilation Error, Wrong Answer, Time Limit Exceeded, Memory Limit Exceeded, Runtime Error және басқа да нәтижелер арқылы оның функционалдық және техникалық сипаттамаларын анықтауға мүмкіндік береді [76].

Codeforces платформасы алгоритмдік программалау есептерін орындауға және ұсынылған программалық шешімдерді автоматты түрде тексеруге мүмкіндік береді. Платформада әрбір есеп үшін орындалу уақыты мен жады көлеміне қойылатын шектеулер белгіленіп, жіберілген программалық код тестілер жиынтығы бойынша тексеріледі [98, 99]. Тексеру нәтижесінде шешімнің қабылдануы немесе анықталған қате түрі, сондай-ақ программаның орындалу уақыты мен пайдаланылған жады көлемі тіркеледі. Сонымен қатар жіберілген шешімдер тарихының сақталуы тапсырманы орындау барысында қалыптасқан нәтижелерді бақылауға мүмкіндік береді. Зерттеулерде Codeforces программалауды оқыту мүмкіндіктері бар платформалардың қатарына жатқызылғанымен, ондағы программалық шешімді автоматты бағалау негізінен кіріс-шығыс тестілеріне сүйенетіні көрсетіледі [100].

HackerRank платформасы программалау тапсырмаларын орындау және олардың нәтижелерін автоматты бағалау мүмкіндігін ұсынады. Программалық шешімді тексеру алдын ала анықталған тестілік жағдайлар негізінде жүзеге асырылады: студенттің коды орындалып, алынған нәтиже күтілетін нәтижемен салыстырылады. Платформада ашық және жасырын тестілік жағдайларды (sample және hidden test cases) қолдануға, әрбір тест үшін балл белгілеуге және орындалған тестілер негізінде жалпы нәтижені автоматты есептеуге болады [101]. Сонымен қатар тексеру нәтижелерінде тестінің орындалу мәртебесі, алынған балл, орындалу уақыты және пайдаланылған жады туралы деректер тіркеледі [102]. Бұл мүмкіндіктер программалық шешімнің функционалдық дұрыстығы мен орындалу тиімділігін автоматты бақылауға мүмкіндік береді.

Программалаудан білімді бақылауды жоғары оқу орнының оқу үдерісінде ұйымдастыруға бағытталған құралдардың бірі – Moodle ортасына кіріктірілетін Virtual Programming Lab (VPL) модулі. VPL оқытушыға программалау тапсырмаларын құрастыруға, студенттің программалық кодын орындауға және алдын ала берілген тестілік жағдайлар негізінде автоматты бағалауға мүмкіндік береді [103]. Тестілік жағдайларда кіріс және күтілетін шығыс деректері, орындалу уақытына қойылатын шектеулер және қате орындалған тестілер үшін айыптау параметрлері белгіленіп, тексеру нәтижесінде студентке баға мен кері байланыс қалыптастырылады [104]. Сонымен қатар VPL Moodle-дың бағалау жүйесімен кіріктірілген және студентке программаны бірнеше рет орындау, тексеру және жетілдіру мүмкіндігін береді [103]. Осы ерекшеліктері VPL-ді программалық шешімді автоматты тексерумен қатар, оны оқу пәні шеңберінде ағымдық және қорытынды бақылауды ұйымдастыруға бейімдеуге мүмкіндік береді.

Қарастырылған платформалар программалық тапсырмаларды автоматты тексеруге бағытталғанымен, олардың қалыптастыратын бақылау деректері мен педагогикалық мүмкіндіктері бірдей емес. Сондықтан олардың программалаудан білімді бақылаудағы мүмкіндіктерін тек техникалық тексеру тұрғысынан ғана емес, студенттің әрекеті туралы деректерді жинау, кері байланыс ұсыну, дербестендіру және білім жағдайын диагностикалау мүмкіндіктері тұрғысынан салыстыру қажет. Осыған байланысты LeetCode, Codeforces, HackerRank және Moodle VPL платформаларының негізгі мүмкіндіктері 3-кестеде салыстырылды.

Кесте 3 – Программалаудан білімді бақылауға арналған цифрлық платформалардың мүмкіндіктерін салыстыру

Салыстыру критерийі	LeetCode	Codeforces	HackerRank	Moodle VPL
Тапсырма түрлері	Негізінен алгоритмдік программалау есептері	Алгоритмдік және жарыстық программалау есептері	Программалау, алгоритмдік және практикалық тапсырмалар	Оқытушы құрастыратын оқу тапсырмалары
Тестілік жағдайлар	Бар, автоматты тексерудің негізі	Бар, автоматты тексерудің негізі	Ашық және жасырын тестілер	Оқытушы құрастыратын тестілер
Функционалдық дұрыстық	Тестілерден өту арқылы анықталады	Тестілерден өту арқылы анықталады	Күтілетін нәтижемен салыстыру арқылы анықталады	Кіріс және күтілетін шығыс нәтижелерін салыстыру арқылы анықталады
Орындалу уақыты және жады	Ескеріледі	Ескеріледі	Тіркеледі	Орындалу уақытын шектеу мүмкіндігі бар

3-кестенің жалғасы

Салыстыру критерийі	LeetCode	Codeforces	HackerRank	Moodle VPL
Код сапасын бағалау	Шектеулі	Шектеулі	Жекелеген бағалау сценарийлерінде мүмкін, бірақ негізгі автоматты бағалау test cases-ке сүйенеді	Оқытушы белгілеген бағалау сценарийіне тәуелді
Орындау үдерісі туралы деректер	Жіберілген шешімдер мен нәтижелер туралы деректер бар	Жіберілімдер тарихы сақталады	Code Playback арқылы код жазу әрекетін қарауға мүмкіндік бар	Бірнеше жіберілім мен олардың нәтижелерін тіркеуге болады
Кері байланыс	Тест нәтижелері және қате түрлері	Тексеру нәтижесі және қате түрлері	Тест нәтижелері, балл және егжей-тегжейлі есеп	Автоматты бағалау есебі және оқытушы баптайтын кері байланыс
Дербестендіру	Шектеулі	Шектеулі	Бағалау параметрлерін баптау мүмкіндігі бар, бірақ білім моделіне негізделген дербестендіру емес	Оқытушы тапсырма мен бағалау шарттарын бейімдей алады
Студент моделін қалыптастыру	Негізгі функциясына жатпайды	Негізгі функциясына жатпайды	Негізгі функциясына жатпайды	Негізгі функциясына жатпайды
GenAI/LLM интеграциясы	Платформаның негізгі автоматты бағалау механизмі емес	Платформаның негізгі автоматты бағалау механизмі емес	GenAI тапсырмалары мен бағалау мүмкіндіктері бар	Негізгі VPL бағалау механизмінің құрамына кірмейді
Білім жағдайын диагностикалау мүмкіндігі	Жекелеген программалық шешімнің нәтижесі туралы дерек береді, бірақ білім жағдайының тұтас моделін қалыптастырмайды	Орындау нәтижелері мен жіберілімдер тарихын береді, бірақ білім жағдайын модельдемейді	Бағалау және әрекет деректері кеңірек, бірақ білім жағдайының динамикалық моделін қалыптастыру негізгі функциясы емес	Оқу пәні аясында бағалауды ұйымдастырады, бірақ студенттің білім жағдайын интеллектуалдық модельдеу негізгі функциясына жатпайды

Осылайша, қазіргі цифрлық платформалар программалық шешімнің функционалдық дұрыстығын, орындалу тиімділігін және жекелеген әрекет

нәтижелерін автоматты түрде тіркеуде тиімді. Алайда олардың негізгі мүмкіндіктері программалық өнімді тексеруге бағытталғандықтан, студенттің программалаудан білім жағдайын көпкомпонентті және динамикалық түрде сипаттау шектеулі. Бұл программалаудан білімді бақылауда платформалардан алынатын деректерді интеллектуалдық талдау құралдарымен толықтыру қажеттігін көрсетеді.

Интеллектуалдық оқу құралдарының программалаудан білімді бақылаудағы маңызды мүмкіндіктерінің бірі – студенттің программалық шешімін мазмұндық тұрғыдан талдау. Бұл жағдайда интеллектуалдық оқу құралдары программалық кодты, тестілеу нәтижелерін, жіберілген қателерді және программаның орындалу көрсеткіштерін бастапқы бақылау деректері ретінде пайдалана алады. Мұндай деректер программаның дұрыс немесе қате орындалуын анықтаумен қатар, студенттің тапсырманы орындау тәсілін талдауға негіз болады. Бұл деректерді интеллектуалдық өңдеуде жасанды интеллект және программалық кодты семантикалық талдау әдістері қолданылуда. Зерттеулерде студенттің программалық шешімін эталондық шешіммен программа құрылымы мен семантикалық ұқсастығы бойынша салыстыру арқылы автоматты бағалау мүмкіндігі көрсетілген [105]. Сонымен қатар LLM негізіндегі интеллектуалдық жүйелер студент кодын логикалық дұрыстық, кодтау стилі және алгоритмдік тиімділік тұрғысынан талдап, алынған нәтижелер негізінде бейімделген кері байланыс қалыптастыра алады [106]. Жасанды интеллект құралдарының кодты талдау, оны жетілдіруге қатысты ұсыныстар беру және жедел кері байланыс қалыптастыру мүмкіндіктері программалаудан білімді бақылау контексінде де қарастырылған [51]. Осылайша, интеллектуалдық талдау программалық шешімнің орындалу нәтижесімен қатар, оның құрылымдық және мазмұндық ерекшеліктері туралы қосымша бақылау ақпаратын алуға мүмкіндік береді.

Мысалы, Code-DKT моделінде студенттің тапсырмаларды орындау тарихы программалық кодтан автоматты түрде алынатын белгілермен толықтырылып, кейінгі тапсырмаларды орындау нәтижесін болжау үшін пайдаланылады [49]. Бұл программалаудан білімді интеллектуалдық бақылауда студенттің жекелеген тапсырмаларды орындау нәтижелерімен қатар, программалық кодтан алынатын деректерді де өзара байланыстыра пайдаланып, оның білім жағдайындағы өзгерістерді бақылау мүмкіндігін көрсетеді.

Интеллектуалдық оқу құралдарының программалаудан білімді бақылаудағы маңызды мүмкіндіктерінің бірі – бақылау нәтижелері негізінде мазмұндық кері байланыс қалыптастыру [107]. Дәстүрлі автоматты тексеру жүйелерінде кері байланыс көбіне программаның дұрыс немесе қате орындалуы, тестілік жағдайлардан өтуі немесе қате түрі туралы хабарламамен шектелсе, жасанды интеллект негізіндегі құралдар программалық код пен анықталған қателерді талдау арқылы олардың ықтимал себептерін түсіндіруге, кодты жетілдіруге бағытталған ұсыныстар беруге және кері байланысты студенттің нақты шешіміне бейімдеуге мүмкіндік береді [51, 106].

GenAI негізіндегі кері байланыстың ерекшелігі оның студенттің нақты программалық шешімінің контексін ескере отырып қалыптастырылуында. Zhang және т.б. зерттеуінде программалау тапсырмалары бойынша LLM көмегімен қалыптастырылған кері байланыс қарастырылып, студенттер

өздерінің программалық коды ескерілген кері байланысты неғұрлым нақты, түсінікті және қатені түзетуге бағытталған деп бағалаған [53]. Бұл нәтижелер GenAI құралдарының программалаудан білімді бақылауда жалпы сипаттағы хабарламамен шектелмей, студенттің нақты шешіміне қатысты мазмұндық кері байланыс қалыптастыру мүмкіндігін көрсетеді.

Сонымен қатар GenAI негізінде қалыптастырылатын кері байланыстың дұрыстығы мен сенімділігі білімді бақылауда ерекше назар аударуды қажет етеді. Silva және Costa төрт LLM моделінің студенттердің программалық шешімдеріндегі қателерді анықтап, кері байланыс қалыптастыру мүмкіндігін зерттеген. Зерттеу нәтижесінде қалыптастырылған кері байланыстардың 63%-ы дұрыс және толық болғанымен, 37%-ында қате жолды көрсету, дұрыс емес түсіндіру немесе кодта жоқ мәселені көрсету сияқты кемшіліктер анықталған [54]. Бұл нәтижелер LLM қалыптастырған кері байланысты білімді бақылаудың дайын нәтижесі ретінде тікелей қабылдауға болмайтынын көрсетеді.

GenAI қалыптастырған кері байланыстың сенімділігімен қатар, оның студенттің нақты білім жағдайын қаншалықты сипаттайтыны да маңызды. ЖИ ұсынған түсіндірме немесе программалық шешім студенттің оны өздігінен орындай алатынын тікелей білдірмейді, өйткені студент дайын кодты немесе ұсынысты өзінің дербес программалау әрекетінсіз де пайдалана алады [51]. Сондықтан GenAI құралдарынан алынған ақпаратты студент білімінің тікелей көрсеткіші ретінде емес, оның нақты оқу әрекеттері мен басқа бақылау нәтижелерімен байланыстыра педагогикалық тұрғыдан интерпретациялау қажет.

GenAI мазмұндық кері байланысты кеңейтеді, бірақ оның сенімділігі тұрақсыз болғандықтан, нәтижелер басқа бақылау деректерімен бірге педагогикалық тұрғыдан тексерілуі тиіс

Жүргізілген талдаудан интеллектуалдық оқу құралдарын қолданып программалаудан білімді бақылаудың ерекшеліктерінің бірі бақылауда пайдаланылатын деректер құрамының кеңеюі екені көрінеді. Дәстүрлі автоматтандырылған бақылауда негізгі деректер тапсырманың орындалу нәтижесі, тестілік жағдайлардан өтуі және алынған балл түрінде қалыптасса, интеллектуалдық құралдарда оларға студенттің программалық коды, кодтың құрылымдық және мазмұндық сипаттамалары, жіберілген қателер, бірнеше тапсырма бойынша жинақталған әрекеттер тарихы сияқты деректер қосылады [105, 106]. Осылайша, программалаудан білімді интеллектуалдық бақылау бір ғана қорытынды нәтижеге емес, студенттің программалау әрекетін әртүрлі қырынан сипаттайтын деректер жиынтығына сүйенеді.

Екінші ерекшелік бақылау деректерін өзара байланыстыра және мазмұндық тұрғыдан талдаумен сипатталады. Интеллектуалдық оқу құралдарында программалық код пен тапсырманы орындау нәтижелері бір-бірінен оқшау көрсеткіштер ретінде ғана қарастырылмай, олардың арасындағы байланыстарды анықтау үшін өңделуі мүмкін. Мысалы, программалық шешімнің құрылымдық және семантикалық сипаттамаларын эталондық шешіммен салыстыру студент кодын тек орындалу нәтижесі бойынша емес, оның мазмұны бойынша бағалауға мүмкіндік береді [105], ал Code-DKT моделінде тапсырмаларды орындау тарихы кодтан алынатын белгілермен байланыстырылып, кейінгі нәтижені болжау үшін пайдаланылады [49]. Осы

тұрғыдан интеллектуалдық бақылауда әртүрлі деректерді байланыстыра талдау студенттің программалау әрекеті туралы жеке көрсеткіштерге қарағанда неғұрлым мазмұнды ақпарат қалыптастыруға мүмкіндік береді.

Үшінші ерекшелік студенттің білім жағдайын бір реттік бақылау нәтижесі бойынша емес, оның уақыт бойынша өзгеруін ескере отырып қарастырумен байланысты. Интеллектуалдық оқу құралдарында студенттің алдыңғы тапсырмаларды орындау нәтижелері жинақталып, кейінгі әрекеттерімен байланыстыра талдануы мүмкін. Мысалы, Code-DKT моделінде студенттің алдыңғы тапсырмалар бойынша әрекеттері мен программалық кодынан алынған белгілер кейінгі тапсырманы орындау нәтижесін болжау үшін пайдаланылады [49]. Бұл программалаудан білімді бақылауда студенттің жекелеген нәтижелерін тіркеумен шектелмей, оның білімді меңгеру барысындағы өзгерістерді динамикада қарастыру мүмкіндігін көрсетеді.

Төртінші ерекшелік бақылау нәтижелері негізінде студенттің нақты программалық шешіміне сәйкес мазмұндық және жекелендірілген кері байланыс қалыптастыру мүмкіндігімен байланысты. Интеллектуалдық оқу құралдары программалық код пен анықталған қателерді талдау нәтижелерін пайдаланып, қатенің ықтимал себептерін түсіндіруге, программалық шешімді жетілдіру жолдарын ұсынуға және кері байланысты нақты орындалған тапсырманың контексіне бейімдеуге мүмкіндік береді [51, 106]. Зерттеулерде студенттің программалық коды ескеріліп қалыптастырылған GenAI кері байланысының нақтылығы, түсініктілігі және түзетуге бағытталуы студенттер тарапынан жоғары бағаланғаны көрсетілген [53]. Бұл интеллектуалдық бақылауда кері байланысты тек бақылау нәтижесін хабарлау құралы ретінде емес, алынған нәтижені студентке мазмұндық тұрғыдан түсіндіру құралы ретінде қарастыруға мүмкіндік береді.

Бесінші ерекшелік – интеллектуалдық оқу құралдары қалыптастырған бақылау нәтижелерінің сенімділігін тексеру және оларды педагогикалық тұрғыдан интерпретациялау қажеттілігі. Интеллектуалдық талдау немесе GenAI арқылы алынған нәтиже студент білімінің дайын көрсеткіші ретінде қабылданбай, басқа бақылау деректерімен, бағаланатын білім мазмұнымен және оқу мақсаттарымен байланыстыра қарастырылуы тиіс [51, 54].

Жоғарыда жүргізілген талдау интеллектуалдық оқу құралдарын қолданып программалаудан білімді бақылаудың ерекшеліктері бақылау деректерінің құрамы, оларды талдау сипаты, білім жағдайының өзгерісін ескеру, кері байланыс қалыптастыру және алынған нәтижелерді педагогикалық интерпретациялау тұрғысынан көрінетінін көрсетті. Анықталған ерекшеліктер мен олардың программалаудан білімді бақылаудағы мәні 4-кестеде жүйеленді.

Кесте 4 – Интеллектуалдық оқу құралдарын қолданып программалаудан білімді бақылаудың ерекшеліктері

Ерекшелік	Бақылаудағы көрінісі	Ғылыми негіз	Білімді бақылаудағы мәні
Бақылау деректері құрамының кеңеюі	Тапсырманың орындалу нәтижесімен қатар программалық код, қателер, орындалу көрсеткіштері және әрекеттер тарихы пайдаланылады	[49, 105, 106]	Студенттің программалау әрекетін әртүрлі деректер негізінде сипаттауға мүмкіндік береді
Бақылау деректерін мазмұндық және өзара байланыстыра талдау	Кодтың құрылымдық және семантикалық сипаттамалары талданады; тапсырма нәтижелері кодтан алынатын белгілермен байланыстырылады	[49, 105, 106]	Бақылауды дұрыс/қате нәтижесін тіркеумен шектемей, программалық шешімнің ерекшеліктері туралы ақпаратпен толықтырады
Білім жағдайының өзгерісін динамикада қарастыру	Бірнеше тапсырма бойынша жинақталған әрекеттер мен кодтан алынған белгілер кейінгі нәтижені болжауда пайдаланылады	[49]	Жекелеген нәтижелермен шектелмей, студенттің білімді меңгеру барысындағы өзгерістерді ескеруге мүмкіндік береді
Мазмұндық және жекелендірілген кері байланыс	Қателер түсіндіріледі, кодты жетілдіру бойынша ұсыныстар беріледі, кері байланыс студенттің нақты программалық шешіміне бейімделеді	[51, 53, 106]	Бақылау нәтижесін студентке түсіндіруге және нақты шешімге қатысты кері байланыс беруге мүмкіндік береді
Бақылау нәтижелерін тексеру және педагогикалық интерпретациялау қажеттілігі	Интеллектуалдық құрал қалыптастырған нәтиженің дұрыстығы мен сенімділігі тексеріліп, басқа бақылау деректерімен байланыстырылады	[51, 54]	Интеллектуалдық құрал нәтижесін студент білімінің дайын көрсеткіші ретінде қабылдауға жол бермейді

Кестеде жүйеленген ерекшеліктер интеллектуалдық оқу құралдарының программалаудан білімді бақылауда атқаратын қызметтері бір ғана міндетпен шектелмейтінін көрсетеді. Олардың бір бөлігі программалық код пен тапсырманы орындау нәтижелерін талдауға бағытталса, екіншісі студенттің жинақталған әрекеттері негізінде білім жағдайының өзгерісін модельдеуге, ал үшіншісі бақылау нәтижелерін түсіндіру және жекелендірілген кері байланыс қалыптастыруға мүмкіндік береді. Сонымен қатар алынған нәтижелердің сенімділігін тексеру және оларды педагогикалық тұрғыдан интерпретациялау қажеттілігі сақталады.

Осы қызметтерді бір модельде өзара байланыстыру студенттің білім жағдайын кешенді сипаттаудың қажетті шарты болып табылады

Алайда қарастырылған тәсілдерде программалық шешімді автоматты тексеру, кодты мазмұндық талдау, студенттің жинақталған әрекеттерін модельдеу және интеллектуалдық кері байланыс қалыптастыру нәтижелерін

студенттің программалаудан білім жағдайын тұтас сипаттау үшін өзара байланыстыру мәселесі толық шешілмеген.

1.3 Программалаудан студенттердің білімін интеллектуалдық бақылау қағидалары және бағалау критерийлері

Зерттеу жұмысының 1.1 және 1.2 пункттерінде программалаудан студенттердің білімін бақылау тек студенттің жазған кодының соңғы нәтижесін тексерумен шектелмейтіні негізделді. Программалаудан білімді бағалауда студенттің есепті түсінуі, алгоритмдік шешім құруы, программаның кодын жазуы, қателерді анықтап түзетуі және шешімін түсіндіруі сияқты әрекеттерін де ескеру маңызды. Бұл әрекеттер барысында программаның коды, тестілеу нәтижелері, тапсырманы орындау тарихы, жіберілген қателер мен оларды түзету әрекеттері туралы әртүрлі деректерді автоматтандырылған және интеллектуалдық құралдар арқылы жинақтап, талдауға болады. Бірақ алынған деректер мен оларды интеллектуалдық өңдеу нәтижесінде қалыптасқан аналитикалық көрсеткіштер студенттің білімі туралы тікелей қорытынды жасауға жеткіліксіз болып табылады.

Алынған деректер мен аналитикалық көрсеткіштер оқу мақсаттарымен, күтілетін оқу нәтижелерімен және бағаланатын әрекеттермен сәйкестендірілген жағдайда ғана студенттің білім деңгейі жайлы педагогикалық тұрғыдан негізделген қорытынды жасауға мүмкіндік береді.

Бақылау нәтижелерін педагогикалық тұрғыдан негіздеу білімді бақылауды ұйымдастыруда басшылыққа алынатын қағидалармен байланысты. Педагогикалық әдебиеттерде қағида педагогикалық үдерісті ұйымдастыруға бағыт беретін жалпы норма және жетекші талап ретінде қарастырылады. П.И. Пидкасистый оқыту қағидаларын оқу үдерісін ұйымдастырудың жалпы нормалары ретінде сипаттайды [39]. В.И. Загвязинский оқыту қағидаларын оқыту заңдылықтарымен және оқу үдерісін ұйымдастыруға қойылатын талаптармен байланыстыра қарастырады [108]. С.Б. Бабаев пен Ж.К. Оңалбек педагогикалық үдеріс қағидаларын оның заңдылықтарына негізделетін, педагог пен білім алушының іс-әрекетіне бағыт беретін жетекші талаптар ретінде қарастырады [109]. Осы тұжырымдарға сүйене отырып, зерттеуімізде қағида бақылау үдерісін ұйымдастыру мен жүзеге асырудың жалпы бағытын айқындайтын жетекші ереже ретінде қарастырылады.

И.П. Подласый бақылаудың негізгі қағидалары ретінде объективтілікті, жүйелілікті және жариялылықты бөліп көрсетеді. Объективтілік бақылау мен бағалаудың белгіленген критерийлерге негізделуін көздесе, жүйелілік бақылаудың формалары, әдістері мен құралдарын өзара байланыста қолданумен сипатталады. Жариялылық білім алушыларды бірдей критерийлер бойынша бағалауды, бағалау нәтижелерін жариялау мен негіздеуді қарастырады [40]. Н.В. Басова бақылаудың валидтілігі мен сенімділігіне тоқталып, валидтілікті бақылаудың оқу мақсаттарына, білім мазмұны мен бағаланатын әрекеттерге сәйкестігімен, ал сенімділікті алынған нәтижелердің тұрақтылығымен сипаттайды [41].

Аталған қағидаларды нақты пәннен білімді бақылауда қолдану барысында пәннің мазмұны мен білім алушы орындайтын әрекеттердің ерекшеліктері ескерілуі тиіс. Осы тұрғыдан программалаудан білімді бақылауда программалау әрекетінің ерекшеліктеріне назар аудару маңызды.

R. Lister және т.б. студенттердің программалаудан білімін бақылауда код жазумен қатар, программалық кодты оқу, яғни оның құрылымы мен орындалу логикасын түсіну, сондай-ақ программаның орындалу барысын қадам бойынша қадағалау әрекеттерінің де маңызды екенін көрсетеді [110]. Программалау тапсырмаларын автоматтандырылған бағалауда программаның функционалдық дұрыстығын тестілік жағдайлар арқылы тексеру кеңінен қолданылады. Мұндай тексеру программаның дұрыс нәтиже беретінін анықтауға мүмкіндік бергенімен, студенттің тапсырманы қалай орындағаны және шешімге қалай келгені туралы толық ақпарат бермейді [38]. Бұл қағида бақылау нысанын программаның нәтижесінен оны құру үдерісіне дейін кеңейтеді.

Студенттің программалау әрекеттері программалық код, тест нәтижелері, қателер және код өзгерістерінің тарихы түріндегі бақылау деректерін қалыптастырады. Алайда бұл деректер студент білімінің тікелей көрсеткіші ретінде емес, олардың қандай білім мен әрекетті сипаттайтыны ескеріліп, оқу мақсаты және бағаланатын білім мазмұнымен сәйкестікте интерпретациялануы тиіс.

Интеллектуалдық бақылаудың тағы бір ерекшелігі – студенттің жекелеген тапсырманы орындау нәтижесімен қатар, оның оқу әрекеттері туралы жинақталған деректер негізінде білім жағдайы мен өзгеру динамикасын талдау мүмкіндігі. Студенттің білімді меңгеруін модельдеу (student modeling) және білімді меңгеру динамикасын қадағалау (knowledge tracing) модельдері студенттің тапсырмаларды орындау тарихы негізінде белгілі бір білім компоненттерін меңгеруін бағалауға және келесі тапсырманы дұрыс орындау ықтималдығын анықтауға мүмкіндік береді. Программалаудан білімді бақылауға бейімделген мұндай модельдерде тапсырмалардың дұрыс немесе қате орындалуы туралы деректермен қатар, студент жазған программалық код туралы ақпарат та пайдаланылуы мүмкін [49, 111].

Студенттің білімді меңгеруін модельдеу нәтижелерін түсіндіруде олардың қандай білім компонентіне, тапсырмалар мен студент әрекеттеріне қатысты екені ескерілуі қажет. Бұл мәселе генеративті жасанды интеллект құралдарын программалау тапсырмаларын орындауда қолдану жағдайында ерекше мәнге ие болады.

Zh. Nurbekova және т.б. GenAI құралдарына шамадан тыс сүйену студенттердің есепті дербес шешу қабілеті мен алгоритмдік ойлауына кері әсер етуі мүмкін екенін көрсетіп, студенттің алдымен тапсырманы өздігінен орындауын, ал ЖИ құралдарын кейін шешімді тексеру, қателерді анықтау және түзету үшін қолдануды ұсынады [112]. Сондықтан GenAI қолданылған жағдайда программалаудан білімді бақылау нәтиженің дұрыстығымен шектелмей, студенттің алынған шешімді өз білімі негізінде тексеру, түсіндіру және негіздеу әрекеттерін де қамтуы тиіс.

Программалаудан білімді интеллектуалдық бақылауда алынған ақпараттың студенттің қандай білімі мен әрекетін сипаттайтынын анықтау маңызды. Программалық код, тестілеу нәтижелері мен тапсырманы орындау тарихынан

алынған ақпарат соңғы программалық нәтижемен және оған жету барысымен өзара байланыста қарастырылуы тиіс. Бұл ақпаратты студенттің білімін бағалауда пайдалану бақылаудың мақсаты мен бағаланатын білім мазмұнына сәйкестігін анықтауды қажет етеді.

Студенттің білімін бағалау, өз кезегінде, нақты белгіленген критерийлерге сүйенуі тиіс. Z. Nurbekova және т.б. программалаудан білімді бақылауда GenAI қолдануды талдай отырып, программаны бағалауда оның дұрыстығын, уақыт пен жад шектеулеріне сәйкестігін, деректер құрылымы мен алгоритмнің дұрыс таңдалуын ескеру қажеттігін көрсетеді [51]. S. Ferrara және S. Qunbar автоматтандырылған бағалау нәтижелерін бағаланатын білім мазмұнымен байланыстыру және олардың қандай негізде алынғанын түсіндіру қажеттілігіне назар аударады [113].

Интеллектуалдық бақылау нәтижелері студенттің білімін бағалауда оқытушы пайдаланатын ақпараттың бір бөлігін құрайды. Интеллектуалдық бақылауда студенттің тапсырманы орындауы туралы деректер өңделіп, оның білімі мен жіберген қателері туралы ақпарат алуға болады. Алайда осы ақпарат негізінде студенттің білімін бағалау және оқу үдерісіне қатысты шешім қабылдау оқытушының қатысуын қажет етеді. Z. Nurbekova және т.б. программалаудан білімді бақылауда GenAI мен онлайн-платформалардың мүмкіндіктерін талдай отырып, автоматтандырылған бақылау жағдайында да студенттің қателерін талдау, күрделі ұғымдарды түсіндіру және оқу үдерісін түзетуде оқытушының рөлі сақталатынын көрсетеді [76]. W. Khan және т.б. ұсынған адам қатысатын ЖИ негізіндегі бағалау моделінде ЖИ студенттің жұмысын талдап, алдын ала бағалайды, ал оқытушы бағалау нәтижесін тексеріп, студенттің жұмысына берілетін қорытынды бағаны анықтайды [114]. Бұл зерттеулер интеллектуалдық бақылаудың оқытушының педагогикалық шешімін алмастырмай, оны қабылдауға қажетті мәліметтермен қамтамасыз етуге бағытталуы тиіс екенін көрсетеді.

Программалаудан білімді бақылаудың пәндік ерекшелігі программалау есебінің шарты, оны шешу алгоритмі, программалық код және орындалу нәтижесі арасындағы байланысты ажырата қарастыру қажеттілігімен анықталады. Бір есеп әртүрлі алгоритмдік тәсілдермен шешілуі, ал таңдалған алгоритм программалау тілінің әртүрлі конструкциялары арқылы іске асырылуы мүмкін. Сонымен қатар алгоритмдік шешім дұрыс құрылғанымен, оны программада іске асыру барысында синтаксистік, логикалық немесе орындалу қателері жіберілуі мүмкін. Керісінше, программаның берілген тестілік жағдайларда дұрыс нәтиже беруі таңдалған алгоритмнің тиімділігін, кодтың сапасын немесе студенттің қолданылған шешімді түсінгенін өздігінен сипаттамайды. Сондықтан программалаудан білімді бақылауда есепті түсіну, алгоритмдік шешім құру, оны программалық кодта іске асыру және алынған нәтижені тексеру өзара байланысты болғанымен, студент білімінің әртүрлі қырларын сипаттайтын әрекеттер ретінде қарастырылуы тиіс.

Аталған ерекшелікті нақты программалау тапсырмасы арқылы көрсетуге болады. Мысалы, берілген санның жай сан екенін анықтау қажет болсын. Бұл тапсырманы шешудің екі программалық нұсқасын қарастырайық.

Мысалы: 1-нұсқа. Бөлгіштерді 2-ден (n-1)-ге дейін тексеру:

```
def is_prime_1(n):
```

```
    if n < 2:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True
```

2-нұсқа. Бөлгіштерді sqrt(n)-ге дейін тексеру:

```
def is_prime_2(n):
    if n < 2:
        return False
    i = 2
    while i * i <= n:
        if n % i == 0:
            return False
        i += 1
    return True
```

Екі программа да санның жай сан екенін дұрыс анықтайды, алайда оларда қолданылған алгоритмдер тиімділігі жағынан ерекшеленеді. Бірінші нұсқада бөлгіштер 2-ден (n-1)-ге дейін тексерілетіндіктен, алгоритмнің уақыттық күрделілігі (O(n), ал екінші нұсқада тек i^2<=n шартын қанағаттандыратын бөлгіштер тексеріліп, уақыттық күрделілік O(sqrt(n)) болады. Демек, бірдей функционалдық нәтиже студенттің алгоритмдік шешімінің тиімділігін бірдей сипаттамайды. Сонымен қатар дұрыс алгоритм таңдалғанның өзінде цикл шекарасын, бөлінгіштікті тексеру шартын немесе (n<2) жағдайын программалық кодта қате іске асыру жекелеген кіріс деректерінде қате нәтиже беруі мүмкін. Бұл жағдайда қатенің себебі есепті шешу алгоритмінде емес, оны программалық іске асыруда болады. Осы мысал программалаудан білімді бақылауда функционалдық нәтижені, алгоритмдік шешімді және оның программалық іске асырылуын өзара байланысты, бірақ ажыратылатын бағалау қырлары ретінде қарастыру қажеттігін көрсетеді.

Аталған ерекшеліктер бақылауда функционалдық нәтиже, алгоритмдік шешім, программалық іске асыру және үдерістік деректерді ажырата әрі өзара байланысты қарастыруды талап етеді

Жүргізілген талдау программалаудан білімді интеллектуалдық бақылаудың қағидаларын анықтауда педагогикалық бақылаудың жалпы талаптарымен қатар, аталған пәндік ерекшеліктерді және интеллектуалдық құралдар арқылы алынатын деректердің сипатын ескеру қажеттігін көрсетті. Программалау тапсырмасының орындалу нәтижесі студент білімінің жекелеген қырларын толық сипаттай алмайтындықтан, бақылауда есепті түсіну, алгоритмдік шешім құру, оны программалық іске асыру, нәтижені тексеру және қатені түзету барысында қалыптасатын деректер оқу мақсаты мен бағаланатын білім мазмұнына сәйкес қарастырылуы тиіс. Бұл ретте интеллектуалдық құралдар арқылы алынған деректер бағалау критерийлерімен байланыстырылып, оларды педагогикалық тұрғыдан түсіндіру және қорытынды шешім қабылдау оқытушының қатысуымен жүзеге асырылуы тиіс.

Осы теориялық тұжырымдар негізінде программалаудан студенттердің білімін интеллектуалдық бақылаудың мынадай қағидалары анықталды:

– *Бақылау мақсатына сәйкестік қағидасы* – интеллектуалдық бақылауда пайдаланылатын деректер мен алынған нәтижелердің бақылау мақсатына, бағаланатын білім мазмұнына және студенттің бағаланатын әрекеттеріне сәйкестігін көздейді.

– *Көпкомпоненттілік және көпдеректілік қағидасы* – программалаудан білімді бақылауда студенттің әртүрлі әрекеттерін қамтуға және оларды бағалау үшін бірнеше өзара толықтырушы дерек көздерін пайдалануды көздейді.

– *Нәтиже мен әрекеттің бірлігі қағидасы* – программалаудан білімді бақылауда студенттің тапсырманы орындау нәтижесі мен оған жету барысында жасаған әрекеттерін өзара байланыста қарастыруды білдіреді.

– *Бағалау критерийлеріне негізделу қағидасы* – студенттің білімін бағалаудың нақты белгіленген критерийлерге негізделуін көздейді. Бұл ретте алынған нәтиженің қандай критерийлер мен деректерге сүйеніп анықталғаны түсінікті болуы тиіс.

– *Педагогикалық шешім мен интеллектуалдық қолдаудың бірлігі қағидасы* – интеллектуалдық бақылауда алынған ақпаратты студенттің білімін бағалау барысында пайдалана отырып, педагогикалық шешімді оқытушының қатысуымен қабылдауды қарастырады.

Анықталған қағидаларды программалаудан білімді интеллектуалдық бақылауда жүзеге асыру студенттің программалаудан білімінің қалыптасу деңгейін және есептерді шешу барысында орындайтын әрекеттерін қандай критерийлер негізінде бағалау қажеттігін нақтылауды талап етеді. Сондықтан бағалау критерийлерін анықтаудың теориялық негізі критериалды бағалаудың педагогикалық мәнімен және оның оқу нәтижелерін бағалаудағы ерекшеліктерімен байланысты қарастырылады.

Критериалды бағалау білім алушының оқу жетістіктерін білім беру мақсаттары мен мазмұнына сәйкес алдын ала анықталған критерийлермен салыстыруға негізделеді. С. Авдарсоль диссертациясында келтірілген О.И. Можаева, А.С. Шилибекова және Д.Б. Зиеденованың анықтамасында критерийлердің нақты айқындалуы және білім беру үдерісінің барлық қатысушыларына алдын ала белгілі болуы атап көрсетіледі [115]. Осы тұрғыдан оқу мақсаты мен күтілетін нәтиже бағалаудың бағытын анықтаса, критерий қол жеткізілген нәтижені бағалаудың негізін нақтылайды. Сондықтан бағалау критерийлері оқу мақсаттарымен және күтілетін нәтижелермен өзара сәйкестікте белгіленуі тиіс [116].

Бағалау критерийлерін қолдану олардың көрсеткіштері мен жетістік деңгейлерін нақтылауды қажет етеді. С. Авдарсольдің еңбегінде Э.А. Красновский мен Т.Л. Коганның көзқарастары баяндалып, білім сапасын бағалауда оның белгілерін бірмәнді сипаттау және білім алушының жауабында байқалатын көрсеткіштерін ажырату қажеттігі көрсетілген [115]. Сондай-ақ В.А. Родионовқа сүйеніп, критерийлердің оқу міндеттерімен анықталатыны, ал дескрипторлардың әр критерий бойынша білім алушының жетістік деңгейлерін сипаттайтыны түсіндіріледі [115]. Осы тұжырымдарды жинақтай отырып, критерий бағаланатын сапаны айқындайды, көрсеткіш сол сапаның жауапта

немесе орындалған әрекетте байқалуын нақтылайды, ал дескриптор критерий бойынша қол жеткізілген деңгейді сипаттайды деп қарастыруға болады.

Критерийлерді білім алушының әрекеттерімен және оқу нәтижелерімен байланыстыру Н.К. Токжигитованың визуалды программалаудан оқу жетістіктерін бағалауға арналған зерттеуінде нақтыланады. Автор мульти-критериалды бағалауды оқу барысында жүзеге асырылатын жобалау және программалау іс-әрекеттерінің түрлеріне сәйкес білім алушылардың оқу нәтижелерін критериалды бағалау ретінде анықтайды. Осыған сәйкес бағалау мазмұны оқу нәтижелерімен және сол нәтижелерге жеткізетін әрекеттермен байланыстырылады. Зерттеуде программаның орындалу дұрыстығын тексеру студенттің программалаудан дайындығын толық сипаттауға жеткіліксіз екені көрсетіліп, программа кодының сапасын және программалық өнімді әзірлеудің есептің қойылуы, алгоритм құру, кодтау, тестілеу мен енгізу кезеңдерін қамтитын кешенді бағалау қажеттілігі негізделеді [31, 117].

Бұл тұжырымдар программалаудан білімді бағалауда студенттің тек дайын программалық өнімін ғана емес, оны әзірлеу барысында жүзеге асыратын әрекеттерін де ескеру қажеттігін көрсетеді. Ал интеллектуалдық бақылау жағдайында бұл талап жаңа мазмұнға ие болады. Себебі студенттің есепті шешу барысы программаның коды, тестілеу нәтижелері, жіберілген қателер, оларды түзету әрекеттері, тапсырманы орындау тарихы және түсіндірмелері сияқты әртүрлі деректер арқылы тіркелуі мүмкін. Мұндай деректер студент әрекетінің жекелеген белгілерін сипаттағанымен, олардың әрқайсысы өздігінен оның программалаудан білім деңгейі туралы педагогикалық қорытынды жасауға жеткіліксіз. Сондықтан программалаудан білімді интеллектуалдық бақылауда бағалау критерийлері студенттің орындайтын әрекеттері мен сол әрекеттер туралы алынатын деректерді бағаланатын білім мазмұнымен байланыстыруға мүмкіндік беруі тиіс.

Программалаудан білімді кешенді бағалаудың қажеттілігін автоматтандырылған бағалау құралдарының қамту ауқымын талдайтын зерттеулер де көрсетеді. M. Messer және т.б. программалауды автоматтандырылған бағалау және кері байланыс құралдарына арналған 121 зерттеуге жүргізген жүйелі талдау нәтижесінде олардың басым бөлігі программалау тапсырмасының функционалдық дұрыстығын бағалауға бағытталғанын анықтаған [38]. Мұндай бағалау негізінен программаны тестік деректер арқылы тексеріп, оның күтілетін нәтижені беруін анықтауға немесе программалық кодты статикалық талдау арқылы студенттің шешімін эталондық шешіммен салыстыруға негізделеді. Сонымен қатар авторлар программалық кодтың сүйемелдеуге қолайлылығы, оқылымдылығы және құжатталуы сияқты сапалық сипаттарын бағалайтын құралдардың салыстырмалы түрде аз екенін көрсетеді.

Қорыта келе, Н.К. Токжигитованың зерттеуінде программалаудан оқу жетістіктерін әрекеттер мен программалық өнімді әзірлеу кезеңдері тұрғысынан кешенді бағалау қажеттілігі негізделсе [31], Messer және т.б. зерттеуінде автоматтандырылған бағалаудың іс жүзінде көбіне программаның функционалдық дұрыстығына шоғырланатыны және кодтың басқа сапалық сипаттарын қамтуының шектеулі екені анықталған [38].

Автоматтандырылған бағалауда программаның функционалдық дұрыстығымен қатар код сапасын және студенттің тапсырманы орындау барысында жинақталатын деректерін ескеру мүмкіндігі Chen, Nguyen және Dow зерттеуінде қарастырылған. Авторлар командалық программалау жобаларында студенттердің жеке үлесін анықтау үшін код сапасын бағалау схемасы енгізілген автоматтандырылған программалауды бағалау жүйесін ұсынған [118]. Жүйеде студенттердің жобаға қатысу белсенділігі апта сайын тіркеліп, код сапасына негізделген көрсеткіштер олардың командалық жұмысқа қосқан үлесінің деңгейін анықтау үшін пайдаланылған. 146 студенттен құралған 41 команданың тапсырмаларды орындау деректеріне жүргізілген эмпирикалық талдау ұсынылған жүйенің ЖОО-дағы топтық жобаларды бақылауда қолданылу мүмкіндігін көрсеткен. Аталған зерттеу программалау әрекетін бағалауда дайын программалық өнімнің нәтижесімен қатар код сапасы мен студенттің жұмыс барысы туралы жинақталатын деректерді де пайдалануға болатындығын көрсетеді.

Қарастырылған зерттеулер программалаудан білімді бағалауда функционалдық дұрыстықпен шектелмей, программа кодының сапасы мен тапсырманы орындау барысында қалыптасатын әртүрлі деректерді ескеру қажеттігін көрсетеді. Интеллектуалдық бақылау жағдайында олардың педагогикалық мәні осы деректердің студенттің программалау тапсырмасын орындау барысында көрінетін білімі мен әрекеттерінің қандай қырларын сипаттайтыны анықталған жағдайда ғана ашылады. Сондықтан бағалау критерийлерін анықтауда интеллектуалдық құралдар тіркейтін көрсеткіштерді тікелей критерий ретінде қабылдамай, алдымен олардың студенттің қандай білімі мен әрекетін сипаттайтынын айқындау қажет.

GenAI жағдайында программалау дағдыларын бағалау мәселесі авторлық зерттеулерде де қарастырылды. Зерттеу барысында функционалдық дұрыстыққа негізделген бағалау студенттің программалау әрекетін толық сипаттауға жеткіліксіз екені көрсетіліп, бағалауда соңғы нәтижемен қатар шешім қабылдау үдерісін, алгоритмдік талдауды және студенттің өз шешімін түсіндіруі мен негіздеуін ескеру қажеттігі айқындалды. Осыған сәйкес программалау дағдыларын нәтижелік, үдерістік және рефлексиялық өлшемдер негізінде кешенді бағалау тәсілі ұсынылды [32, 119]. Бұл нәтижелер программалаудан білімді бағалау критерийлерін тек программаның функционалдық дұрыстығымен шектемей, студенттің есепті шешу барысында жүзеге асыратын әрекеттерімен байланыстыра анықтау қажеттігін көрсетеді.

Осыған байланысты бағалау критерийлерін студенттің программалау әрекетінде көрінетін білім сипаттары мен оларды айқындайтын көрсеткіштер негізінде қарастыру маңызды. Мұндай талдау бағалау критерийлерін алдын ала белгіленген техникалық көрсеткіштерден емес, программалау тапсырмасын орындау барысында көрінетін студенттің білімі мен әрекетінің мазмұны негізінде айқындауға мүмкіндік береді [120].

Программалаудан білімді бағалау критерийлерін айқындау мәселесі авторлардың алдыңғы зерттеуінде де қарастырылды. Зерттеуде программалау құзыреттілігін бағалау теориялық білім, алгоритмдік ойлау, практикалық дағдылар, жобалық және коммуникативтік дағдылар бағыттары бойынша жүйеленді [121]. Алгоритмдік ойлауды бағалауда есепті талдау мен алгоритм

құру, ал практикалық дағдыларды бағалауда код жазу және қателерді түзету көрсеткіштері ажыратылып, олардың төмен, орта және жоғары деңгейлік сипаттамалары ұсынылды.

Программалаудан білімді бағалауда осындай танымдық әрекеттерді сипаттаудың теориялық негіздерінің бірі ретінде Блум таксономиясының Л. Андерсон мен Д. Кратвол ұсынған жаңартылған нұсқасын қарастыруға болады. Онда танымдық үдерістер есте сақтау, түсіну, қолдану, талдау, бағалау және жасау әрекеттері арқылы сипатталады [45]. Бұл жіктеудің программалаудан білімді бақылау үшін маңызы студенттің білімін белгілі бір деңгейге механикалық жатқызуда емес, программалау тапсырмасын орындау барысында қандай танымдық әрекеттер жүзеге асатынын және олардың қандай нәтижелер мен деректер арқылы байқалатынын анықтауда көрінеді [122].

Блум таксономиясының танымдық деңгейлерін программалаудан білімді бағалаумен байланыстыру біздің алдыңғы зерттеулерімізде де қарастырылды. Программалаудан студенттердің білімін бағалау критерийлері мен дескрипторларын жасауға арналған зерттеуде программалаудың негізгі мазмұндық бөлімдері бойынша бағалау критерийлері анықталып, олар Блум таксономиясының когнитивтік деңгейлерімен сәйкестендірілді. Бағалау дескрипторларын құрастыруда студенттің өлшенетін әрекеттеріне сүйену, логикалық және синтаксистік қателерді ажырату және білімнің қалыптасуын деңгейлік сипаттау негізге алынды [123]. Бұл нәтижелер программалаудан білімді бағалау критерийлерін студенттің орындаған әрекетімен ғана емес, сол әрекеттің танымдық мазмұнымен байланыстыра қарастыру қажеттігін көрсетеді.

Программалаудан білімді бағалау критерийлерін интеллектуалдық құралдардың мүмкіндіктерімен байланыстыру мәселесі кейінгі авторлық зерттеулерде жалғасын тапты. Программалау тапсырмаларын репродуктивті, конструктивті және зерттеу сипатындағы деңгейлерге бөліп қарастыру негізінде әр деңгейде студент орындайтын әрекеттердің сипаты мен оларды бағалау көрсеткіштерін ажырату қажеттілігі анықталды. Репродуктивті деңгейде дайын алгоритмді қолдану, конструктивті деңгейде өзіндік алгоритм құрастыру, ал зерттеу сипатындағы тапсырмаларда күрделі программалау міндеттерін орындау қарастырылды. Сонымен қатар GenAI технологиясының программалық кодты автоматты тексеру, оның логикалық құрылымын талдау және бағалау нәтижелерін өңдеу мүмкіндіктері зерделенді [124]. Бұл тұжырымдар интеллектуалдық құралдар арқылы алынатын көрсеткіштерді студент орындайтын тапсырманың мазмұны мен танымдық күрделілігімен байланыстыра қарастыру қажеттігін көрсетеді

Программалау тапсырмаларын орындау барысында студенттің білімі мен әрекетін сипаттайтын көрсеткіштерді нақтылау кейінгі авторлық зерттеуде жалғасын тапты. Зерттеуде студенттердің программалау есептерін шешу үдерісі есепті түсіну, алгоритмдік ойлау, шешімнің дұрыстығы мен тиімділігі, код сапасы, шеттік жағдайларды өңдеу және шешімді түсіндіру сияқты жеті педагогикалық көрсеткіш бойынша қарастырылды [125]. Эксперимент нәтижелері тапсырмалардың күрделілігі артқан сайын студенттердің нәтижелеріндегі айырмашылықтар әсіресе алгоритмнің тиімділігі, шеттік жағдайларды өңдеу және шешімді түсіндіру көрсеткіштерінде айқынырақ

көрінетінін көрсетті. Бұл нәтижелер программалаудан білімді бағалауда дайын программалық нәтижемен қатар есепті шешу үдерісінде көрінетін танымдық, алгоритмдік және түсіндірмелік сипаттарды да ескеру қажеттігін негіздейді.

Осы нәтижелерді диссертациялық зерттеудің міндеттері тұрғысынан қарастырғанда, анықталған көрсеткіштерді тікелей бағалау критерийлері ретінде қабылдаудан гөрі, олардың программалау әрекетінің қандай қырларын сипаттайтынын ескере отырып жүйелеу қажет болды. Көрсеткіштердің мазмұндық байланысын талдау есепті түсінуді дербес бағалау қыры ретінде қарастыруға, алгоритмдік ойлау мен шешім тиімділігін алгоритмдік шешім аясында, ал программаның дұрыстығы мен шеттік жағдайларды өңдеуді функционалдық дұрыстық аясында біріктіруге негіз болды. Код сапасы программалық шешімнің құрылымдық-сапалық сипаттарын, ал шешімді түсіндіру студенттің өз әрекетін саналы түрде түсіндіруі мен негіздеуін сипаттайтындықтан, олар жеке бағалау қырлары ретінде сақталды. Сонымен бірге алдыңғы теориялық талдауда программалау әрекетінің маңызды құрамдасы ретінде айқындалған қатені анықтау, оның себебін талдау және түзету әрекеттері нәтижелік көрсеткіштерден өзгеше үдерістік ақпарат беретіндіктен, диссертациялық зерттеуде жеке бағалау қыры ретінде қарастырылды.

Программалау есебін шешу оның шартын талдаудан басталады. Студент берілген және ізделінді шамаларды, кіріс және шығыс деректерін, қойылған талаптарды айқындап, олардың арасындағы байланысты түсінуі қажет. Есеп мазмұнын осылай ұғыну алгоритмдік шешім құруға негіз болады. Блум–Андерсон–Кратвол таксономиясы бойынша есеп шартын ұғыну түсіну, ал берілгендер мен талаптар арасындағы байланысты ажырату талдау танымдық үдерістерімен байланысты. Авторлық зерттеуде де есепті түсіну программалау есебін шешу үдерісін сипаттайтын педагогикалық көрсеткіштердің бірі ретінде қарастырылған [45]. Осыған сүйеніп, студенттің есеп мазмұнын, бастапқы шарттары мен күтілетін нәтижесін дұрыс түсінуі тапсырма мазмұнын түсіну критерийі ретінде бөліп алуға болады.

Есептің шартын түсіну оны шешу жолын бірден анықтап бермейді. Студент есепке сәйкес орындалатын әрекеттердің ретін белгілеп, алгоритмдік құрылымдарды таңдап, олардың өзара байланысын ұйымдастыруы қажет. Бір есептің бірнеше шешу тәсілі болуы мүмкін. Сондықтан дайын алгоритмді қолданумен қатар, есептің ерекшелігіне сай тәсілді таңдау немесе өзіндік алгоритм құру және оның тиімділігін ескеру алгоритмдік шешімді бағалауда мәнге ие болады.

Блум–Андерсон–Кратвол таксономиясы бойынша белгілі алгоритмдік құрылымдарды пайдалану қолдану, есепке сәйкес тәсілді таңдау мен алгоритм қадамдарының байланысын анықтау талдау, ал өзіндік алгоритм құру жасау танымдық үдерістерімен байланысты. Авторлық зерттеуде алгоритмдік ойлау мен шешімнің тиімділігі программалау есептерін орындаудың педагогикалық көрсеткіштері ретінде қарастырылып, күрделі тапсырмаларда тиімді алгоритм құру студенттерге қиындық туғызатыны анықталды [45]. Осы тұрғыдан алгоритмдік шешім критерийі студенттің есеп мазмұнын әрекеттер жүйесіне айналдыруын, шешу тәсілін негізді таңдауын және оның тиімділігін ескеруін сипаттайды.

Алгоритмдік шешімнің дұрыс құрылуы оның программалық кодта қатесіз іске асырылатынын өздігінен білдірмейді. Код жазу барысында студент программалау тілінің конструкцияларын қолданып, алгоритм қадамдарын орындалатын программаға айналдырады. Осы іске асырудың есеп талабына сәйкестігі функционалдық дұрыстық критерийі арқылы бағаланады. Программаның берілген кіріс деректері бойынша күтілетін нәтиже беруімен шектелмей, есеп шартында көзделген негізгі және шеттік жағдайларда да дұрыс орындалуы ескеріледі. Сондықтан бұл критерий программалық шешімнің қойылған талаптарды орындауын сипаттайды; ал тексеру нәтижесін түсіндіру қолданылған тестілердің қандай жағдайларды қамтығанына тәуелді.

Блум–Андерсон–Кратвол таксономиясы бойынша алгоритмді программалық кодта жүзеге асыру қолдану танымдық үдерісімен байланысты. Ал студенттің алынған нәтижені есеп талабымен салыстырып, орындалмаған жағдайды және оның себебін анықтауы талдау үдерісін көрсетеді. Автоматтандырылған бағалау жүйесінде тестілік жағдайлар программаның функционалдық дұрыстығын тексеруге мүмкіндік береді, бірақ тест нәтижесі студенттің бұл талдауды өзі жүргізгенін дәлелдемейді. Авторлық зерттеуде шешімнің дұрыстығы мен шеттік жағдайларды өңдеу программалау есебін орындаудың педагогикалық көрсеткіштері ретінде қарастырылған [45]. Сондықтан интеллектуалдық бақылауда тест нәтижелері функционалдық дұрыстық туралы дерек ретінде алынып, студенттің қатені түсіндіруі мен оны түзету әрекеті бөлек қарастырылуы қажет.

Программаның дұрыс нәтиже беруі оның кодының қалай құрылғанын толық сипаттамайды. Бірдей есепті дұрыс шешетін программалар құрылымы, оқылымдылығы, программалау конструкцияларын қолдануы және сүйемелдеуге қолайлылығы жағынан ерекшеленуі мүмкін. Сондықтан функционалдық дұрыстықпен қатар программалық шешімнің осы сипаттарын да қарастыру қажет. Олар код сапасын жеке бағалау критерийі ретінде бөліп көрсетуге негіз болады.

Код сапасы студенттің программалау тілінің конструкцияларын орынды қолдануынан, код бөліктерінің өзара байланысын түсіндіруінен және қабылдаған шешімін сапа талаптарына сүйеніп негіздеуінен байқалады. Бұл әрекеттер Блум–Андерсон–Кратвол таксономиясындағы қолдану, талдау және бағалау танымдық үдерістеріне сәйкес келеді. Автоматтандырылған бағалау жөніндегі зерттеулерде кодтың оқылымдылығы, сүйемелдеуге қолайлылығы мен құжатталуын ескеру қажеттігі атап көрсетілген [45]. Авторлық зерттеуде де код сапасы программалау есебін шешудің педагогикалық көрсеткіші ретінде қарастырылған [45]. Кодтың құрылымдық белгілері бағалауға дерек бергенімен, студенттің өз шешімін түсіндіріп, оның сапасын негіздей алуы бөлек анықталуы қажет.

Программалау барысында жіберілген қате студенттің тиісті білімді меңгермегенін бірден білдірмейді. Оның себебі есепті шешу тәсілінен, алгоритмді кодта іске асырудан немесе программалау тілінің конструкцияларын қолданудан туындауы мүмкін. Сондықтан бақылауда қатенің болу фактісімен шектелмей, студенттің нақты және күтілетін нәтижені салыстыруы, қатенің себебін анықтауы, кодқа өзгеріс енгізуі және нәтижені қайта тексеруі ескеріледі. Бұл әрекеттер дайын программаның нәтижесінен

көрінбейтін үдерістік ақпарат береді және қатені анықтау мен түзетуді дербес бағалау критерийі ретінде қарастыруға негіз болады.

Қатенің себебін анықтау Блум–Андерсон–Кратвол таксономиясындағы талдау, түзету тәсілін таңдап негіздеу бағалау, ал өзгерісті кодта жүзеге асыру қолдану үдерістерімен байланысты. Авторлық зерттеуде бағалау дескрипторларын құрастыру кезінде логикалық және синтаксистік қателерді ажыратуға назар аударылған [45]. Интеллектуалдық бақылауда қате хабарламасы мен код өзгерістерінің тарихы студент әрекеті туралы дерек береді; оның қатенің себебін түсінгені енгізген өзгерісін түсіндіруі және қайта тексеру нәтижесі арқылы анықталады.

Программаның дұрыс жұмыс істеуі студенттің оны қандай шешу тәсілімен құрғанын және қабылдаған шешімдерін қаншалықты түсінетінін толық көрсетпейді. Мұны анықтау үшін студент қолданған алгоритмін, оның қадамдарының логикасын және алгоритмнің кодта қалай іске асырылғанын түсіндіруі қажет. Таңдалған тәсілдің орындылығын негіздеуі де дайын программаның нәтижесінен алынбайтын ақпарат береді. Сондықтан шешімді түсіндіру және негіздеу программалаудан білімді бағалаудың жеке қыры ретінде қарастырылады.

Студент алгоритм мен код бөліктерінің қызметін түсіндіргенде Блум–Андерсон–Кратвол таксономиясындағы түсіну, олардың өзара байланысын ашқанда талдау, ал шешу тәсілін белгіленген талаптарға сүйеніп негіздегенде бағалау танымдық үдерістері көрінеді. Авторлық зерттеуде шешімді түсіндіру педагогикалық көрсеткіштердің бірі ретінде қарастырылып, эксперимент барысында студенттерге қиындық туғызғаны анықталған [45]. Осы критерий бойынша студенттің түсіндірмесі кодтың дұрыс орындалуын қайталайтын мәлімет ретінде емес, оның өз шешімін қаншалықты түсінетінін анықтайтын дәлел ретінде пайдаланылады.

Теориялық талдау мен авторлық зерттеулер нәтижесінде программалаудан білімнің алты өзара байланысты бағалау қыры айқындалды Осы негізде тапсырма мазмұнын түсіну, алгоритмдік шешім, функционалдық дұрыстық, код сапасы, қатені анықтау және түзету, шешімді түсіндіру және негіздеу критерийлері айқындалды. Бұл критерийлер тапсырманы орындау кезеңдерінің тізімі емес. Олар студенттің программалау әрекетінде байқалатын, өзара байланысты болғанымен, бөлек бағалауды қажет ететін білім қырларын сипаттайды.

Интеллектуалдық бақылауда әр критерийді қолдану үшін оның студент әрекетінде қалай байқалатынын және сол әрекет туралы қандай дерек алынатынын анықтау қажет. Код, тестілеу нәтижесі, өзгерістер тарихы мен студент түсіндірмесі бағалауға дерек береді, бірақ олардан есептелген техникалық белгілер педагогикалық қорытындымен теңестірілмейді. Көрсеткіштің мәні оның бағалау критерийімен және тапсырманың оқу мақсатымен байланысы арқылы түсіндіріледі. Осы байланысты нақтылау үшін алты критерий, олардың байқалатын көрсеткіштері, дерек көздері және студент әрекетіне сәйкес танымдық үдерістер 5-кестеде жүйеленді.

Кесте 5 – Программалаудан білімді интеллектуалдық бақылаудың бағалау критерийлері, көрсеткіштері және дереккөздер

Бағалау критерийі	Блум–Андерсон–Кратвол таксономиясымен сәйкестігі	Негізгі көрсеткіштер	Дерек көздері
Тапсырма мазмұнын түсіну	Түсіну; берілгендер мен талаптар арасындағы байланысты ажыратқанда талдау	Есептің мақсатын, кіріс және шығыс деректерін, шектеулерін анықтау	Есеп шарты бойынша студенттің жауабы; кіріс–шығыс пен шектеулерге берген түсіндірмесі
Алгоритмдік шешім	Белгілі тәсілді қолданғанда қолдану; тәсілдерді салыстырғанда талдау; өзіндік алгоритм құрғанда жасау	Алгоритмнің қисындылығы; құрылымдарды таңдау; тәсілдің тиімділігін негіздеу	Алгоритм немесе шешу жоспары; код құрылымы; балама тәсілдер туралы студенттің негіздемесі
Функционалдық дұрыстық	Шешімді іске асырғанда қолдану; сәтсіз жағдайдың себебін өзі анықтағанда талдау	Негізгі және шеттік жағдайлардағы нәтиженің дұрыстығы; сәтсіз жағдайды түсіндіру	Күтілетін нәтижесі белгіленген тестілер; кіріс–шығыс және орындалу нәтижелері; студенттің сәтсіз тестіге түсіндірмесі
Код сапасы	Конструкцияларды пайдаланғанда қолдану; код бөліктерінің байланысын ашқанда талдау; өз шешімін сапа талаптарымен салыстырғанда бағалау	Құрылымның біріздігі; атаулардың түсініктілігі; конструкциялардың орындылығы	Программалық код; статикалық талдау нәтижесі; студенттің код құрылымы туралы түсіндірмесі
Қатені анықтау және түзету	Себебін анықтағанда талдау; түзету тәсілін таңдағанда бағалау; өзгерісті енгізгенде қолдану	Қатенің себебін анықтау; түзетуді негіздеу; нәтижені қайта тексеру	Қате хабарламасы мен сәтсіз тестілер; код өзгерістерінің тарихы; студент түсіндірмесі; қайта тестілеу нәтижесі
Шешімді түсіндіру және негіздеу	Код пен алгоритм қызметін түсіндіргенде түсіну; олардың байланысын ашқанда талдау; тәсілін негіздегенде бағалау	Алгоритм қадамдары мен код қызметін түсіндіру; шешімнің орындылығын негіздеу	Студенттің ауызша немесе жазбаша түсіндірмесі; алгоритм сипаттамасы; негіздеу сұрақтарына жауаптары

5-кестеде бағалау критерийлері студент әрекетінде байқалатын көрсеткіштермен және оларды анықтауға қажетті дерек көздерімен

сәйкестендірілді. Блум–Андерсон–Кратвол таксономиясымен байланысы әр критерий аясында студенттің қандай әрекеті қарастырылатынын нақтылайды. Осылайша, программалық код пен тест нәтижелері бағалау мазмұнымен байланыстырылған дерек ретінде қолданылады.

Анықталған қағидалар бақылау деректерін жинау мен түсіндіруге қойылатын талаптарды, ал критерийлер студент білімінің бағаланатын қырларын белгілейді. Тапсырманың нәтижесі мен оны орындау барысы бірге қарастырылғанда студенттің білімі туралы диагностикалық қорытынды жасауға негіз қалыптасады. Деректер әртүрлі түсіндіруге жол берген жағдайда қорытындыны оқытушы қосымша тексереді. Бұл байланыс программалаудан білімді интеллектуалдық бақылау моделін құруға негіз болады.

1-тарау бойынша қорытынды

Жоғары оқу орнында программалаудан студенттердің білімін бақылаудың қазіргі жағдайын талдау бұл үдерістің программалауды оқыту мазмұны мен қолданылатын цифрлық құралдардың дамуына байланысты өзгеріп отырғанын көрсетті. Қазіргі автоматтандырылған бақылау жүйелері программалық кодты тестілеуге, оның функционалдық дұрыстығын анықтауға және студентке жедел кері байланыс беруге мүмкіндік береді. Сонымен бірге студенттің программалау тапсырмасын орындауы тек дайын программалық кодпен шектелмейді. Есепті түсіну, шешу алгоритмін құру, кодты жазу, оны тексеру, қателерді анықтау және түзету, қабылданған шешімді түсіндіру де студенттің программалаудан білімін сипаттайды. Сондықтан программалаудан білімді бақылауда тапсырманың соңғы нәтижесімен қатар оны орындау үдерісін де ескеру маңызды.

Программалаудан білімді бақылауда интеллектуалдық оқу құралдарын қолдану бақылау мүмкіндіктерін кеңейтеді. Интеллектуалдық жүйелер студенттің оқу әрекеті туралы деректерді жинақтауға және талдауға, оның білім жағдайын сипаттауға, кері байланыс беруге мүмкіндік береді. Ал генеративті жасанды интеллект құралдарының программалық кодты генерациялау, толықтыру, талдау, қателерді анықтау және түзету, программалық шешімді түсіндіру мүмкіндіктері программалаудан білімді бақылауға жаңа талаптар қояды. Мұндай жағдайда студент ұсынған дайын программалық код оның білім деңгейін толық сипаттай алмайды. Осыған байланысты студенттің тапсырманы орындау барысындағы әрекеттерін, жіберген қателері мен оларды түзету жолдарын, шешімді түсіндіруін және генеративті ЖИ-мен өзара әрекеттесуін бақылау деректері ретінде қарастыру қажет.

Зерттеу барысында программалаудан студенттердің білімін интеллектуалдық бақылауды ұйымдастыруға негіз болатын қағидалар анықталды. Оларға педагогикалық мақсаттылық және валидтілік; көпкомпоненттілік және көпдеректілік; нәтиже мен үдерістің бірлігі; критерийлік негізділік және түсіндірмелілік; педагогикалық шешім мен интеллектуалдық қолдаудың бірлігі қағидалары жатады. Бұл қағидалар программалаудан білімді бақылауда студент әрекетінің нәтижесі мен үдерісін өзара байланыста қарастыруға, әртүрлі деректерді пайдалануға және алынған нәтижелерді педагогикалық тұрғыдан түсіндіруге негіз болады.

Программалау әрекетінің ерекшеліктерін және жүргізілген зерттеулердің нәтижелерін жинақтау негізінде программалаудан білімді интеллектуалдық бақылаудың бағалау критерийлері мен көрсеткіштері нақтыланды. Олар студенттің есептің мазмұнын түсінуін, алгоритмдік шешім құруын, программалық шешімнің функционалдық дұрыстығын, код сапасын, қателерді анықтау мен түзетуін, шешімді түсіндіруі мен негіздеуін бағалауға бағытталған. Әрбір көрсеткішті студенттің программалау әрекетінен алынатын нақты деректермен байланыстыру білімді бағалауда тек соңғы нәтижені ғана емес, тапсырманы орындау үдерісін де ескеруге мүмкіндік береді.

Сонымен, бірінші бөлімде жүргізілген зерттеу нәтижесінде программалаудан студенттердің білімін бақылаудың қазіргі жағдайы талданып, интеллектуалдық оқу құралдарын қолдану жағдайындағы бақылаудың

ерекшеліктері айқындалды, интеллектуалдық бақылаудың қағидалары анықталып, бағалау критерийлері мен көрсеткіштері нақтыланды. Алынған нәтижелер программалаудан студенттердің білімін бақылауда нәтиже мен үдерісті сипаттайтын деректерді бірлікте қарастыру қажеттігін көрсетті және программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін жасауға теориялық негіз болды.

2 ПРОГРАММАЛАУДАН СТУДЕНТТЕРДІҢ БІЛІМІН БАҚЫЛАУДЫҢ ИНТЕЛЛЕКТУАЛДЫҚ МОДЕЛІН ЖАСАУ ЖӘНЕ ІСКЕ АСЫРУ

2.1 Программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі

Зерттеу жұмысының алдыңғы бөлімдерінде жүргізілген теориялық талдау программалаудан студенттердің білімін интеллектуалдық бақылау өзара байланысты бірнеше үдерісті қамтитынын көрсетті. Мұндай бақылауда студенттің білімін анықтау тапсырманың орындалу нәтижесін тіркеумен ғана шектелмей, бағаланатын білім мазмұнын, оны студенттің программалау әрекетінде сипаттайтын критерийлер мен көрсеткіштерді, студент білімінің көрінуіне мүмкіндік беретін программалау тапсырмаларын, оларды орындау барысында қалыптасатын деректерді және осы деректерді талдау мен педагогикалық тұрғыдан интерпретациялау тәсілдерін өзара байланыста қарастыруды талап етеді. Аталған элементтердің әрқайсысы бақылау үдерісінде белгілі бір қызмет атқарғанымен, студенттің программалаудан білім жағдайы туралы негізді қорытынды олардың өзара байланысы арқылы қалыптасады. Сондықтан программалаудан студенттердің білімін интеллектуалдық бақылауды жекелеген бақылау және бағалау әрекеттерінің жиынтығы ретінде емес, оның мақсаты, бағалау негіздері, тапсырмалары, деректері, оларды талдау және интерпретациялау үдерістері өзара байланысқан тұтас жүйе ретінде қарастыру қажеттілігі туындайды.

Программалаудан студенттердің білімін интеллектуалдық бақылауды ұйымдастыруда, ең алдымен, бақылаудың педагогикалық мақсаты мен нысанын нақтылау қажет. Зерттеудің теориялық бөлімінде жүргізілген талдау программалаудан білімді бақылаудың нысаны студенттің тапсырманы орындау нәтижесімен немесе дайын программалық кодпен ғана шектелмейтінін көрсетті. Программалаудан білім студенттің есептің мазмұнын түсінуі, оны шешу тәсілін анықтауы, алгоритм құруы, алгоритмді программалық кодта жүзеге асыруы, алынған нәтижені тексеруі, жіберілген қателерді анықтауы және түзетуі барысында көрінеді. Сондықтан студенттің программалаудан білімін бақылауда соңғы нәтижемен қатар оны алуға бағытталған әрекеттерді де ескеру қажет. Бұл бақылаудың мақсатын тапсырманың орындалу дұрыстығын анықтаумен шектемей, студенттің программалаудан білім жағдайын анықтауға бағыттауға мүмкіндік береді.

Программалаудан білімді студенттің тапсырманы орындау барысында жүзеге асыратын әрекеттерімен байланыстыра бағалау қажеттілігі Н.К. Токжигитованың зерттеуінде де негізделген. Автор визуалды программалау бойынша оқу жетістіктерін бағалау критерийлерін анықтауда программалау үдерісінің өзін негізге алып, студенттің тапсырманы орындау әрекетін міндетті түсіну және тиімді шешу стратегиясын таңдау, есепті шешу алгоритмін әзірлеу, алгоритмді программалық кодта жүзеге асыру, кодты тексеру және қойылған талаптарға сәйкес түзету кезеңдерімен байланыстырады. Осы әрекеттерді талдау негізінде программалаудан оқу жетістіктерінің мазмұны программалау тілінің синтаксисін, программалау қағидалары мен тұжырымдарын білу, кодты

түсіну және білімді қолданумен қатар, мәселені шешу мен жоспарлау, программаны ретке келтіру, түсінікті код жазу және шешім құру дағдыларымен нақтыланады. Бұл программалаудан білімді бағалауда дайын программаның орындалу дұрыстығымен шектелмей, тапсырманы шешудің әртүрлі кезеңдерінде көрінетін білім мен әрекет сипаттарын ескеру қажеттігін көрсетеді.

Осы ұстаным зерттеуіміздің 1.3 бөлімінде программалаудан студенттердің білімін интеллектуалдық бақылаудың бағалау критерийлері мен көрсеткіштерін анықтауға негіз болды. Теориялық талдау мен авторлық зерттеулер нәтижелерін жүйелеу негізінде тапсырма мазмұнын түсіну, алгоритмдік шешім, функционалдық дұрыстық, код сапасы, қатені анықтау және түзету, шешімді түсіндіру және негіздеу бағалау критерийлері айқындалды. Аталған критерийлер программалау тапсырмасын орындау кезеңдерін тікелей қайталамай, студент білімінің программалау әрекеті барысында байқалатын педагогикалық мәні бар қырларын сипаттайды. Әр критерий студент әрекетінде байқалатын тиісті көрсеткіштер арқылы нақтыланып, оларды анықтауға мүмкіндік беретін дерек көздерімен байланыстырылды.

1.3 бөлімде анықталған бағалау критерийлері мен көрсеткіштері студент жетістігінің байқалу дәрежесін саралауға негіз болады. Әр критерий бойынша деңгейлік дескрипторлар эксперименттік жұмыста қолданылатын тапсырмалар мен алынатын деректерге сәйкес нақтыланады.

Студенттің критерийлер бойынша жетістігін бағалау тиісті программалау тапсырмаларын іріктеуді және олардың күрделілігін алдын ала саралауды қажет етеді. J. Sheard және т.б. программалау бойынша емтихан тапсырмаларының күрделілігін бірнеше белгі арқылы сипаттап, олардың қатарына тұжырымдамалық күрделілік пен Блум таксономиясына негізделген интеллектуалдық күрделілікті енгізеді [126]. Авторлар осы белгілерді тапсырмалардың қиындық бағасымен салыстырып, олардың арасындағы байланысты анықтаған.

Осыған сәйкес зерттеуімізде программалау тапсырмаларын іріктеу кезінде 1.3 бөлімде анықталған бағалау критерийлері негізге алынды. Тапсырмалар студенттің осы критерийлерге сәйкес білім сипаттарын көрсетуіне мүмкіндік беретін мазмұны және оларды орындауға қажетті әрекеттердің күрделілігі бойынша сараланды. Тапсырманың күрделілік деңгейі оны орындаған студенттің жетістік деңгейін өздігінен анықтамайды. Студенттің әр критерий бойынша жетістігі тапсырманы орындау барысында байқалған көрсеткіштер мен оларды растайтын деректер негізінде бағаланады.

Программалау тапсырмасын орындау бір ғана танымдық әрекетпен шектелмейді. S. Masapanta-Carrión және J.Á. Velázquez-Iturbide программалау жаттығуларын оларды шешуге қажетті бірнеше когнитивтік үдеріс бойынша жіктеуді ұсынады және жүргізілген зерттеуде мұндай жіктеудің оқытушылар үшін табиғи болғанын көрсетеді [127]. Осы тұрғыдан программалау тапсырмасы студенттің бірнеше өзара байланысты әрекетінің көрінуіне мүмкіндік беретін кешенді бақылау бірлігі ретінде қарастырылуы мүмкін.

Зерттеуімізде осы ерекшелік 1.3 бөлімде анықталған бағалау көрсеткіштерін программалау тапсырмаларымен байланыстыруда ескерілді: бір тапсырманы орындау барысында бірнеше көрсеткіштің байқалуы мүмкін, ал

белгілі бір көрсеткіш әртүрлі тапсырмаларды орындау нәтижелері негізінде анықталады.

Программалау әрекетінің бақылау үшін дерек көзі ретіндегі маңыздылығы программалық тапсырманы орындаудың соңғы нәтижесі мен оған жету үдерісінің студент білімі туралы әртүрлі ақпарат беруімен анықталады. Зерттеудің 1.2 бөлімінде қарастырылған үдеріске бағытталған тәсілдерде программалаудың дайын нәтижесімен қатар, оны әзірлеу барысында қалыптасатын аралық код нұсқалары, компиляция нәтижелері және программаны құру тарихы сияқты деректерді талдаудың маңыздылығы көрсетіледі [69]. Бұл студенттің программалау әрекетін тек соңғы программалық өнім арқылы емес, оны құру барысында орындалған әрекеттер мен олардың нәтижелері арқылы да қарастыруға негіз болады. Осы тұрғыдан программалау әрекеті студенттің тапсырманы орындау үдерісі туралы ақпарат қалыптастыратын маңызды дерек көзі болып табылады.

Программалау әрекетінен алынатын деректер мазмұны мен ақпараттық мүмкіндігі жағынан біртекті емес және студенттің программалау дағдысының әртүрлі қырларын сипаттауға мүмкіндік береді. Zhang және т.б. студенттердің программалау тапсырмаларын орындау барысында қалыптасқан жіберілімдер тарихын (submission traces) талдап, олардың дұрыстығы мен тапсырманы орындауға жұмсалған уақытқа қатысты 11 белгіні анықтаған. Осы белгілерді факторлық талдау нәтижесінде авторлар код жазу дағдысының төрт жасырын құрамдасын: код стилін сақтау, синтаксистік және семантикалық дағдыларды, сондай-ақ синтаксистік қателерді түзету дағдысын бөліп көрсеткен [128]. Зерттеу нәтижелері программалау әрекеті барысында жинақталатын үдерістік деректерді талдау студенттің код жазу дағдысын бір ғана нәтижелік көрсеткішпен емес, оның жекелеген құрамдастары бойынша сипаттауға мүмкіндік беретінін көрсетеді.

Алайда программалау әрекетінен алынған деректерді студент білімінің белгілі бір көрсеткішімен тікелей және бірмәнді сәйкестендіруге болмайды. Бір ғана дерек немесе одан анықталған белгі студент білімінің әртүрлі сипаттарымен байланысты болуы мүмкін, ал белгілі бір бағалау көрсеткішін анықтау бірнеше дерек көзінен алынған ақпаратты қажет етуі ықтимал. Сондықтан интеллектуалдық бақылауда программалау әрекетінен алынған бастапқы деректерді білім көрсеткішінің өзі ретінде емес, оны анықтауға пайдаланылатын ақпараттық негіз ретінде қарастыру қажет.

Осы талап зерттеудің 1.3 бөлімінде бағалау көрсеткіштерін оларды сипаттайтын дерек көздерімен байланыстыруға негіз болды. Программалық кодтан, тестілеу нәтижелерінен, қателер мен оларды түзету әрекеттерінен және тапсырманы орындау тарихынан алынатын ақпарат осы тұрғыдан жеке-жеке бақылау нәтижесі ретінде емес, тиісті бағалау көрсеткіштерін анықтауға пайдаланылатын деректер ретінде қарастырылды. Көрсеткіштер мен дерек көздерінің мұндай байланысы студенттің программалау әрекетінен алынған әртекті деректерді бағаланатын білім мазмұнымен сәйкестендіріп, оларды тиісті көрсеткіштер тұрғысынан талдауға мүмкіндік береді. Демек, интеллектуалдық бақылауда деректерді жинақтаумен ғана шектелмей, олардан бағалау көрсеткіштерін сипаттайтын белгілерді анықтау және сол белгілерді педагогикалық тұрғыдан интерпретациялау қажеттілігі туындайды.

Программалау әрекетінен алынған бастапқы деректерден бағалау көрсеткіштерін сипаттайтын белгілерді анықтау оларды тиісті талдау тәсілдері арқылы өңдеуді қажет етеді. Мұнда талдау тәсілін таңдау пайдаланылатын деректің түріне және одан алынуы тиіс ақпаратқа байланысты. Программалық кодтың құрылымы мен мазмұндық ерекшеліктерін анықтау бір талдау тәсілдерін қажет етсе, программаның орындалуы мен тестілеу нәтижелерін тексеру басқа тәсілдер арқылы жүзеге асырылады. Ал студенттің бірнеше тапсырманы орындауы барысында жинақталған деректер оның білім жағдайындағы өзгерістерді уақыт бойынша қарастыруға мүмкіндік береді. Сондықтан интеллектуалдық бақылауда әртүрлі талдау тәсілдерінің қолданылуы олардың технологиялық мүмкіндіктерімен ғана емес, нақты деректен бағалау үшін қажетті қандай белгіні анықтай алатындығымен негізделуі тиіс.

Тапсырманы орындау барысында алынған деректер студенттің программалау әрекетінің жекелеген нәтижелерін сипаттайды, алайда оларды тікелей білім деңгейі ретінде қарастыруға болмайды. Студенттің білім жағдайын анықтау үшін алынған деректерді бағалау көрсеткіштерімен байланыстыра өңдеу және олардың өзгерісін талдау қажет. Осы міндет программалық код пен оның орындалу нәтижелерін талдау әдістерін студенттің жинақталған оқу әрекеттері негізінде білім жағдайын модельдейтін интеллектуалдық тәсілдермен үйлестіруді талап етеді.

Программалық код пен оның орындалу нәтижелерін талдауда статикалық және динамикалық талдау тәсілдері әртүрлі ақпарат алуға мүмкіндік береді. K.M. Ala-Mutka программалау тапсырмаларын автоматтандырылған бағалау тәсілдерін талдай отырып, студенттің программалық шешімін тексеруде программаның орындалу нәтижелерімен қатар оның кодына қатысты сипаттамаларды да бағалау мүмкіндігін көрсетеді [33]. Messer, Brown, Kölling және Shi программалау тапсырмаларын автоматтандырылған бағалау құралдарына жүргізген жүйелі шолуында динамикалық талдаудың көбіне программаның функционалдық дұрыстығын unit-тестілер арқылы тексеруге, ал статикалық талдаудың программалық кодты орындамай талдауға және оның сапалық сипаттарын бағалауға қолданылатынын көрсетеді [38]. Осыған сәйкес статикалық талдау программалық кодтан оның құрылымы мен сапасына қатысты белгілерді, ал динамикалық талдау программаның орындалуы, тестілерден өтуі және орындалу кезінде туындайтын қателер туралы белгілерді анықтауға мүмкіндік береді. Интеллектуалдық бақылау тұрғысынан мұндай талдау нәтижелері студент білімінің тікелей бағасы болып табылмайды, олар тиісті бағалау көрсеткіштерін сипаттауға пайдаланылатын ақпараттық белгілер ретінде қарастырылады.

Алайда жеке программалау тапсырмасы бойынша статикалық және динамикалық талдау нәтижелері студенттің белгілі бір уақыттағы әрекеті туралы ақпарат бергенімен, оның программалаудан білім жағдайын толық сипаттау үшін жеткіліксіз. Студенттің білімі әртүрлі тапсырмаларды орындау барысында көрінетіндіктен, жекелеген тапсырмалардан алынған белгілерді өзара байланыстырып, олардың уақыт бойынша өзгеруін ескеру қажет. Бұл жағдайда бақылау нысаны бір программалық шешімнің сипаттамаларынан студенттің жинақталған оқу әрекеттеріне қарай кеңейеді. Соның нәтижесінде

студенттің алдыңғы тапсырмаларды орындау нәтижелері, жіберген қателері, оларды түзетуі және кейінгі тапсырмалардағы әрекеттері оның білім жағдайын сипаттауға пайдаланылатын өзара байланысты деректер ретінде қарастырылады.

Студенттің жинақталған оқу әрекеттері негізінде оның білім жағдайын анықтау student modeling және knowledge tracing тәсілдерінің негізгі міндеттерінің бірі болып табылады. Corbett пен Anderson студенттің оқу тапсырмаларын орындау барысында жинақталатын әрекеттері негізінде оның тиісті білім бірліктерін меңгеру ықтималдығын бағалауды knowledge tracing ретінде қарастырады [47]. Кейін Piech және т.б. бұл тәсілді дамыта отырып, студенттің оқу әрекеттерінің реттілігі негізінде оның білім жағдайындағы өзгерістерді модельдеуге арналған Deep Knowledge Tracing моделін ұсынды [48]. Осы тұрғыдан интеллектуалдық бақылауда knowledge tracing тәсілдері жекелеген тапсырмалардың нәтижелерін бір-бірінен тәуелсіз қарастырумен шектелмей, студенттің алдыңғы оқу әрекеттерін ескере отырып, оның білім жағдайын уақыт бойынша модельдеуге мүмкіндік береді.

Программалаудан білімді бақылауда студенттің тапсырманы орындау нәтижесімен қатар оның жазған программалық кодын ескеру білім жағдайын модельдеудің ақпараттық мүмкіндігін кеңейтеді. Осы бағытта Shi және т.б. ұсынған Code-DKT моделінде студенттің алдыңғы тапсырмаларды орындау деректері программалық кодтан автоматты түрде алынатын белгілермен толықтырылады [49]. Авторлар Code-DKT моделінде студенттің программалық кодынан пәндік-спецификалық белгілерді автоматты түрде анықтап, оларды Deep Knowledge Tracing моделіне енгізеді. Мұндай тәсіл студенттің білім жағдайындағы өзгерістерді тапсырманың дұрыс немесе қате орындалуы туралы ақпаратпен ғана шектемей, оның программалық шешімінде көрінетін сипаттарды да ескере отырып модельдеуге мүмкіндік береді. Бұл программалаудан интеллектуалдық бақылауда кодтан және тапсырманы орындау тарихынан алынған деректерді бірлесіп талдаудың қажеттілігін көрсетеді [129].

Алайда интеллектуалдық әдістер арқылы студенттің білім жағдайын модельдеу бақылаудың педагогикалық нәтижесін өздігінен анықтамайды. Статикалық және динамикалық талдау арқылы алынған белгілер, сондай-ақ студенттің тапсырмаларды орындау тарихы негізінде модельденген білім жағдайы бақылаудың мақсаты мен бағаланатын білім мазмұны тұрғысынан түсіндірілуі қажет. Осыған байланысты интеллектуалдық талдау нәтижелерін зерттеудің 1.3 бөлімінде анықталған бағалау көрсеткіштерімен байланыстыру маңызды. Бұл жағдайда программалау әрекетінен алынған деректер мен оларды талдау нәтижелері студент білімінің тікелей бағасы ретінде емес, тиісті көрсеткіштердің байқалу дәрежесін анықтауға пайдаланылатын ақпарат ретінде қарастырылады.

Бағалау көрсеткіштерімен байланыстырылған талдау нәтижелері студент білімінің жекелеген сипаттарының байқалу дәрежесін анықтауға негіз болады. Бұл үшін әртүрлі дерек көздерінен алынған белгілер тиісті көрсеткіштер тұрғысынан біріктіріліп, олардың 1.3 бөлімде белгіленген деңгейлік сипаттамаларға сәйкестігі анықталады. Нәтижесінде интеллектуалдық талдау барысында алынған ақпарат педагогикалық бағалау жүйесіне енгізіліп,

студенттің программалаудан білімінің қалыптасу деңгейін сипаттауға мүмкіндік береді. Интеллектуалдық бақылауда білім жағдайын анықтау деректерді автоматты өңдеудің нәтижесімен шектелмей, оларды алдын ала белгіленген бағалау көрсеткіштері мен деңгейлері негізінде педагогикалық интерпретациялауды қамтиды.

Студенттің программалаудан анықталған білім жағдайы бақылаудың қорытынды нәтижесі ғана емес, одан кейінгі педагогикалық әрекеттерді айқындауға негіз болатын ақпарат ретінде қарастырылуы тиіс. Бағалау көрсеткіштері мен олардың деңгейлік сипаттамалары негізінде алынған нәтиже студенттің программалаудан білімінің қандай тұстары жеткілікті қалыптасқанын және қандай тұстары қосымша жетілдіруді қажет ететінін анықтауға мүмкіндік береді. Осы ақпаратты педагогикалық тұрғыдан интерпретациялау студентке оның қол жеткізген нәтижелері мен қиындықтары туралы кері байланыс беруге, сондай-ақ кейінгі бақылауда ұсынылатын тапсырмаларды оның анықталған білім жағдайына сәйкес таңдауға негіз болады. Интеллектуалдық бақылау нәтижесі білім жағдайын тіркеумен аяқталмай, оны студенттің әрі қарайғы оқу әрекетін ұйымдастыруда пайдаланумен жалғасады.

Жүргізілген талдауды жинақтау программалаудан студенттердің білімін интеллектуалдық бақылаудың өзара байланысты бірнеше үдерістің бірлігінде жүзеге асырылатынын көрсетеді. Бақылаудың педагогикалық мақсаты мен бағаланатын білім мазмұны студент білімінің программалау әрекетінде байқалуын сипаттайтын бағалау көрсеткіштерін айқындайды; олардың көріну дәрежесін анықтау тиісті деңгейдегі программалау тапсырмалары арқылы жүзеге асырылады. Тапсырмаларды орындау барысында қалыптасатын программалық код, тестілеу нәтижелері, қателер, оларды түзету әрекеттері және орындау тарихы студенттің программалау әрекеті туралы бастапқы деректерді құрайды. Бұл деректерді статикалық және динамикалық талдау, студенттің білім жағдайын модельдеу және білімді меңгеру динамикасын қадағалау тәсілдері арқылы өңдеу бағалау көрсеткіштерін сипаттайтын ақпараттық белгілерді анықтауға мүмкіндік береді. Ал алынған нәтижелерді белгіленген көрсеткіштер мен деңгейлер тұрғысынан педагогикалық интерпретациялау студенттің программалаудан білім жағдайын анықтауға және оның кейінгі оқу әрекетіне қатысты кері байланыс қалыптастыруға негіз болады.

Анықталған байланыстар программалаудан студенттердің білімін интеллектуалдық бақылауды жекелеген бағалау немесе технологиялық әрекеттердің жиынтығы ретінде емес, өзара тәуелді элементтері белгілі бір реттілікпен қызмет ететін тұтас жүйе ретінде қарастыру қажеттігін көрсетеді. Мұндай жүйеде бақылаудың педагогикалық мақсаты бағалау негіздерін айқындайды, бағалау критерийлері мен көрсеткіштері программалау тапсырмаларын іріктеуге бағдар береді, тапсырмаларды орындау барысында қалыптасқан деректер интеллектуалдық талдаудың ақпараттық негізін құрайды, ал талдау нәтижелері студенттің білім жағдайын анықтау және педагогикалық тұрғыдан интерпретациялау үшін пайдаланылады. Сонымен қатар анықталған білім жағдайының кейінгі тапсырмаларды таңдауға ықпал етуі бақылаудың аталған элементтері арасында кері байланыстың болуын талап етеді. Осы өзара

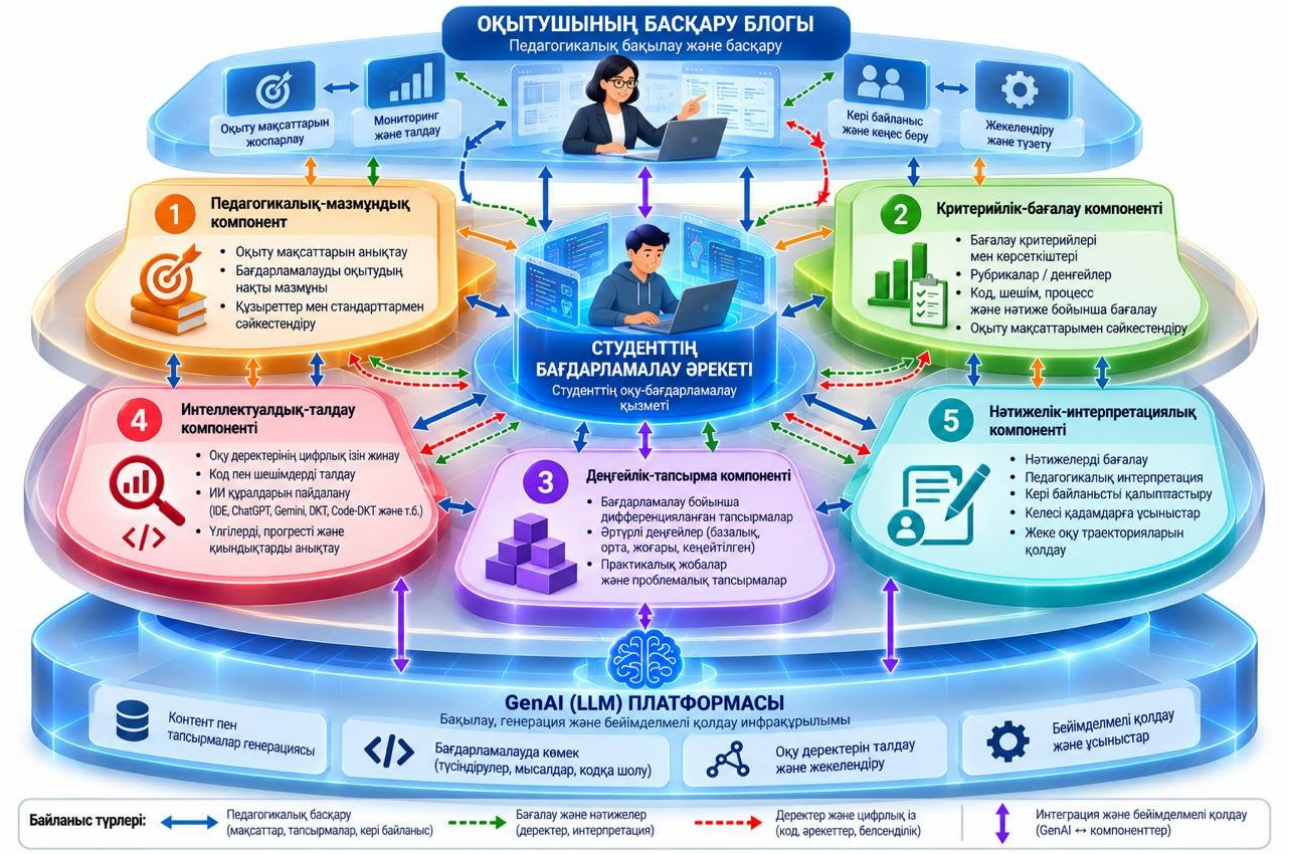
байланыстарды құрылымдық түрде біріктіру программалаудан студенттердің білімін интеллектуалдық бақылау моделін әзірлеуге негіз болады.

Білім беру технологиялары саласындағы қазіргі зерттеулерде модельді әзірлеудің теориялық негізділігі мен жүйелілігіне ерекше мән беріледі. N. Tusquellas және т.б. [130] білім беру технологиялары саласында модельді әзірлеуді мәселені анықтау, тұжырымдамалық негіздеу, модельді әзірлеу және оның ғылыми үлесін айқындау кезеңдерін қамтитын жүйелі үдеріс ретінде қарастырады. Осы тұрғыдан программалаудан студенттердің білімін интеллектуалдық бақылау моделінің құрылымын анықтау оның элементтерін алдын ала белгілеуге емес, зерттеудің теориялық бөлімінде анықталған программалаудан білімді бақылаудың ерекшеліктерін, бағалау негіздерін, программалау әрекетінен алынатын деректерді, оларды интеллектуалдық талдау нәтижелері мен педагогикалық интерпретация арасындағы байланыстарды жүйелеуге негізделеді [130].

Осыған сүйене отырып, программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі бақылаудың педагогикалық мақсаты мен бағаланатын білім мазмұнын, бағалау критерийлері мен көрсеткіштерін, студент білімінің байқалуына мүмкіндік беретін программалау тапсырмаларын, оларды орындау барысында қалыптасатын деректерді, осы деректерді интеллектуалдық талдауды және алынған нәтижелерді педагогикалық интерпретациялауды өзара байланыста қамтуы тиіс деп есептейміз. Осыған сәйкес модель құрылымында педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттер анықталды.

Программалаудан студенттердің білімін интеллектуалдық бақылаудың анықталған компоненттері мен олардың өзара байланыстары 2-суретте модель түрінде ұсынылды.

2-суретте ұсынылған модель программалаудан студенттердің білімін интеллектуалдық бақылауды педагогикалық мақсаттан бастап студенттің білім жағдайын анықтау, оны педагогикалық тұрғыдан интерпретациялау және келесі оқу әрекетін нақтылауға дейінгі өзара байланысты циклдік үдеріс ретінде көрсетеді. Модельдің құрылымын педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық бес компонент құрайды. Компоненттердің барлығы модельдің орталық нысаны болып табылатын студенттің программалау әрекетімен байланыстырылған. Бұл құрылымда оқытушы педагогикалық бақылау мен шешім қабылдаудың негізгі субъектісі ретінде, ал GenAI/LLM бақылау деректерін талдауды, мониторингті және кері байланысты қолдайтын көмекші интеллектуалдық құрал ретінде қарастырылады.



Сурет 2 – Программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі

Модельдің орталық ядросы – студенттің программалау әрекеті. Мұндай құрылым программалаудан білімнің дайын программалық өніммен ғана емес, оны құру барысында орындалатын өзара байланысты әрекеттерден көрінуімен негізделеді. Студент тапсырманы орындау кезінде есептің мазмұнын түсінеді, шешу тәсілін жоспарлайды, алгоритм құрады, оны программалық кодта жүзеге асырады, программаны тестілейді, анықталған қателерді түзетеді және қабылданған шешімді түсіндіреді. Сондықтан модельде программалау әрекеті шартты түрде «түсіну → жоспарлау → кодтау → тестілеу → түзету → түсіндіру» циклі арқылы беріледі. Бұл әрекеттер барысында студенттің білімінің байқалу белгілері және интеллектуалдық бақылауға қажетті цифрлық деректер қалыптасады.

Педагогикалық-мақсаттық компонент бақылаудың педагогикалық бағытын анықтайды және «нені және не үшін бақылаймыз?» деген сұраққа жауап береді. Оның құрамына бақылау мақсаты, бақылау нысаны және бағаланатын білім мазмұны кіреді. Бақылау мақсаты студенттің программалаудан білім жағдайын анықтауға бағытталады. Бақылау нысаны – студенттің программалау тапсырмаларын орындау барысындағы білімінің программалау әрекетінде көрінуі. Бағаланатын білім мазмұны программалау курсының негізгі мазмұндық компоненттерін, соның ішінде программалау тұжырымдамаларын, алгоритмдерді, программалау конструкцияларын, деректер құрылымдарын

және оларды есептерді шешуде қолдануды қамтиды. Осы компонент модельдің кейінгі компоненттері үшін бастапқы педагогикалық негіз қалыптастырады.

Критерийлік-бағалау компоненті студенттің программалаудан білімін қандай негізде бағалау керектігін анықтайды. Компонент бағалау критерийлерін, бағалау көрсеткіштерін және деңгейлік сипаттамаларды біріктіреді. Зерттеудің 1.3 бөлімінде негізделген тапсырма мазмұнын түсіну, алгоритмдік шешім, функционалдық дұрыстық, код сапасы, қатені анықтау және түзету, шешімді түсіндіру және негіздеу критерийлері студент білімінің программалау әрекетінде байқалатын алты қырын сипаттайды. Көрсеткіштер әр критерийдің студент әрекетінде байқалу белгілерін нақтылайды, ал деңгейлік сипаттамалар сол көрсеткіштердің байқалу дәрежесін ажыратуға мүмкіндік береді. Осылайша, білімді бақылау бір ғана «дұрыс/қате» нәтижесіне емес, программалау әрекетінің бірнеше педагогикалық маңызды сипаттарына сүйенеді.

Деңгейлік-тапсырма компоненті бағалау көрсеткіштерін қандай программалау тапсырмалары арқылы анықтау керектігін белгілейді. Оның құрамына программалау тапсырмалары, тапсырмалардың күрделілік деңгейлері және тапсырмалар мен бағалау көрсеткіштерінің байланысы кіреді. Тапсырмалар студенттің тиісті білімін практикалық әрекетте көрсетуіне мүмкіндік беретіндей іріктеліп, мазмұны мен орындалу күрделілігі бойынша сараланады. Бір тапсырманы орындау барысында бірнеше бағалау көрсеткіші байқалуы мүмкін, ал белгілі бір көрсеткіш бірнеше тапсырма бойынша жинақталған деректер негізінде анықталуы ықтимал. Сондықтан бұл компонент критерийлік-бағалау негіздерін студенттің нақты программалау әрекетімен байланыстыратын қызмет атқарады.

Деңгейлік-тапсырма компонентінен студентке берілген тапсырма модельдің орталық ядросындағы программалау әрекетін іске қосады. Осы әрекет барысында программалық код, тестілеу нәтижелері, жіберілген қателер мен оларды түзету әрекеттері, код нұсқалары, тапсырманы орындау тарихы, алгоритмдік түсіндірмелер, қосымша контекст және GenAI қолданылған жағдайда онымен өзара әрекеттесу деректері қалыптасады. Бұл деректер студент білімінің дайын бағасы ретінде емес, интеллектуалдық талдау үшін бастапқы ақпараттық негіз ретінде қарастырылады. Мұндай ұстаным студенттің соңғы программалық нәтижесімен қатар, сол нәтижеге жету үдерісін де бақылауға мүмкіндік береді.

Интеллектуалдық-талдау компоненті «қандай деректерді аламыз және оларды қалай талдаймыз?» деген сұраққа жауап береді. Компонент төрт өзара байланысты бөлікті қамтиды: цифрлық орталар мен құралдар; программалау әрекетінен алынатын цифрлық деректер; интеллектуалдық талдау тәсілдері; ақпараттық белгілер. Цифрлық орталар студенттің тапсырманы орындауын және оқытушының бақылауын қамтамасыз етеді. Статикалық талдау программалық кодтың құрылымы мен сапалық сипаттарын, динамикалық талдау программаның орындалуы мен тест нәтижелерін анықтайды; student modeling және knowledge tracing студенттің жинақталған әрекеттері негізінде білім жағдайындағы өзгерістерді уақыт бойынша модельдеуге мүмкіндік береді; GenAI/LLM-негізіндегі талдау кодты, түсіндірмені және әрекет деректерін мазмұндық тұрғыдан талдауды қолдайды.

Интеллектуалдық талдаудың нәтижесінде кодтық, әрекеттік, нәтижелік және мазмұндық ақпараттық белгілер анықталады. Кодтық белгілер программалық шешімнің құрылымы мен сапасын; әрекеттік белгілер тапсырманы орындау, қатені анықтау және түзету әрекеттерін; нәтижелік белгілер программаның орындалуы мен тестілеу нәтижелерін; мазмұндық белгілер есепті түсіну, алгоритмдік шешімді түсіндіру және негіздеу сипаттарын көрсетеді. Ақпараттық белгі бағалау көрсеткішімен автоматты түрде теңестірілмейді: бір көрсеткішті сипаттау үшін бірнеше дерек көзінен алынған белгілер бірге пайдаланылуы мүмкін. Сондықтан интеллектуалдық талдау компонентінің нәтижесі қорытынды баға емес, педагогикалық интерпретацияға арналған құрылымдалған ақпарат болып табылады.

Нәтижелік-интерпретациялық компонент интеллектуалдық талдаудан алынған ақпаратты педагогикалық мәні бар бақылау нәтижесіне айналдырады. Ол ақпараттық белгілерді бағалау көрсеткіштерімен сәйкестендіруді, көрсеткіштердің байқалу деңгейін анықтауды, студенттің программалаудан білім жағдайын белгілеуді, педагогикалық интерпретацияны және кері байланысты қамтиды. Әртүрлі дерек көздерінен алынған белгілер критерийлік-бағалау компонентінде белгіленген көрсеткіштер мен деңгейлік сипаттамаларға сәйкестендіріледі. Соның нәтижесінде студенттің бір ғана қорытынды балы емес, программалаудан білімінің қалыптасқан және қосымша жетілдіруді қажет ететін тұстарын көрсететін құрылымдалған білім жағдайы анықталады.

Модельде оқытушының рөлі барлық компоненттерді қамтитын педагогикалық бақылау және басқару контуры арқылы көрсетілген. Оқытушы тапсырмаларды іріктейді және ұйымдастырады, критерийлер мен рубриканы белгілейді, бақылау үдерісін бағыттайды, алынған нәтижелерді интерпретациялайды және студентке кері байланыс береді. Осыған байланысты модель толық автоматтандырылған бағалау жүйесін білдірмейді: интеллектуалдық технологиялар бақылау ақпаратын жинау мен талдауды қолдағанымен, қорытынды педагогикалық интерпретация мен шешім қабылдау оқытушының құзыретінде қалады.

GenAI/LLM модельдің бес негізгі компонентінің бірі емес, олардың жұмысын қолдайтын көлденең интеллектуалдық мониторинг құралы ретінде қарастырылады. Ол программалық код пен тест нәтижелерін талдауға, студенттің ЖИ қолдану сипатын анықтауға, критерийлер бойынша алдын ала аналитикалық ақпарат қалыптастыруға, жекелендірілген кері байланыс дайындауға және оқытушыға аналитикалық қолдау көрсетуге қатысады [131]. GenAI нәтижелері студент білімінің дайын бағасы ретінде қабылданбайды; олар белгіленген критерийлер мен көрсеткіштерге сәйкестендіріліп, оқытушының педагогикалық бақылауы жағдайында пайдаланылады. Осылайша, GenAI студенттің орнына программалау әрекетін орындаушы немесе оқытушының орнына қорытынды шешім қабылдаушы емес, бақылаушы және көмекші интеллектуалдық құрал қызметін атқарады.

Модельдің циклдік сипаты нәтижелік-интерпретациялық компоненттен деңгейлік-тапсырма компонентіне бағытталған кері байланыс арқылы қамтамасыз етіледі. Студенттің анықталған білім жағдайы, қиындықтары және жетілдіруді қажет ететін көрсеткіштері келесі программалау тапсырмасын таңдау, оның күрделілігін нақтылау немесе кері байланысты даралау кезінде

ескеріледі. Сондықтан модельдің жұмыс логикасын «педагогикалық мақсат → критерийлер мен көрсеткіштер → деңгейлік программалау тапсырмасы → студенттің программалау әрекеті → цифрлық деректер → интеллектуалдық талдау → ақпараттық белгілер → көрсеткіштердің байқалу деңгейі → білім жағдайы → педагогикалық интерпретация → кері байланыс → келесі тапсырма» тізбегімен сипаттауға болады.

Осылайша, ұсынылған модельде студенттің программалау әрекеті бақылаудың орталық ядросы, бес компонент осы әрекетті мақсатты ұйымдастыру, бағалау, деректерін талдау және нәтижелерін интерпретациялау механизмдері, оқытушы педагогикалық бақылау мен шешім қабылдаудың негізгі субъектісі, ал GenAI/LLM барлық бақылау циклін ақпараттық-аналитикалық тұрғыдан қолдайтын көмекші және мониторингтік құрал болып табылады. Мұндай құрылым студенттің программалаудан білімін тек программаның соңғы нәтижесі арқылы емес, есепті түсінуден бастап шешімді негіздеуге дейінгі әрекеттерден алынған әртекті деректерді өзара байланыстыра талдау негізінде анықтауға мүмкіндік береді. Осы модельді оқу үдерісінде іске асырудың нақты әдістері мен ұйымдастырушылық тетіктері зерттеудің 2.2 бөлімінде қарастырылады.

2.2 Интеллектуалдық модель негізінде программалаудан студенттердің білімін бақылауды ұйымдастыру әдістемесі

Зерттеу жұмысының 2.1 бөлімінде программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысы ретінде негізделді. Сондықтан бұл бөлімде модель компоненттерінің мазмұнын қайталау емес, олардың программалау пәніндегі практикалық, зертханалық және жобалық жұмыстарды бақылау барысында өзара әрекеттесу тәртібі қарастырылады.

Интеллектуалдық модельді оқу үдерісінде қолдану техникалық бақылау құралын таңдаудан емес, педагогикалық мақсатты анықтаудан басталады. Оқытушы алдымен студенттің қандай білім мазмұны мен программалау әрекеті бақыланатынын белгілейді. Осы мақсатқа сәйкес модельде белгіленген бағалау критерийлері мен көрсеткіштері таңдалып, олардың студент әрекетінде көрінуіне мүмкіндік беретін программалау тапсырмасы анықталады. Осы тұрғыдан бақылау тапсырмасы өздігінен бағалау құралы ретінде емес, алдын ала белгіленген көрсеткіштер бойынша студенттің білім жағдайын сипаттайтын деректерді алуға мүмкіндік беретін диагностикалық құрал ретінде қарастырылады.

Тапсырманы орындау барысында студент программалық код, тестілеу нәтижелері, қате хабарламалары, кодтың өзгертілген нұсқалары, қайта тестілеу нәтижелері, алгоритмдік түсіндірме және басқа да цифрлық деректер қалыптастырады. GenAI қолданылған жағдайда онымен өзара әрекеттесу фрагменттері де шешімнің қалыптасу үдерісін сипаттайтын қосымша дерек көзі ретінде пайдаланылуы мүмкін. Алайда бұл деректер студент білімінің дайын бағасы болып табылмайды. Олар автоматты тестілеу, статикалық және динамикалық талдау, үдерістік аналитика және қажет болған жағдайда LLM-

негізіндегі семантикалық талдау арқылы өңделіп, бағалау көрсеткіштерін сипаттайтын ақпараттық белгілерге айналады.

Мысалы, программаның негізгі тесттерден өту нәтижесі бастапқы бақылау дерегі болып табылады. Оны талдау арқылы программаның берілген кіріс жағдайларында дұрыс жұмыс істеу дәрежесін сипаттайтын ақпараттық белгі алынады. Бұл белгі функционалдық дұрыстық көрсеткішімен сәйкестендіріліп, белгіленген деңгейлік сипаттамалар негізінде интерпретацияланады. Сол сияқты студенттің алгоритмді таңдау себебін түсіндіруі оның алгоритмдік шешімін және шешімді түсіндіруі мен негіздеуін сипаттайтын ақпараттық белгілерді анықтауға мүмкіндік береді. Осылайша бір тапсырманы орындау барысында алынған әртекті деректер тиісті көрсеткіштермен сәйкестендірілу арқылы студенттің программалаудан білімінің бірнеше қырын сипаттайды.

Интеллектуалдық талдау нәтижесінде алынған ақпараттық белгілер критерийлер мен көрсеткіштер жүйесі арқылы педагогикалық тұрғыдан интерпретацияланып, студенттің программалаудан білім жағдайы анықталады. Мұнда бақылау нәтижесі бір ғана қорытынды балмен шектелмейді: студенттің қай көрсеткіштер бойынша жеткілікті деңгейге жеткені, қандай көрсеткіштер бойынша қиындықтары бар екені және оларды жетілдіру үшін қандай келесі оқу әрекеті қажет екені айқындалады.

Бақылау анықталған нәтижені тіркеумен аяқталмайды. Студенттің білім жағдайы диагностикалық кері байланыс қалыптастыруға және келесі тапсырманы таңдауға негіз болады. Мысалы, студенттің программасы дұрыс жұмыс істегенімен, алгоритмді таңдауын негіздеуде қиындық анықталса, келесі тапсырма синтаксистік білімді қайталауға емес, бірнеше алгоритмдік шешімді салыстыруға және олардың ішінен тиімдісін таңдап, оны негіздеуге бағытталады. Осылайша кері байланыс нәтижелік-интерпретациялық компонентті деңгейлік-тапсырма компонентімен қайта байланыстырып, модельдің циклдік сипатын қамтамасыз етеді.

Демек, интеллектуалдық модель негізінде программалаудан студенттердің білімін бақылауды ұйымдастырудың жалпы әдістемелік логикасы «педагогикалық мақсат → критерийлер мен көрсеткіштер → тапсырма → студент әрекетінен алынған деректер → интеллектуалдық талдау → ақпараттық белгілер → көрсеткіштер бойынша деңгейді анықтау → білім жағдайын педагогикалық интерпретациялау → кері байланыс → келесі тапсырманы таңдау» реттілігімен сипатталады. Бұл реттілік модельдің жаңа компоненттерін білдірмейді, керісінше 2.1 бөлімінде негізделген бес компоненттің оқу үдерісінде өзара байланыста іске асу механизмін көрсетеді.

Енді программалаудан студенттердің білімін бақылауды ұйымдастыру әдістерін интеллектуалдық модельді программалау тапсырмасында іске асыру мысалында қарастырайық.

2.1-бөлімде негізделген интеллектуалдық модельдің жұмыс істеу принципін келесі есеп арқылы көрсетейік. Берілген a, b, c сандары үшін a^b mod c мәнін есептеу қажет, мұнда $1 \le a, b, c < 2^{63}$. Үлкен b мәндерінде a^b санын тікелей есептеу тиімсіз болғандықтан, тапсырма студенттің есеп шартын түсінуін, тиімді алгоритм таңдауын, оны программалық кодта жүзеге асыруын, тестілеуін, қатені түзетуін және шешімді негіздеуін кешенді бақылауға мүмкіндік береді.

Педагогикалық-мақсаттық компонентте, бақылау мақсаты – студенттің модульдік арифметика мен жылдам дәрежелеу алгоритмін түсінуін және қолдана алуын анықтау ретінде белгіленеді. Бұл жерде тек 3^2 mod 4 = 1 нәтижесін алу емес, студенттің неліктен басқа алгоритм қажет екенін түсінуі бақыланады. Бақылау нысаны – дайын нәтиже ғана емес, студенттің есепті шешу әрекеті. Мысалы, студент дайын pow(a,b,c) функциясын қолданып дұрыс жауап алуы мүмкін. Бірақ бұл оның модульдік жылдам дәрежелеудің логикасын меңгергенін өздігінен дәлелдемейді. Сондықтан студенттің алгоритмді түсіндіруі де бақылауға алынады. Білім мазмұнына – дәрежелеу, қалдық алу операциясы, цикл, шартты оператор, екілік жіктеу және алгоритмнің уақыттық күрделілігі кіреді.

Критерийлік-бағалау компонентінде, тапсырма мазмұнын түсіну, алгоритмдік шешім, функционалдық дұрыстық, код сапасы, қатені анықтау және түзету, шешімді түсіндіру және негіздеу критерийлері қолданылады. Мысалы, «алгоритмдік шешім» критерийінің көрсеткіші – студенттің жылдам модульдік дәрежелеу алгоритмін дұрыс таңдап, оның тиімділігін негіздей алуы. Бағалау көрсеткіштері төмен, орта және жоғары деңгейлер арқылы сипатталады. Көрсеткіштердің байқалу деңгейі. Мысалы, «Алгоритмдік шешім» критерийін алайық. Төмен деңгей: студент a^b-ны тікелей немесе b рет көбейту арқылы есептеуге әрекет жасайды және үлкен b үшін мәселені ескермейді. Орта деңгей: жылдам дәрежелеуді қолданады, бірақ оның неліктен тиімді екенін немесе алгоритмнің қадамдарын толық негіздей алмайды. Жоғары деңгей: жылдам модульдік дәрежелеуді дұрыс қолданады, дәрежені екілік жіктеу логикасын түсіндіреді және алгоритмнің күрделілігінің O(logb) екенін негіздейді.

Осы есепте критерийлердің көрсеткіштері былайша нақтыланады: тапсырма мазмұнын түсіну – студент a^b мәнін тікелей есептеудің неге тиімсіз екенін түсінеді; алгоритмдік шешім – жылдам модульдік дәрежелеу алгоритмін таңдап, оның логикасын дұрыс құрады; функционалдық дұрыстық – программа негізгі және шеттік тестілерде дұрыс нәтиже береді; код сапасы – код құрылымды, түсінікті және артық әрекеттерсіз жазылған; қатені анықтау және түзету – қате нәтиженің себебін анықтап, кодты түзетеді және қайта тексереді; шешімді түсіндіру және негіздеу – алгоритмнің жұмысын және оның тиімділігін түсіндіреді.

Деңгейлік-тапсырма компонентінде осы көрсеткіштердің байқалуына мүмкіндік беретін тапсырма ұйымдастырылады. Студент есептің шектеулерін түсінуі, алгоритмді таңдауы, оның кодын іске асыруы, тестілеуі және тиімділігін негіздеуі тиіс. Бастапқы деңгейде кіші мәндер үшін a^b mod c есептеледі; орта деңгейде үлкен b үшін b рет көбейтпей есептеу талап етіледі; жоғары деңгейде студенттен тек код емес, неліктен O(b) тәсілінің орнына O(logb) алгоритмі таңдалғанын түсіндіру талап етіледі. Сонда бір ғана шешімнен алгоритм таңдау, функционалдық дұрыстық, код сапасы, тестілеу, қатені түзету және шешімді негіздеу туралы деректер алынады.

Студенттің программалау әрекеті «түсіну → жоспарлау → кодтау → тестілеу → түзету → түсіндіру» реттілігімен жүзеге асады. Студент алдымен 1<=a,b,c<2^63 шектеуін талдайды және a^b-ны толық есептеудің тиімсіз екенін, сонымен бірге модульдік арифметиканың (xy) mod c = ((x mod c)(y mod c))

қасиетін пайдалануға болатынын анықтайды. Жоспарлауда жылдам модульдік дәрежелеу алгоритмін таңдайды. Негізгі идея: result = 1; a = a mod c; егер b тақ болса, нәтижені ағымдағы a-ға көбейту; a-ны квадраттау; b-ны екіге бөлу; b=0 болғанша қайталау, яғни келесі алгоритмді кодтайды:

```
def mod_pow(a, b, c):
    result = 1
    a %= c
    while b > 0:
        if b % 2 == 1:
            result = (result * a) % c
        a = (a * a) % c
        b //= 2
    return result
```

Мұнда студенттің кодты жай көшіріп жазуы емес, әр оператордың қызметін түсінуі маңызды. Тестілеу сатысында 3 2 4 → 1, 2 10 1000 → 24 т.с.с негізгі және шеттік тестілермен тексереді, қате анықталса, оның себебін тауып кодты түзетеді және қайта тестілейді. Соңында модуль операциясының қолданылуын және алгоритмнің O(log b) күрделілігін, неліктен a^b-ны тікелей есептемейміз?, әр көбейтуден кейін % c қолдануға болады?, b-ның тақ немесе жұп болуы алгоритмге қалай әсер етеді?, b әр қадамда екіге бөлінеді?, алгоритмнің уақыттық күрделілігі қандай?, c=1 болған жағдайда нәтиже неліктен әрқашан 0? деген сұрақтарға жауап студенттің дайын кодты пайдаланғанын емес, алгоритмді түсінгенін анықтауға көмектеседі.

Интеллектуалдық-талдау компонентінде программалық бастапқы код, кодтың аралық нұсқалары, тестілік жағдайлар, тест нәтижелері, қате хабарламалары, кодқа енгізілген түзетулер, қайта тестілеу нәтижелері, студенттің алгоритмдік түсіндірмесі және GenAI қолданылған жағдайда өзара әрекеттесу деректері жинақталады. Интеллектуалдық талдау тәсілдері: автоматтандырылған тестілеу программа әртүрлі a,b,c мәндері үшін дұрыс нәтиже бере ме?, статикалық талдау – кодтың құрылымы қандай, қайталанатын немесе артық әрекеттер бар ма?, динамикалық талдау – программа үлкен кіріс деректерінде уақыт шектеуінен өте ме?, үдерістік талдау – студент бастапқы қатеден дұрыс шешімге қалай келді? және LLM негізіндегі талдау – студенттің түсіндірмесі оның жазған кодына және алгоритм логикасына сәйкес келе ме?. Талдау нәтижесінде, «барлық бақылау тестінен өтті»; «b рет қайталанатын цикл қолданылған»; «жылдам дәрежелеуге көшкеннен кейін орындалу уақыты азайды»; «модуль операциясын дұрыс қолданды»; «қатені анықтап, кодты қайта тестіледі»; «O(log b) күрделілігін дұрыс түсіндірді» сияқты ақпараттық белгілер анықталады.

Нәтижелік-интерпретациялық компонентте ақпараттық белгілер тиісті бағалау көрсеткіштерімен сәйкестендіріліп, олардың байқалу деңгейі анықталады. Мысалы, барлық негізгі және шеттік тестілерден өту функционалдық дұрыстықты, O(log b) алгоритмін таңдау алгоритмдік шешімді, ал қатені анықтап түзету қатені анықтау және түзету көрсеткішін сипаттайды. Осы нәтижелердің жиынтығы студенттің білім жағдайын анықтауға және диагностикалық кері байланыс қалыптастыруға негіз болады.

Осылайша A^B mod C есебінде бақылау дұрыс сандық нәтижені алумен шектелмейді. Интеллектуалдық модель студенттің есепті түсінуінен бастап алгоритмді таңдау, кодтау, тестілеу, түзету және шешімді негіздеуге дейінгі әрекеттерін өзара байланыста бағалауға мүмкіндік береді. Бұл нәтижелер студенттің программалаудан білім жағдайының профилін қалыптастырады. Мұнда студентке жай ғана «85 балл» деген баға қоюдан гөрі, оның қай компонентті жақсы меңгергені және қай жерде қиындық бар екені анықталады. Мысалы: Студент есептің шектеулерін дұрыс түсініп, жылдам модульдік дәрежелеу алгоритмін таңдады және программаны функционалдық тұрғыдан дұрыс жүзеге асырды. Қатені анықтау және түзету әрекеті қалыптасқан. Алайда алгоритмнің уақыттық күрделілігін және таңдалған шешімнің қарапайым дәрежелеуден артықшылығын толық негіздеуде қиындық байқалады. Бұл қорытынды студенттің нақты білім жағдайын сипаттайды. Анықталған қиындыққа сәйкес келесі тапсырма таңдалып, модельдің кері байланыс циклі іске асырылады. Мысалы, студентке дайын алгоритмді қайта қолдану емес, O(B) және O(log B) алгоритмдерін салыстырып, олардың тиімділігін эксперименттік және теориялық тұрғыдан негіздеу ұсынылуы мүмкін.

Бұл мысал интеллектуалдық модельдің маңызды ерекшелігін нақты көрсетеді: дұрыс жауап бақылаудың соңғы мақсаты емес. Тапсырма нәтижесіне студент өзі құрған тиімді алгоритм арқылы да, дайын шешімді көшіріп алуы арқылы да жетуі мүмкін. Сондықтан интеллектуалдық бақылауда нәтиженің дұрыстығы студенттің алгоритмді таңдау, кодтау, тестілеу, қатені түзету және шешімді негіздеу әрекеттері туралы деректермен бірге қарастырылады. Нәтижесінде студенттің программалаудан білімі бір қорытынды нәтижемен емес, бірнеше өзара байланысты бағалау көрсеткіштері бойынша сипатталады.

Қысқа диагностикалық тапсырмалар әдісі студенттің программалау бойынша жекелеген білімдері мен әрекеттерін анықтауға және тапсырманы орындауда кездесетін қиындықтарын нақтылауға бағытталады. Бұл әдіс жаңа тақырыпты меңгергеннен кейін, практикалық жұмыстың басында немесе студенттің программалау тапсырмасын орындауда қандай қиындыққа тап болғанын анықтау қажет болған жағдайда қолданылады.

Бұл әдісте студентке көлемді программаны толық әзірлеу міндетті емес. Код фрагментінің нәтижесін анықтау, белгілі бір программалық конструкцияның қызметін түсіндіру, алгоритмнің жетіспейтін бөлігін толықтыру, қатені табу, тестілік жағдай құрастыру сияқты қысқа тапсырмалар беріледі. Осындай тапсырмалар студенттің нақты қай білім немесе әрекет бойынша қиындыққа тап болғанын анықтауға мүмкіндік береді.

Мысалы, студент толық программалау есебін орындай алмаған жағдайда қиындықтың шартты операторды қолданумен, цикл логикасын түсінумен немесе есепті шешу алгоритмін құрумен байланысты екенін анықтау қажет болуы мүмкін. «Циклдік алгоритмдер» тақырыбында студентке for және while циклдерінің қызметін салыстыру, қай жағдайда қайталану саны алдын ала белгілі болатынын түсіндіру және берілген есепке сәйкес циклді таңдауды негіздеу ұсынылады. Алынған жауаптар студенттің циклдік конструкциялардың қызметін түсінуі және оларды есеп шартына сәйкес таңдай алуын анықтауға мүмкіндік береді.

Қысқа диагностикалық тапсырмалардың бір түрі ретінде кодты оқу және талдау тапсырмалары да қолданылады. Студент дайын функцияның қызметін, белгілі кіріс деректері бойынша алынатын нәтижені, қолданылған конструкциялардың рөлін, кодтағы ықтимал қателер мен тиімсіз тұстарды анықтайды. Бұл студенттің программаны өздігінен жазу қабілетімен қатар, дайын программалық кодты түсіну және талдау деңгейін анықтауға мүмкіндік береді.

Мысалы, Python программалау тілінде қайталанатын элементтерді алып тастап, олардың алғашқы кездесу ретін сақтайтын unique_values() функциясы студентке дайын код түрінде ұсынылады. Студенттен функцияның қызметін түсіндіру, [2, 1, 2, 3, 1] кіріс деректері үшін [2, 1, 3] нәтижесінің қалай қалыптасатынын қадам бойынша көрсету, үлкен көлемдегі тізім үшін қолданылған алгоритмнің тиімділігін бағалау және балама шешім ұсыну талап етіледі.

```
def unique_values(a):
    result = []
    for x in a:
        if x not in result:
            result.append(x)
    return result
```

Мұндай тапсырма дайын кодтың орындалу нәтижесін анықтаумен ғана шектелмей, оның логикасын түсінуді, алгоритмдік шешімді талдауды және оны негіздеуді бақылауға мүмкіндік береді. Осылайша бір код фрагменті негізінде студенттің программалаудан білімін сипаттайтын бірнеше көрсеткіш бойынша бақылау деректері алынады.

Қысқа диагностикалық тапсырмалардың түрлері, олар арқылы бақыланатын білім мен әрекеттер, алынатын негізгі деректер және сәйкес бағалау критерийлері 6-кестеде жүйеленді.

Кесте 6 – Қысқа диагностикалық тапсырмалар арқылы программалаудан білімді бақылау мазмұны

Қысқа диагностикалық тапсырма	Бақыланатын білім/әрекет	Негізгі дерек	Сәйкес бағалау критерийі
Шешімді түсіндіру және негіздеу	Кодтың орындалу логикасын түсіну	Жауап, түсіндірме	Пәндік-теориялық түсіну
Кодты толықтыру	Программалық конструкцияны орынды қолдану	Код фрагменті	Алгоритмдік шешім
Қатені табу	Қатенің себебі мен салдарын түсіну	Қате орны, түсіндірме	Қатені анықтау және түзету
Тест құру	Қалыпты және шеттік жағдайларды ажырату	Тесттер, күтілетін нәтиже	Функционалдық дұрыстық
Код нәтижесін болжау және түсіндіру	Кодтың орындалу қадамдарын түсіндіру, нәтиженің қалай	Болжанған нәтиже және студенттің түсіндірмесі	Шешімді түсіндіру және негіздеу

6-кестенің жалғасы

Қысқа диагностикалық тапсырма	Бақыланатын білім/әрекет	Негізгі дерек	Сәйкес бағалау критерийі
	қалыптасатынын негіздеу		

Қысқа диагностикалық тапсырмалар студенттің жекелеген білімдері мен әрекеттерін нақтылауға мүмкіндік берсе, күрделілік деңгейлері бойынша сараланған программалау тапсырмалары бірнеше көрсеткіштің бір тапсырма аясында кешенді көрінуін қамтамасыз етеді. Сондықтан программалаудан білімді бақылауды ұйымдастыруда тапсырмалардың күрделілік деңгейімен қатар студенттің шешімді қалыптастыру үдерісін бақылау да ескеріледі.

Авторлардың бұған дейінгі зерттеуінде программалаудан білімді цифрлық бақылауды ұйымдастыруда шешімді қадамдық тексеру, көпдеңгейлі тапсырмалар және бейімделген кері байланыс тәсілдері қарастырылған [112]. Қадамдық тексеру студенттің есепті шешу барысын кезеңдер бойынша бақылауға, көпдеңгейлі тапсырмалар оның білімін біртіндеп күрделенетін есептер арқылы анықтауға, ал бейімделген кері байланыс дайын шешімді ұсынбай, анықталған қиындыққа сәйкес бағыттаушы ақпарат беруге мүмкіндік береді. Аталған тәсілдер интеллектуалдық модельдің деңгейлік-тапсырма компонентін практикалық іске асыруда ескерілді.

Программалаудан білімді деңгейлік бақылау әдісі студентке ұсынылатын тапсырмалардың күрделілігін біртіндеп арттыру арқылы оның программалаудан білімін кешенді бақылауға негізделеді. Мұнда тапсырмалардың күрделілігі тек код жолдарының саны немесе есеп шартының көлемімен анықталмайды. Ол тапсырманы орындауға қажетті білім мен әрекеттердің саны және өзара байланысы, алгоритмді таңдау еркіндігі, шеттік жағдайларды ескеру және қабылданған шешімді негіздеу қажеттілігі арқылы анықталады.

Бірінші деңгейдегі тапсырмалар программалаудың базалық конструкцияларын тікелей қолдануды талап етеді. Екінші деңгейдегі тапсырмаларда бірнеше конструкцияны бір алгоритм аясында байланыстырып қолдану қажет болады. Үшінші деңгейдегі тапсырмаларда студент есепті шешу тәсілін өздігінен таңдап, алынған шешімнің дұрыстығы мен тиімділігін негіздейді.

Бұл деңгейлер студенттің білім деңгейін тікелей білдірмейді, олар бақылау тапсырмаларын күрделілігі бойынша ұйымдастыру деңгейлері болып табылады. Студенттің нақты білім жағдайы әртүрлі деңгейдегі тапсырмаларды орындау барысында көрсеткіштер бойынша алынған нәтижелердің жиынтығы негізінде анықталады.

Программалаудан білімді деңгейлік бақылауды ұйымдастыруда әр деңгейге сәйкес тапсырмалардың мазмұны, олар арқылы бақыланатын білім мен әрекеттер, басым бағалау көрсеткіштері және алынатын бақылау деректері өзара сәйкестендіріледі. Осы сәйкестік негізінде деңгейлік бақылауға арналған программалау тапсырмаларының мазмұны 7-кестеде берілді.

Кесте 7 – Программалаудан білімді деңгейлік бақылауға арналған тапсырмалардың мазмұны

Деңгей	Программалау тапсырмасы	Бақыланатын білім мен әрекет	Негізгі критерийлер	Бақылау деректері
I	Санның жұп немесе тақ екенін анықтау	Шартты операторды және қалдық операциясын қолдану	Пәндік-теориялық түсіну; функционалдық дұрыстық	Алгоритмдік шешім; функционалдық дұрыстық
I	Үш санның максимумын табу	Шартты конструкцияны және функцияны қолдану	Пәндік-теориялық түсіну; функционалдық дұрыстық	Алгоритмдік шешім; функционалдық дұрыстық
I	Сан цифрларын кері шығару	Цикл мен арифметикалық амалдарды қолдану	Алгоритмдік шешім; функционалдық дұрыстық	Код; тестілеу нәтижелері
II	Массивтегі тақ сандардың қосындысын табу	Массив, цикл және шартты конструкцияны байланыстырып қолдану	Алгоритмдік шешім; функционалдық дұрыстық; код сапасы	Код; тестілеу нәтижелері
II	Палиндромды тексеру	Цикл мен салыстыру амалдарын байланыстырып қолдану	Алгоритмдік шешім; функционалдық дұрыстық; шешімді түсіндіру және негіздеу	Код; тестілеу нәтижелері; түсіндірме
II	Санның бөлгіштерін табу	Цикл мен қалдық операциясын пайдаланып алгоритм құру	Алгоритмдік шешім	Код; тестілеу және орындалу нәтижелері
III	Цифрлардан палиндром құру мүмкіндігін анықтау	Элементтердің жиілігін анықтау және шешім шартын құру	Алгоритмдік шешім; шешімді түсіндіру және негіздеу	Код; тестілеу нәтижелері; түсіндірме
III	Ең жиі кездесетін элементті анықтау	Массивпен және жиілік құрылымымен жұмыс істеу тәсілін таңдау	Алгоритмдік шешім; код сапасы	Код; тестілеу нәтижелері
III	Келесі үлкен санды құру	Іздеу және сұрыптау әрекеттерін байланыстырып, шешу тәсілін таңдау	Алгоритмдік шешім; шешімді түсіндіру және негіздеу	Код; орындау үдерісі; түсіндірме

7-кестеде берілген тапсырмалардың күрделілік деңгейлері олардың мазмұны мен студенттен талап етілетін программалау әрекеттеріне байланысты ажыратылады. Мысалы, I деңгейдегі «Санның жұп немесе тақ екенін анықтау» есебінде студент қалдық операциясы мен шартты операторды қолданады. Мұнда студенттің санның жұп немесе тақ болуын анықтау принципін түсінуі және оны программалық конструкциялар арқылы дұрыс жүзеге асыра алуы бақыланады. Сондықтан бұл деңгейде алгоритмнің тиімділігін күрделі талдау негізгі бақылау міндеті болып табылмайды.

II деңгейдегі «Массивтегі тақ сандардың қосындысын табу» есебі массив элементтерін цикл арқылы қарауды, шартқа сәйкес элементтерді анықтауды және олардың қосындысын есептеуді талап етеді. Бірнеше программалық конструкция бір алгоритм аясында байланыстырылатындықтан, функционалдық дұрыстықпен қатар алгоритмдік шешім мен код сапасын да бақылауға мүмкіндік туады.

III деңгейдегі «Ең жиі кездесетін элемент» есебінде дұрыс нәтижеге әртүрлі алгоритмдік тәсілдер арқылы қол жеткізуге болады. Мысалы, студент әр элементтің жиілігін бірнеше рет есептеуі, сұрыптауды қолдануы немесе сөздік (dictionary/hash map) құрылымын пайдалануы мүмкін. Сондықтан бұл деңгейде дұрыс нәтиже алумен қатар таңдалған алгоритмдік шешімнің тиімділігі және оны негіздеу қабілеті де ескеріледі.

Деңгейлік бақылау әдісі студенттің қай күрделілік деңгейіндегі тапсырмаларда қиындыққа кездесетінін анықтауға мүмкіндік береді. Мысалы, студент I деңгейдегі тапсырмаларды тұрақты орындап, II деңгейдегі тапсырмаларда қиналса, бұл оның жекелеген программалық конструкцияларды қолдана алғанымен, оларды біртұтас алгоритмде байланыстыруда қиындықтары бар екенін көрсетуі мүмкін. Ал III деңгейдегі тапсырманы орындаудағы қиындық базалық білімнің жеткіліксіздігін міндетті түрде білдірмейді; ол алгоритмді таңдау, оның тиімділігін бағалау, шешімді негіздеу немесе шеттік жағдайларды ескеру әрекеттерімен байланысты болуы мүмкін.

Нәтижелік және үдерістік деректер негізінде бақылау әдісі студенттің программалау есебін орындау нәтижесімен қатар, шешімді қалыптастыру барысында жасаған әрекеттерін бақылауға бағытталады. Әртүрлі кезеңдерде жиналған деректерден тиісті бағалау критерийлерінің көрсеткіштерін сипаттайтын ақпараттық белгілер анықталады.

Мысалы, программаның барлық ұсынылған тесттерден өтуі оның тексерілген жағдайлардағы функционалдық дұрыстығын көрсетеді, бірақ студенттің алгоритмді түсінгенін өздігінен дәлелдемейді. Ал алгоритмді дұрыс түсіндіру оның кодта қатесіз жүзеге асырылғанын білдірмейді. Сондықтан нәтижелік және үдерістік деректер студент білімінің әртүрлі қырларын сипаттайтын өзара толықтырушы деректер ретінде қарастырылады.

Әдістің практикалық іске асуын бірнеше квадраттың жиынтық ауданын горизонталь сызық арқылы тең екі бөлікке бөлетін (y) координатасын анықтау есебі арқылы қарастыруға болады. Әр квадрат ((x, y, l)) параметрлерімен беріледі. Студент квадраттың бөлуші сызықтан толық төмен, толық жоғары немесе сызықпен қиылысатын жағдайларын ажыратып, ауданның бөлуші сызық координатасына байланысты өзгеруін модельдеуі қажет.

Мысалы, бөлуші сызықтың (h) координатасы берілген жағдайда студент әр квадраттың осы сызықтан төмен орналасқан бөлігінің ауданын есептейтін функция құрады. Егер квадрат толығымен сызықтан жоғары орналасса, оның төменгі бөлікке қосатын ауданы нөлге тең; квадрат толығымен сызықтан төмен орналасса, оның толық ауданы есепке алынады; ал сызық квадратты қиып өтсе, тек сызықтан төмен орналасқан бөлігінің ауданы есептеледі.

```
def area_below(squares, h):
    area = 0.0
    for x, y, l in squares:
        if h <= y:
            continue
        elif h >= y + l:
            area += l * l
        else:
            area += (h - y) * l
    return area
```

Есепті орындау үдерісі бірнеше кезең бойынша бақыланады. Бірінші кезеңде студент есеп шартын формалдайды: кіріс деректерін, анықталатын нәтижені, есептеу дәлдігін және шеттік жағдайларды белгілейді. Екінші кезеңде ауданның бөлуші сызық координатасына тәуелді өзгеру сипатын анықтап, бинарлық іздеу (binary search) алгоритмін қолдануды негіздейді. Квадраттардың жалпы ауданын есептеп, оның жартысын мақсатты мән ретінде анықтайды. Бинарлық іздеудің әр қадамында mid мәні үшін сызықтан төмен орналасқан аудан есептеліп, оның мақсатты мәннен үлкен немесе кіші болуына байланысты іздеу шекарасы өзгертіледі.

Үшінші кезеңде таңдалған алгоритм программалық код түрінде жүзеге асырылады. Төртінші кезеңде студент бір квадрат, бірнеше квадрат, қабаттасатын квадраттар, бөлуші сызық квадраттың ішінен өтетін жағдай және үлкен координаталар үшін тесттер құрады. Бесінші кезеңде тестілеу барысында қате анықталған жағдайда оның себебін анықтап, кодқа өзгеріс енгізеді және қайта тестілеу жүргізеді. Алтыншы кезеңде ауданның есептелу тәсілін, іздеу шекараларын, тоқтау шартын және алгоритмнің есептеу күрделілігін түсіндіреді.

Бұл жағдайда бақылау программаның соңғы сандық жауабын тексерумен шектелмейді. Студенттен $h <= y$, $h >= y + l$ және $y < h < y + l$ жағдайларының геометриялық мағынасын түсіндіру, ауданның (h)-қа тәуелді өзгеруі неліктен бинарлық іздеуді қолдануға мүмкіндік беретінін негіздеу және алгоритмнің тоқтау шартын түсіндіру талап етіледі.

Осы үдерісте программаның соңғы коды мен тестілеу нәтижелері нәтижелік деректер ретінде, ал есепті формалдау нәтижелері, алгоритмді таңдау негіздемесі, тесттер, қате туралы ақпарат, кодтың өзгертілген нұсқалары, қайта тестілеу нәтижелері және студенттің түсіндірмелері үдерістік деректер ретінде жинақталады.

Интеллектуалдық талдау барысында бұл деректерден тиісті бағалау көрсеткіштерін сипаттайтын ақпараттық белгілер анықталады. Программаның тесттерден өту нәтижелері функционалдық дұрыстықты, есепті модельдеу мен алгоритмді таңдау алгоритмдік шешімді, есептеу күрделілігін талдау

тиімділікті, қателерді анықтап кодқа өзгеріс енгізу қатені анықтау және түзетуді, ал қабылданған шешімді негіздеу шешімді түсіндіру мен негіздеуді сипаттайды.

Осылайша нәтижелік және үдерістік деректер негізінде бақылау студенттің тек соңғы программалық нәтижесін ғана емес, есеп шартын түсінуден бастап алгоритмді таңдау, программалау, тестілеу, қатені түзету және шешімді негіздеуге дейінгі әрекеттерін қамтиды. Бұл әрекеттердің әр кезеңінде қалыптасатын бақылау деректері тиісті бағалау критерийлерінің көрсеткіштерімен сәйкестендіріледі. Программалау тапсырмасын орындау кезеңдері, әр кезеңде жиналатын бақылау деректері және сәйкес бағалау критерийлері 8-кестеде берілген.

Кесте 8 – Программалау тапсырмасын орындау кезеңдері бойынша бақылау деректері мен бағалау критерийлері

Кезең	Студент әрекеті	Жиналатын бақылау дерегі	Сәйкес бағалау критерийлері
Тапсырма мазмұнын түсіну	Кіріс және шығыс деректерін, есеп шарттары мен шектеулерін анықтау	Есеп шартының формалданған сипаттамасы; мәтіндік түсіндірме	Тапсырма мазмұнын түсіну
Алгоритмді таңдау	Шешу тәсілін таңдау және негіздеу	Алгоритм сипаттамасы; псевдокод; таңдау негіздемесі	Алгоритмдік шешім
Кодтау	Алгоритмді программалық код түрінде жүзеге асыру	Бастапқы программалық код	Код сапасы
Тестілеу	Қалыпты және шеттік жағдайларға арналған тесттерді құру және орындау	Тестілік жағдайлар; тестілеу нәтижелері	Функционалдық дұрыстық
Қатені түзету	Қатені анықтау, кодқа өзгеріс енгізу және қайта тестілеу	Қате туралы ақпарат; кодтағы өзгерістер; қайта тестілеу нәтижелері	Қатені анықтау және түзету
Түсіндіру және негіздеу	Шешім логикасын, алгоритмді таңдау себебін және оның тиімділігін түсіндіру	Жазбаша немесе ауызша түсіндірме	Шешімді түсіндіру және негіздеу; алгоритмдік шешім

Программалық кодты тестілеу, қатені анықтау және түзету әдісі. Бұл әдіс студенттің программалық шешімнің дұрыстығын тексеру, қатені анықтау, оның себебін түсіндіру және кодты түзету әрекеттерін бақылауға бағытталады. Интеллектуалдық бақылауда тестілеу (testing) мен қатені анықтау және түзету (debugging) программаның жұмысын техникалық тексерумен ғана шектелмей, студенттің программалау барысында орындайтын пәндік әрекеттері ретінде қарастырылады.

Тестілеу барысында дайын тестілік жағдайларды қолданумен қатар, оларды студенттің өзі құруы да маңызды. Дайын тесттер программаның белгілі

кіріс деректері үшін дұрыс жұмыс істейтінін тексеруге мүмкіндік береді, бірақ студенттің есеп шартынан тексерілуі тиіс жағдайларды өздігінен анықтай алуын толық сипаттамайды. Сондықтан студенттің қалыпты және шеттік жағдайларды анықтауы, оларға сәйкес кіріс деректерін таңдауы және күтілетін нәтижені көрсетуі бақылау әрекетінің құрамына енгізіледі.

Бұл тұрғыдан тест арқылы әзірлеу – Test-Driven Development (TDD) логикасының жекелеген элементтерін оқу тапсырмаларына бейімдеп қолдануға болады [132]. Мұнда жұмыс «тест құру → программалық шешімді іске асыру → тестілеу → кодты жетілдіру» реттілігімен ұйымдастырылады. Студент алдымен есеп шартына сәйкес тестілік жағдайларды анықтайды, содан кейін осы тесттерден өтетін программалық кодты әзірлейді және тестілеу нәтижелеріне сәйкес оны жетілдіреді. Мұндай ұйымдастыру студенттің есеп шартын тестілік жағдайларға түрлендіруін, шеттік жағдайларды алдын ала ескеруін және программалық шешімді тест нәтижелеріне сүйеніп жетілдіруін бақылауға мүмкіндік береді.

Мысалы, берілген бүтін санның жай сан екенін анықтайтын is_prime(n) функциясы үшін студент теріс сан, 0, 1, 2, жай сан және құрама сан жағдайларын қамтитын тесттер құрады. Әр тест үшін тексерілетін жағдай мен күтілетін нәтиже көрсетіледі. Бұл жерде тесттердің саны емес, олардың есеп шартын және ықтимал шеттік жағдайларды қаншалықты қамтитыны маңызды.

Қатені анықтау және түзету әрекетін бақылау үшін студентке қатесі бар программалық код та ұсынылуы мүмкін. Мысалы:

```
def average(values):
    total = 0
    for i in range(len(values) - 1):
        total += values[i]
    return total / len(values)
```

Берілген кодта цикл соңғы элементті қосындыға қоспайды. Студент тестілеу арқылы нәтиженің дұрыс еместігін анықтап, қатенің range(len(values) - 1) өрнегіне байланысты екенін түсіндіреді, кодты түзетеді және қайта тестілеу жүргізеді.

Бұл жағдайда бақылау «бастапқы программалық код → тестілеу нәтижесі → қатені анықтау → қатенің себебін түсіндіру → кодты түзету → қайта тестілеу → түзетуді негіздеу» реттілігімен жүзеге асырылады. Бастапқы және түзетілген код, тестілік жағдайлар, тестілеу нәтижелері, анықталған қате, кодқа енгізілген өзгерістер және студенттің түсіндірмесі бақылау деректері ретінде жинақталады.

Мұнда кодқа енгізілген өзгерістердің саны өздігінен студент білімінің көрсеткіші болып табылмайды. Бақылау үшін студенттің қатені дұрыс анықтауы, оның себебін түсіндіруі, тиісті түзету енгізуі және түзетілген программалық шешімнің дұрыстығын қайта тестілеу арқылы тексеруі маңызды. Осы деректер функционалдық дұрыстықты, қатені анықтау және түзетуді, сондай-ақ шешімді түсіндіру мен негіздеуді сипаттайтын ақпараттық белгілерді анықтауға мүмкіндік береді.

Юнит-тестілеуді ұйымдастырудың қарапайым мысалы ретінде Java тіліндегі Calculator класын қарастыруға болады:

```
public class Calculator {
```

```
    public int add(int a, int b) {
        return a + b;
    }
}
```

add() әдісінің жұмысын тексеру үшін JUnit көмегімен келесі тест құрылады:

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;

public class CalculatorTest {
    @Test
    public void testAdd() {
        Calculator calc = new Calculator();
        int result = calc.add(2, 3);
        assertEquals(5, result);
    }
}
```

Мұнда @Test аннотациясы әдістің тест ретінде орындалатынын, ал assertEquals(expected, actual) күтілетін және нақты нәтижелердің сәйкестігін тексеруді қамтамасыз етеді. Тест арқылы әзірлеу логикасын қолданғанда студент алдымен күтілетін нәтижені сипаттайтын тест құрып, одан кейін сол талапты қанағаттандыратын программалық шешімді әзірлейді.

Интеллектуалдық бақылау тұрғысынан тесттің орындалу нәтижесімен қатар студенттің тестілік жағдайды неліктен таңдағаны, қандай нәтижені күткені және тест орындалмаған жағдайда қатенің себебін қалай анықтап, кодты қалай түзеткені де маңызды. Сондықтан бақылау студенттің соңғы дұрыс кодымен ғана шектелмей, «тест → нәтиже → қатені анықтау → түзету → қайта тестілеу → түзетуді негіздеу» үдерісін қамтиды.

Жобалық программалау жұмысын кезеңдік бақылауда программалық код нұсқалары мен оларға енгізілген өзгерістердің тарихы студент әрекетінің динамикасын сипаттайтын маңызды дерек көзі болып табылады. Жобаны орындау барысында сақталған кодтың бастапқы, аралық және соңғы нұсқалары студенттің программалық шешімді қалай қалыптастырғанын, анықталған қателерді қалай түзеткенін және бастапқы шешімді қалай жетілдіргенін бақылауға мүмкіндік береді.

Бұл ретте кодқа енгізілген өзгерістердің саны өздігінен студент білімінің көрсеткіші болып табылмайды. Олардың диагностикалық маңызы өзгерістің мазмұны, оны енгізу себебі және алынған нәтижемен байланысы арқылы анықталады. Мысалы, тестілеу барысында анықталған қатеден кейін студенттің тиісті код бөлігін өзгертіп, қайта тестілеу жүргізуі оның қатені анықтау және түзету әрекетін сипаттайтын деректерді қалыптастырады. Ал программалық кодтың құрылымын жетілдіруге бағытталған өзгерістер код сапасын сипаттайтын қосымша ақпарат көзі ретінде пайдаланылуы мүмкін.

Жобалық жұмысты бақылауда студенттің жұмыс барысындағы әрекеттері мен жобаның қорытынды нәтижесі өзара байланыста қарастырылады. Жұмыс барысында программалық кодтың аралық нұсқалары, тестілеу нәтижелері, анықталған қателер және кодқа енгізілген өзгерістер бақыланса, жұмыстың

соңында дайын программалық өнімнің функционалдық дұрыстығы, код сапасы және студенттің қабылданған шешімдерді түсіндіруі мен негіздеуі бағаланады.

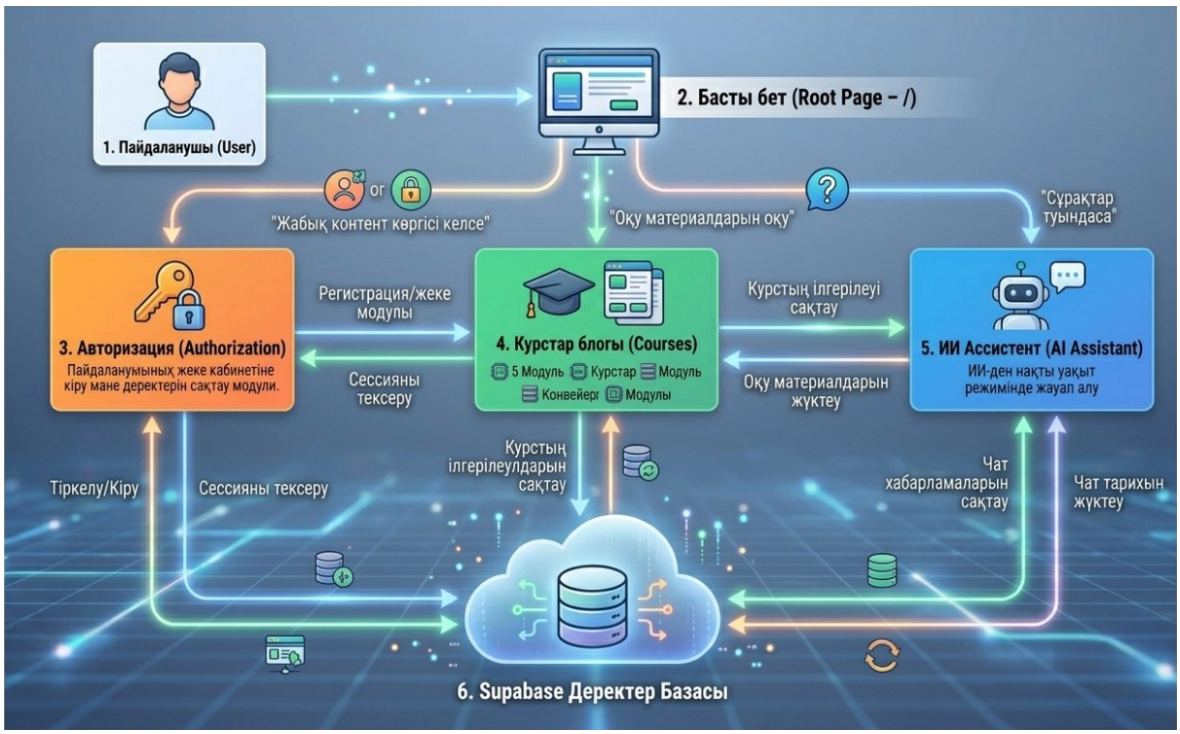
Мұндай кезеңдік бақылаудың қажеттілігі қорытынды программалық өнімдері ұқсас студенттердің оны әзірлеу үдерісі әртүрлі болуы мүмкіндігімен түсіндіріледі. Бір студент программалық шешімді өздігінен жобалап, тестілік жағдайларды құрып, анықталған қателерді кезең-кезеңімен түзете алады, ал екінші студент дайын шешімдерді біріктіргенімен, олардың жұмыс істеу логикасын жеткілікті негіздей алмауы мүмкін. Тек соңғы программалық өнімге негізделген бақылау бұл айырмашылықтарды толық көрсетуге мүмкіндік бермейді.

Жобалық программалау жұмысын орындау барысында қалыптасатын негізгі бақылау деректері мен олардың бағалау көрсеткіштерімен байланысы KomekArch жобасының мысалында нақтыланып, 9-кестеде жүйеленді.

KomekArch толықстекті программалық жобасын кезеңдік бақылау мысалы ретінде қарастырайық. Жоба клиенттік және серверлік бөліктерден, мәліметтер қорынан және олардың өзара әрекеттесуін қамтамасыз ететін программалық құралдардан тұрады. Жобаның архитектурасы пайдаланушы интерфейсін, авторизация модулін, курстар блогын, ЖИ-ассистентті және Supabase мәліметтер қорын қамтиды (3-сурет). Жобаны орындауда мәліметтер қорын жобалау, серверлік API құру, аутентификация мен қауіпсіздік механизмдерін ұйымдастыру, клиенттік және серверлік бөліктерді байланыстыру, код өзгерістерінің тарихын сақтау және дайын өнімді орналастыру әрекеттері бақыланады.

Жобалық жұмысты 3–4 студенттен тұратын шағын топтарда ұйымдастыруға болады. Топ мүшелері жобаны жоспарлау, архитектурасын әзірлеу және программалық іске асыру міндеттерін бөліседі. Әр студенттің білімін бағалауда оның орындаған тапсырмалары, кодқа енгізген өзгерістері, тестілеу нәтижелері және қабылдаған шешімдеріне берген түсіндірмесі ескеріледі.

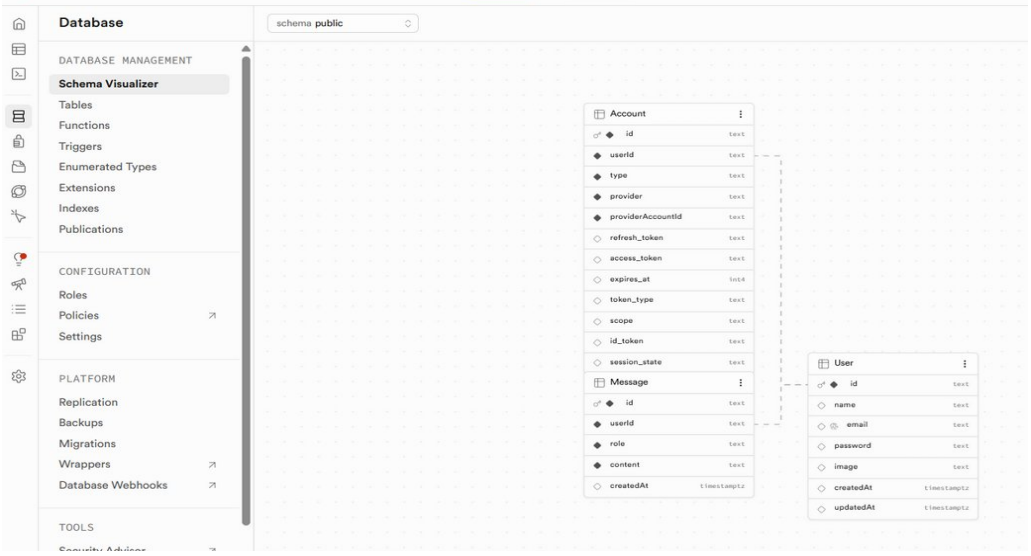
Жұмыс барысындағы деректер GitHub ортасында жинақталады. GenAI қолданылған жағдайда студент оны қай кезеңде, қандай мақсатпен пайдаланғанын, ұсынылған шешімді қалай тексергенін және қандай өзгерістер енгізгенін тіркейді. Бұл мәліметтер шешімнің қалыптасу барысын сипаттайтын қосымша деректер ретінде пайдаланылады. Жобаны қорғауда әр студент өз үлесін және қабылдаған шешімдерін түсіндіріп, негіздейді.



Сурет 3 – KomekArch толықстекті программалық жобасының архитектурасы

Осы бақылауды жобаның жекелеген тапсырмалары арқылы нақтылауға болады.

Мәліметтер қоры моделін жобалауды бақылау. Студентке User (пайдаланушы) және Message (хабарлама) объектілері арасындағы байланысты сипаттайтын реляциялық мәліметтер қоры моделін құру тапсырмасы беріледі. Модельде біріншілік кілт (primary key), сыртқы кілт (foreign key) және «бірден-көпке» (One-to-Many) қатынасы қолданылады. Бақылау модельдің техникалық дұрыстығымен қатар, студенттің объектілер арасындағы байланысты түсіндіруі мен құрылымдық таңдауын негіздеуін қамтиды (4-сурет).

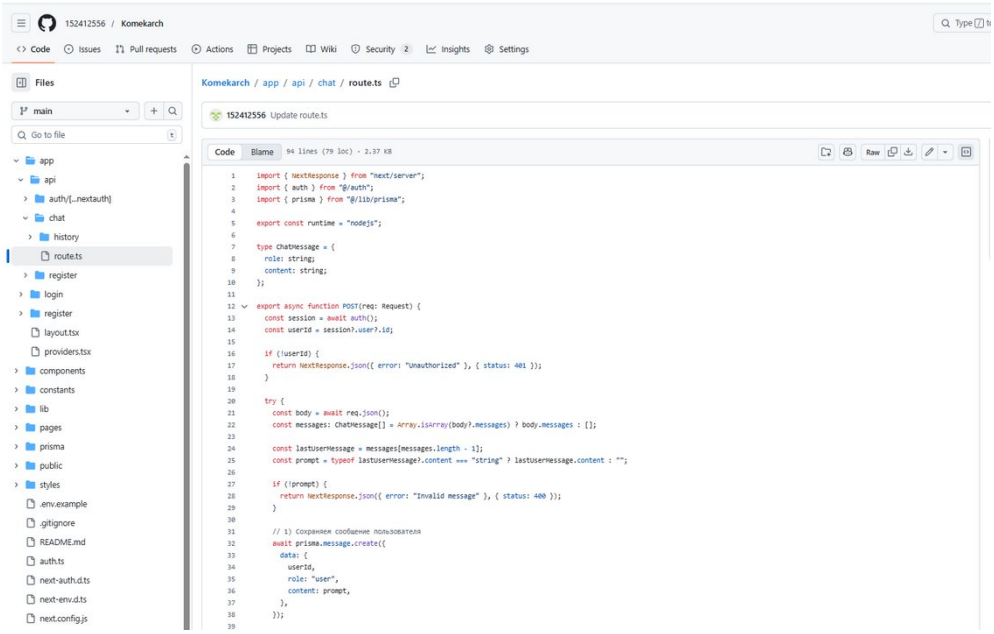


Сурет 4 – Жобалық тапсырмадағы мәліметтер қоры моделін бақылау мысалы

Бұл кезеңде студенттің тапсырма талаптарын, сақталатын деректер мен олардың байланыстарын анықтауы тапсырма мазмұнын түсіну критерийі бойынша, ал құрылған мәліметтер қоры моделінің құрылымын түсіндіруі және таңдауын негіздеуі шешімді түсіндіру және негіздеу критерийі бойынша бағаланады.

Бақылау деректері ретінде мәліметтер қорының схемасы, schema.prisma файлы, программалық кодтың аралық нұсқалары мен оларға енгізілген өзгерістер және студенттің түсіндірмесі пайдаланылады.

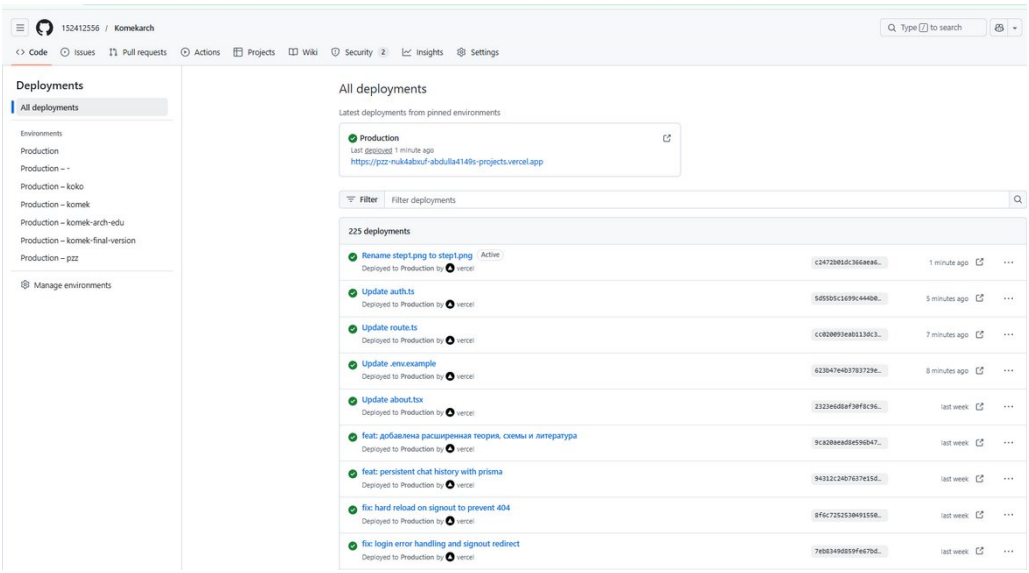
Серверлік API әзірлеуді бақылау. Келесі кезеңде студентке чат тарихын алу және сақтау үшін API маршруттарын әзірлеу тапсырмасы беріледі. Тапсырманы орындау барысында студент GET және POST әдістерін, JSON форматындағы деректер алмасуды, async/await асинхронды әрекеттерін, мәліметтер қорымен байланысты, аутентификацияны және HTTP күй кодтарын (HTTP status codes) қолданады. Бақылау API маршруттарының дұрыс жұмыс істеуін тексерумен ғана шектелмейді. Студенттен қолданылған HTTP күй кодтарының қандай жағдайларды білдіретінін, аутентификацияның қай кезеңде және қандай мақсатпен тексерілетінін, сондай-ақ қабылданған программалық шешімді түсіндіру және негіздеу талап етіледі (5-сурет).



Сурет 5 – Серверлік API кодын жобалық бақылау мысалы

GitHub цифрлық іздері арқылы үдерісті бақылау. Version control system (нұсқаларды басқару жүйесі) жобалық бақылаудың маңызды дерек көзін қалыптастырады. GitHub-тағы commit (коммит), code diff (кодтағы өзгеріс), branch (тармақ) және build status (жинақталу күйі) программалық шешімнің уақыт бойынша қалай өзгергенін көруге мүмкіндік береді. Коммит саны немесе жиілігі студент білімінің тікелей өлшемі ретінде қолданылмайды. Мысалы, он шағын коммит бір мазмұнды архитектуралық өзгерістен маңыздырақ деген қорытынды жасауға болмайды. Диагностикалық талдау «қандай проблема

анықталды -> қандай өзгеріс енгізілді -> тест нәтижесі қалай өзгерді -> студент өзгерісті түсіндіре ала ма?» байланысына сүйенеді (6-сурет).



Сурет 6 – GitHub ортасындағы жобалық программалау әрекетінің цифрлық іздері

CI/CD (Continuous Integration/Continuous Deployment – үздіксіз интеграция және орналастыру) нәтижелері де жобалық жұмыстың орындалу үдерісі туралы қосымша техникалық деректер береді. «Программалық кодтағы өзгеріс → программаны жинақтау → қатенің анықталуы → кодты түзету → сәтті жинақтау» реттілігі программалық өнімнің жұмыс күйіне қалай жеткізілгенін бақылауға мүмкіндік береді. Алайда программаның сәтті жинақталуы немесе орналастырылуы студенттің программалаудан білімінің барлық көрсеткіштері бойынша жоғары деңгейге жеткенін білдірмейді. Ол программалық өнімнің белгілі техникалық талаптарға сәйкес жұмыс істейтінін сипаттайтын бақылау деректерінің бірі ретінде қарастырылады.

Жобалық программалау жұмысын интеллектуалдық модель негізінде бақылауды жобаның орындалу логикасына сәйкес бес бірізді кезең бойынша ұйымдастыруға болады. Әр кезеңде студенттің белгілі бір әрекеттері бақыланып, сол әрекеттер барысында қалыптасқан деректер тиісті бағалау көрсеткіштерімен сәйкестендіріледі. Жобалық программалау жұмысын кезеңдік бақылаудың құрылымы 9-кестеде берілген. 8-кестеден айырмашылығы, мұнда бақылау бір программалау тапсырмасын орындау кезеңдері бойынша емес, ұзақ мерзімді жобалық жұмыстың кезеңдері бойынша ұйымдастырылады: бақылау деректері жобаның талаптары мен жоспарын, кодтың аралық нұсқаларын, CI/CD нәтижелерін және жобаны қорғау материалдарын қамтиды, ал бір кезеңде бірнеше бағалау критерийі қатар қолданылады.

Кесте 9 – Жобалық программалау жұмысын кезеңдік бақылау кезеңдері, бақылау деректері және бағалау критерийлері

Кезең	Студент әрекеті	Негізгі бақылау деректері	Сәйкес бағалау критерийлері
Мәселені қою	Проблеманы, мақсатты, күтілетін нәтижені және шектеулерді анықтау	Талаптардың сипаттамасы, бастапқы шешім тұжырымдамасы	Тапсырма мазмұнын түсіну; шешімді түсіндіру және негіздеу
Жоспарлау	Есепті формалдау, программалық шешімнің құрылымын және алгоритмдерін таңдау, жұмыс кезеңдерін жоспарлау	Жоспар, схема, алгоритм сипаттамасы, таңдалған шешімнің негіздемесі	Алгоритмдік шешім; шешімді түсіндіру және негіздеу
Іске асыру	Программалық кодты әзірлеу, қажетті функцияларды іске асыру және алгоритмнің тиімділігін ескере отырып кодты жетілдіру	Бастапқы программалық код, кодтың аралық нұсқалары мен оларға енгізілген өзгерістер, жинақтау нәтижелері	Функционалдық дұрыстық; код сапасы
Тестілеу	Тестілік жағдайларды құру және орындау, қатені анықтау, кодты түзету және қайта тестілеу	Тестілік жағдайлар, тестілеу нәтижелері, анықталған қателер, кодқа енгізілген өзгерістер, қайта тестілеу нәтижелері	Функционалдық дұрыстық; қатені анықтау және түзету
Жобаны қорғау	Программалық өнімді ұсыну, қабылданған шешімдерді түсіндіру, қолданылған алгоритмнің тиімділігін негіздеу және сұрақтарға жауап беру	Программалық өнімнің демонстрациясы, студенттің түсіндірмесі және сұрақтарға жауаптары	Шешімді түсіндіру және негіздеу

Кезеңдік бақылауда студенттің программалық өнімі мен оны әзірлеу барысындағы әрекеттері өзара байланыста бағаланады. Әр кезеңде бақылау нысаны, жиналатын деректер және сәйкес бағалау критерийлері нақтыланады.

Бұл бес кезеңді KomekArch жобасының мысалында көрсетуге болады. Мәселені қою кезеңінде студент жүйенің мақсаты мен функционалдық талаптарын анықтайды; жоспарлау кезеңінде User–Message мәліметтер моделін, API құрылымын және негізгі модульдерді жобалайды; іске асыру кезеңінде мәліметтер қорын, серверлік маршруттарды, аутентификация механизмдерін және интерфейсті әзірлейді; тестілеу кезеңінде қателерді анықтап, түзетеді және нәтижені қайта тексереді; жобаны қорғауда қабылдаған архитектуралық және алгоритмдік шешімдерін түсіндіреді және негіздейді.

GenAI ұсынған программалық шешімді тексеру және талдау арқылы бақылау әдісі. Генеративті жасанды интеллект (GenAI) құралдары программалау тапсырмасына дайын код немесе алгоритмдік шешім ұсына алатындықтан, студенттің білімін бақылауды тек GenAI құралын қолданған

немесе қолданбаған фактісі негізінде ұйымдастыру жеткіліксіз. Интеллектуалдық модельде GenAI-мен өзара әрекеттесу студенттің программалаудан білімінің жеке бағалау көрсеткіші ретінде емес, программалық шешімнің қалыптасу контексін сипаттайтын қосымша дерек көзі ретінде қарастырылады. Сондықтан бақылаудың негізгі нысаны – студенттің GenAI ұсынған программалық шешімді түсінуі, тексеруі, талдауы, қажет болған жағдайда түзетуі және енгізілген өзгерістерді негіздеуі.

Осы мақсатта GenAI ұсынған программалық шешімді тексеру және талдау тапсырмасы қолданылады. Студентке GenAI қалыптастырған дайын программалық код ұсынылады. Студент кодтың жұмыс логикасын түсіндіреді, оны әртүрлі тестілік жағдайларда тексереді, алгоритмдік және программалық қателерді анықтайды, қосымша шеттік жағдайларға арналған тесттер құрады. Қате анықталған жағдайда кодқа тиісті өзгерістер енгізіп, қайта тестілеу жүргізеді және енгізілген өзгерістің себебін түсіндіреді. Осылайша GenAI ұсынған дайын шешім студенттің жауабы ретінде емес, оның программалаудан білімін бақылауға арналған талдау объектісі ретінде пайдаланылады.

Бұл әдісті негіздеуге авторлардың Codeforces программалау есептері негізінде жүргізген алдыңғы зерттеуінің нәтижелері мүмкіндік берді [***]. Зерттеуде GenAI модельдері қалыптастырған программалық шешімдер тестілік жағдайлар арқылы тексеріліп, сыртқы құрылымы дұрыс және есептің шешіміне сәйкес көрінетін кодтың барлық тестілік жағдайлардан міндетті түрде өте бермейтіні анықталды. Қарастырылған мысалдардың бірінде ChatGPT қалыптастырған шешімде A массивін анықтауға байланысты қате анықталып, кодқа тиісті түзету енгізілгеннен кейін ғана программа барлық тестілік жағдайлардан өтті. Бұл GenAI қалыптастырған программалық шешімді дайын дұрыс жауап ретінде қабылдамай, оның дұрыстығын тестілеу, қатесін анықтау және шешім логикасын талдау қажеттігін көрсетті.

Оқу үдерісінде мұндай тапсырма студенттің GenAI ұсынған кодты механикалық қабылдауын емес, оны программалау біліміне сүйене отырып тексеруін талап етеді. Бақылау барысында GenAI ұсынған бастапқы код, студент құрастырған тестілік жағдайлар, тестілеу нәтижелері, анықталған қателер, кодқа енгізілген өзгерістер, қайта тестілеу нәтижелері және студенттің түсіндірмесі бақылау деректері ретінде жинақталады. Бұл деректер функционалдық дұрыстық, алгоритмдік шешім, қатені анықтау және түзету, тиімділік, код сапасы және шешімді түсіндіру мен негіздеу көрсеткіштері бойынша студенттің білім жағдайын сипаттауға мүмкіндік береді.

Осыған сәйкес бақылау «GenAI ұсынған шешім → студенттің талдауы → тестілеу → қатені анықтау → түзету → қайта тестілеу → шешімді негіздеу» реттілігімен ұйымдастырылады. Мұнда GenAI ұсынған бастапқы шешімнің сапасы студенттің бағасын тікелей анықтамайды. Бақылау үшін студенттің программалық шешімді түсінуі, қатені анықтауы, оның себебін түсіндіруі, тиісті түзету енгізуі, түзетудің дұрыстығын қайта тестілеу арқылы тексеруі және қабылданған шешімді негіздеуі маңызды. Осы әрекеттер нәтижесінде алгоритмдік шешім, функционалдық дұрыстық, қатені анықтау және түзету, шешімді түсіндіру және негіздеу критерийлерінің көрсеткіштерін сипаттайтын ақпараттық белгілер қалыптасады.

Алдын ала жүргізілген зерттеу нәтижелері GenAI-дың тек қате программалық шешімнің көзі ғана емес, есепті шешудің тиімді алгоритмдік тәсілдерін де ұсына алатынын көрсетті. Мысалы, (x^2-1) өрнегінің ең кіші бөлгішін анықтауға арналған есепте толық іздеудің орнына (x^2-1=(x-1)(x+1)) түрлендіруін пайдалану есептеу көлемін азайтатын шешім ретінде ұсынылған. Мұндай жағдайда бақылаудың мақсаты GenAI ұсынған дайын шешімді қабылдау емес, студенттің қолданылған математикалық түрлендіруді түсіндіруі, бастапқы және ұсынылған алгоритмдік тәсілдерді салыстыруы және олардың тиімділік айырмашылығын негіздеуі болып табылады. Осылайша GenAI ұсынған дұрыс шешімнің өзі студенттің алгоритмдік шешім мен тиімділікті түсінуін және қабылданған шешімді негіздеуін бақылауға арналған тапсырма ретінде пайдаланылуы мүмкін.

GenAI-мен өзара әрекеттесу барысында қалыптасатын сұраныстар (prompts), алынған жауаптар және кодқа енгізілген өзгерістер де қосымша бақылау деректері ретінде тіркелуі мүмкін. Алайда сұраныстар саны, GenAI жауабын пайдалану көлемі немесе кодқа енгізілген өзгерістер саны өздігінен студент білімінің деңгейін білдірмейді және тікелей бағалау көрсеткіші ретінде қолданылмайды. Олардың педагогикалық мағынасы студенттің GenAI ұсынған шешімді қалай талдағаны, тексергені, өзгерткені және қабылданған шешімді қаншалықты негіздей алғаны туралы қосымша контекстік және үдерістік ақпарат беруімен анықталады.

Әртекті бақылау деректерін интеллектуалдық талдау әдісі 2.1-бөлімде негізделген интеллектуалдық-талдау компонентін практикалық тұрғыдан іске асырады. Программалық код пен оның орындалу нәтижелері автоматтандырылған тестілеу, статикалық және динамикалық талдау арқылы зерттеледі. Код өзгерістері мен түзету әрекеттері үдерістік талдау арқылы, ал студенттің түсіндірмелері мен негіздемелері LLM-негізіндегі семантикалық талдау арқылы қарастырылады.

Талдаудан алынған ақпараттық белгілер тиісті бағалау критерийлерінің көрсеткіштерімен сәйкестендіріліп, педагогикалық тұрғыдан түсіндіріледі. Мысалы, кодтың жоғары құрылымдық күрделілігі студент білімінің төмендігін өздігінен білдірмейді. LLM қорытындылары да бағалаудың жалғыз негізі ретінде қолданылмай, басқа бақылау деректерімен салыстырылады және қажет болған жағдайда оқытушы тарапынан тексеріледі.

Бірнеше тапсырма бойынша жинақталған нәтижелер студенттің білім жағдайын және оның уақыт бойынша өзгеруін сипаттауға пайдаланылады. Бақылау деректерінің түрлері, оларды талдау тәсілдері, алынатын ақпараттық белгілер және сәйкес бағалау критерийлері 10-кестеде берілген.

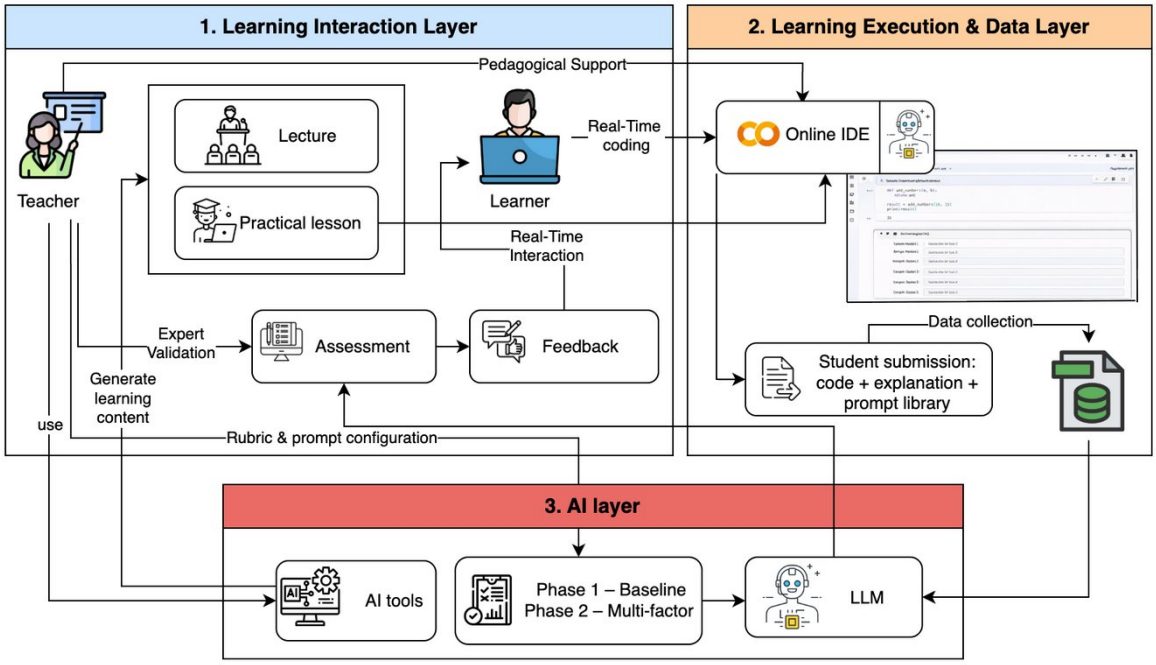
Кесте 10 – Бақылау деректерін талдау тәсілдері және олардың бағалау критерийлерімен байланысы

Бақылау деректері	Талдау тәсілі	Алынатын ақпараттық белгілер	Байланысты бағалау көрсеткіштері
Программалық код, тестілік жағдайлар, күтілетін және нақты нәтижелер	Автоматтан дырылған тестілеу	Тесттерден өту нәтижесі, дұрыс және қате нәтижелер, шеттік жағдайларды өңдеу	Функционалдық дұрыстық
Программалық код	Статикалық талдау	Код құрылымы, күрделілігі, қайталануы, модульділігі және ұйымдастырылуы туралы белгілер	Код сапасы
Программаның орындалу нәтижелері, тестілік деректер	Динамикалық талдау	Орындалу уақыты, жадты пайдалану, орындалу тұрақтылығы және орындалу кезіндегі қателер	Алгоритмдік шешім; функционалдық дұрыстық
Программалық кодтың аралық нұсқалары, тестілеу және қайта тестілеу нәтижелері, анықталған қателер мен енгізілген өзгерістер	Үдерістік талдау	Қатенің анықталуы, оған сәйкес енгізілген түзету, түзетуден кейінгі нәтиженің өзгеруі	Қатені анықтау және түзету
Программалық код, алгоритмдік түсіндірме, студенттің жауаптары мен негіздемелері	LLM-негізіндегі семантикалық талдау	Код пен түсіндірменің мағыналық сәйкестігі, алгоритм логикасының түсіндірілуі, қабылданған шешімнің негізделуі	Алгоритмдік шешім; шешімді түсіндіру және негіздеу

Программалау сабағында интеллектуалдық бақылауды ұйымдастырудың практикалық жұмыс үдерісі. Интеллектуалдық модельді нақты оқу үдерісінде іске асыру барысында студенттің программалық коды, алгоритмдік түсіндірмесі, тестілеу нәтижелері, кодқа енгізілген өзгерістері және GenAI қолданылған жағдайда онымен өзара әрекеттесу деректері жинақталады. Мұнда студенттің соңғы программалық нәтижесімен қатар, шешімді қалыптастыру, тексеру, түзету және негіздеу барысында орындалған әрекеттері де ескеріледі.

Жиналған бақылау деректері олардың ерекшелігіне сәйкес автоматтандырылған тестілеу, статикалық және динамикалық талдау, үдерістік және LLM-негізіндегі семантикалық талдау арқылы өңделеді. Алынған ақпараттық белгілер тиісті бағалау көрсеткіштерімен сәйкестендіріліп, студенттің программалаудан білім жағдайын педагогикалық тұрғыдан интерпретациялауға негіз болады. Бақылау нәтижелері кері байланыс қалыптастыру және келесі оқу тапсырмасын анықтау үшін пайдаланылады, осылайша интеллектуалдық бақылаудың циклдік сипаты қамтамасыз етіледі.

Интеллектуалдық модельді программалау сабағында іске асыру барысында оқытушының педагогикалық әрекеті, студенттің тапсырманы орындау үдерісі, бақылау деректерін жинау және оларды интеллектуалдық талдау өзара байланыста ұйымдастырылады. Осы үдерістің негізгі қатысушылары, қалыптасатын бақылау деректері және оларды өңдеу құралдарының өзара байланысы 7-суретте көрсетілген.



Сурет 7 – Программалау сабағында интеллектуалдық бақылауды ұйымдастырудың жұмыс ағыны

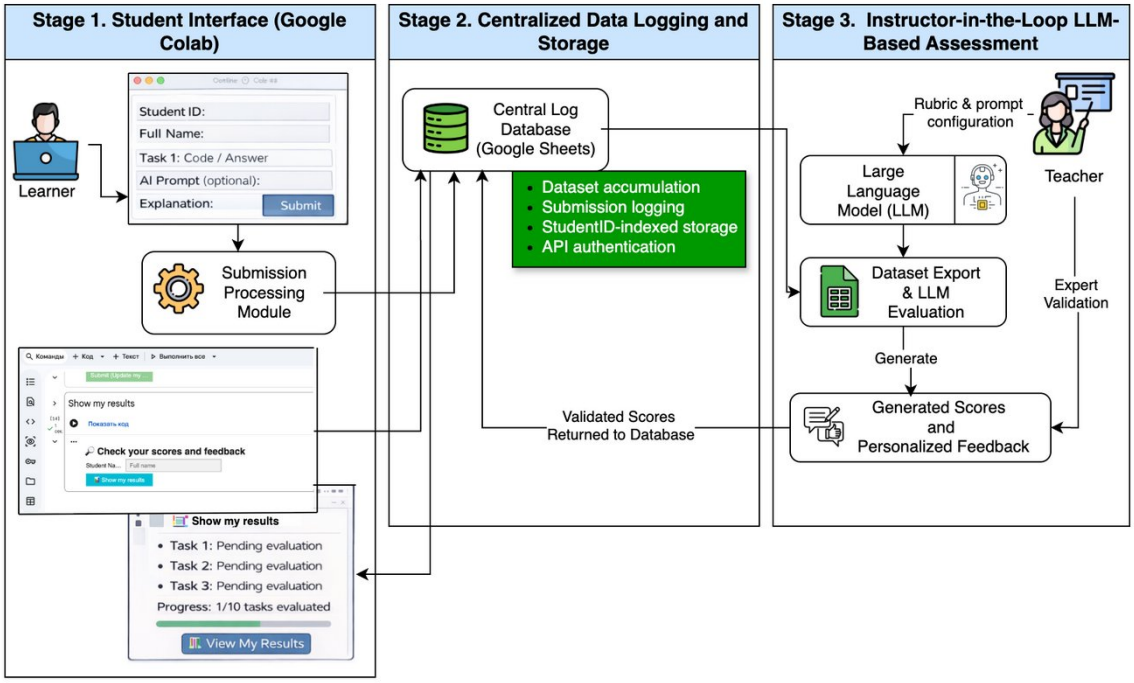
7-суретте интеллектуалдық модельді программалау сабағында іске асырудың жұмыс ағыны көрсетілген. Жұмыс ағыны өзара байланысты үш блокты қамтиды.

Бірінші блок бақылауды педагогикалық тұрғыдан ұйымдастырумен байланысты. Оқытушы оқу мақсатын анықтап, студенттің бақыланатын білімі мен программалау әрекеттерін белгілейді, оларға сәйкес бағалау көрсеткіштері мен программалау тапсырмасын таңдайды. Тапсырманың күрделілік деңгейі бақылау мақсатына сәйкес анықталады. Егер тапсырманы немесе оған қатысты материалдарды дайындауда GenAI қолданылса, алынған мазмұн студентке ұсынылғанға дейін оқытушы тарапынан тексеріледі.

Екінші блок студенттің программалау тапсырмасын орындау үдерісін қамтиды. Студент тапсырманы орындау барысында программалық код әзірлейді, тестілік жағдайларды қолданады немесе құрастырады, анықталған қателерді түзетеді және қабылданған алгоритмдік шешімді түсіндіреді және негіздейді. Осы әрекеттер барысында программалық код, тестілеу нәтижелері, кодтың аралық нұсқалары, анықталған қателер, енгізілген түзетулер және алгоритмдік түсіндірмелер сияқты бақылау деректері қалыптасады. GenAI қолданылған жағдайда онымен өзара әрекеттесу деректері программалық шешімнің қалыптасу үдерісін сипаттайтын қосымша дерек көзі ретінде тіркелуі мүмкін.

Үшінші блок жиналған бақылау деректерін интеллектуалдық талдау және олардың нәтижелерін педагогикалық интерпретациялаумен байланысты. Бақылау деректері олардың ерекшелігіне сәйкес автоматтандырылған тестілеу, статикалық және динамикалық талдау, үдерістік және LLM-негізіндегі семантикалық талдау арқылы өңделеді. Талдау нәтижесінде алынған ақпараттық белгілер тиісті бағалау көрсеткіштерімен сәйкестендіріледі. Олардың жиынтығы студенттің программалаудан білім жағдайын анықтауға, нәтижелерді педагогикалық тұрғыдан интерпретациялауға және диагностикалық кері байланыс қалыптастыруға негіз болады.

Интеллектуалдық бақылауда студенттің программалау әрекетінен алынған деректерді жинау, оларды LLM көмегімен талдау және алынған нәтижені оқытушының тексеруі өзара байланысты кезеңдер ретінде ұйымдастырылуы мүмкін. Студенттің бастапқы жауабынан диагностикалық кері байланыс қалыптастыруға дейінгі осы жұмыс үдерісі 8-суретте көрсетілген.



Сурет 8 – LLM-негізіндегі талдау мен оқытушының тексеруін қамтитын интеллектуалдық бақылаудың жұмыс үдерісі

8-суретте интеллектуалдық бақылауды ұйымдастырудың үш кезеңі көрсетілген. Бірінші кезеңде студент программалау тапсырмасын орындап, программалық кодын, алгоритмдік түсіндірмесін және GenAI қолданылған жағдайда онымен өзара әрекеттесу деректерін ұсынады. Екінші кезеңде тапсырманы орындау барысында қалыптасқан бақылау деректері жинақталып, кейінгі талдау үшін сақталады. Үшінші кезеңде жинақталған деректер LLM-негізіндегі семантикалық талдауға беріліп, алынған нәтижелер алдын ала белгіленген бағалау көрсеткіштерімен сәйкестендіріледі.

LLM арқылы алынған нәтиже студент білімінің дайын бағасы ретінде қабылданбайды. Оқытушы қажет болған жағдайда студенттің бастапқы программалық кодын, түсіндірмесін және талдау нәтижелерін салыстырып, жүйе қалыптастырған қорытындыны тексереді және нақтылайды. Тексерілген

нәтижелер студенттің білім жағдайын педагогикалық тұрғыдан интерпретациялауға және диагностикалық кері байланыс қалыптастыруға пайдаланылады.

Интеллектуалдық модель негізінде программалаудан білімді бақылауды практикалық ұйымдастыру барысында орындалатын негізгі кезеңдер, әр кезеңде пайдаланылатын бақылау деректері мен талдау құралдары және олардың нәтижелері 11-кестеде жүйеленіп берілді.

Кесте 11 – Интеллектуалдық модель негізінде программалаудан білімді бақылауды ұйымдастырудың жұмыс кезеңдері

Кезең	Процесс	Бақылау деректері мен құралдары	Нәтиже
1	Бақылау тапсырмасын анықтау	Оқу мақсаты, бақыланатын білім мен әрекеттер, бағалау көрсеткіштері, тапсырманың күрделілік деңгейі	Бақылау мақсатына сәйкес тапсырма
2	Студенттің тапсырманы орындауы	Программалық код, алгоритмдік түсіндірме, тестілік жағдайлар, қажет болған жағдайда GenAI-мен өзара әрекеттесу деректері	Студент әрекетін сипаттайтын бастапқы бақылау деректері
3	Бақылау деректерін жинақтау	Код нұсқалары, тестілеу нәтижелері, қателер, енгізілген түзетулер, түсіндірмелер	Талдауға арналған бақылау деректерінің жиынтығы
4	Бақылау деректерін интеллектуалдық талдау	Автоматтандырылған тестілеу, статикалық және динамикалық талдау, үдерістік талдау, LLM-негізіндегі семантикалық талдау	Бағалау көрсеткіштерін сипаттайтын ақпараттық белгілер
5	Ақпараттық белгілерді бағалау көрсеткіштерімен сәйкестендіру	Талдау нәтижелері, бағалау көрсеткіштері және олардың деңгейлік сипаттамалары	Көрсеткіштер бойынша студент нәтижелерінің сипаттамасы
6	Білім жағдайын педагогикалық интерпретациялау	Көрсеткіштер бойынша алынған нәтижелер, қажет болған жағдайда оқытушының тексеруі	Студенттің программалаудан білім жағдайының сипаттамасы
7	Диагностикалық кері байланыс және келесі тапсырманы анықтау	Анықталған жетістіктер мен қиындықтар	Диагностикалық кері байланыс және келесі оқу әрекеті

Интеллектуалдық модель негізінде бақылауды осылай ұйымдастыру барысында 2.1-бөлімде негізделген бағалау критерийі мен көрсеткіштерінің жүйесі сақталады, ал GenAI-мен өзара әрекеттесу деректері студент білімінің жеке бағалау көрсеткіші ретінде емес, программалық шешімнің қалыптасу үдерісін сипаттайтын қосымша дерек көзі ретінде пайдаланылады.

Бақылау нәтижелерін педагогикалық интерпретациялау және кері байланыс әдісі 2.1-бөлімде негізделген нәтижелік-интерпретациялық компонентті практикалық деңгейде іске асыруға бағытталады. Әдістің мақсаты

– бақылау нәтижесін автоматты түрде бір қорытынды балға айналдыру емес, интеллектуалдық талдау нәтижесінде алынған ақпараттық белгілерді тиісті бағалау көрсеткіштерімен сәйкестендіріп, олардың байқалу деңгейін анықтау және осы негізде студенттің программалаудан білім жағдайын педагогикалық тұрғыдан интерпретациялау.

Сондықтан бақылау нәтижесі бір ғана қорытынды балмен шектелмейді. Әрбір бағалау көрсеткіші бойынша студенттің қандай нәтижеге жеткені, қандай қиындықтары бар екені және қандай білім мен әрекеттерді одан әрі жетілдіру қажеттігі анықталады. Осы нәтижелер студентке берілетін диагностикалық кері байланыстың мазмұнын анықтауға негіз болады.

Мысалы, екі студент бір программалау тапсырмасы бойынша бірдей қорытынды балл алуы мүмкін. Бірінші студенттің программасы барлық тестілік жағдайлардан өткенімен, ол алгоритмді түсіндіруде және қабылданған шешімді негіздеуде қиындыққа кездесуі мүмкін. Екінші студент алгоритмді дұрыс түсіндіріп, шешімін негіздей алғанымен, программасында бір шеттік жағдайды ескермеуі мүмкін. Бір ғана қорытынды балл бұл айырмашылықтарды толық көрсете алмайды. Ал бағалау көрсеткіштері бойынша алынған нәтижелер бірінші студентке шешімді түсіндіру мен негіздеуді, екінші студентке функционалдық дұрыстықты жетілдіруге бағытталған кері байланыс беруге мүмкіндік береді.

Кері байланыс үш деңгейде ұйымдастырылады. Нәтижелік кері байланыс студентке программаның қандай тестілік жағдайлардан өткенін немесе өтпегенін көрсетеді. Бағалау көрсеткіштері бойынша кері байланыс студенттің қай көрсеткіштер бойынша жеткілікті нәтижеге жеткенін және қай көрсеткіштер бойынша қиындықтары бар екенін анықтауға мүмкіндік береді. Диагностикалық кері байланыс анықталған қиындықтың қандай білім мен программалау әрекеттерімен байланысты екенін түсіндіреді және келесі оқу әрекетінде неге назар аудару қажеттігін көрсетеді [133].

Автоматтандырылған және оқытушы тарапынан берілетін кері байланыс өзара толықтырушы қызмет атқарады. Автоматтандырылған кері байланыс программаның функционалдық дұрыстығын, тестілеу нәтижелерін және программалық кодтың кейбір құрылымдық сипаттарын жедел көрсетуге мүмкіндік береді. Ал алгоритмдік шешімнің негізділігін, анықталған қателердің себептерін, балама шешу тәсілдерін және студенттің өз шешімін түсіндіруін педагогикалық тұрғыдан интерпретациялау қажет болған жағдайда оқытушының бағалауымен толықтырылады. Сондықтан интеллектуалдық модельде автоматтандырылған кері байланыс оқытушының педагогикалық шешімін алмастырмайды, керісінше бақылау барысында алынған деректер мен ақпараттық белгілерді жүйелеп, педагогикалық интерпретацияға қосымша ақпарат береді.

Мұндай ұйымдастыруда кері байланыс орындалған тапсырманың нәтижесін хабарлаумен ғана шектелмей, студенттің келесі оқу әрекетін анықтауға негіз болады. Студенттің анықталған білім жағдайына сәйкес келесі тапсырманың күрделілік деңгейін, тапсырманың мазмұнын немесе басым назар аударылатын бағалау көрсеткішін таңдауға болады. Мысалы, функционалдық дұрыстығы жеткілікті болғанымен, алгоритмдік шешімді негіздеуде қиындық байқалған студентке бірнеше шешу тәсілін салыстыруды және таңдалған

алгоритмнің тиімділігін негіздеуді талап ететін тапсырма ұсынылады. Осылайша нәтижелік-интерпретациялық компоненттен алынған ақпарат деңгейлік-тапсырма компонентінде келесі тапсырманы таңдауға пайдаланылады және интеллектуалдық бақылаудың циклдік сипаты қамтамасыз етіледі.

Әр бағалау критерийі бойынша студенттің тапсырманы орындау нәтижелері мен әрекеттері бағаланып, жетістік деңгейі анықталады. Анықталған қиындықтар негізінде студентке нені жетілдіру керектігі туралы кері байланыс беріледі. Мысалы, шартты студенттің жұмысында функционалдық дұрыстық жоғары деңгейде бағаланды: программа негізгі және шеттік тестілік жағдайларда дұрыс нәтиже береді, ал тексеру барысында анықталған сәтсіз жағдайларды студент өзі түсіндіріп, түзеткен. Қалған критерийлер бойынша орта деңгей байқалды: студент есептің мақсаты мен кіріс және шығыс деректерін дұрыс анықтап, есепке сәйкес алгоритм құрғанымен, шектеулердің шешімге әсерін, таңдаған тәсілінің тиімділігін және түзету тәсілін толық негіздемейді. Мұндай нәтиже студентке келесі тапсырмада шектеулерді талдауға және шешімін негіздеуге назар аудару қажеттігін көрсетеді.

Мысалы, жүйе: «Программа негізгі тестілік жағдайлардан өтті, бірақ бос массив және қайталанатын мәндер жағдайлары толық ескерілмеген. Функционалдық дұрыстық көрсеткіші бойынша шеттік жағдайларды өңдеуде қиындық байқалады. Келесі тапсырмада программаны іске асырмас бұрын шеттік жағдайларға арналған тестілік жағдайлар жиынын құруға назар аудару қажет» деген диагностикалық кері байланыс қалыптастыра алады.

Егер программа функционалдық тұрғыдан дұрыс жұмыс істегенімен, қолданылған алгоритм қабаттасқан циклдерге негізделіп, үлкен көлемдегі деректер үшін орындалу тиімділігі төмендесе, кері байланыс: «Программа дұрыс нәтиже береді, алайда деректер көлемі артқан жағдайда орындалу уақыты ұлғаяды. Тиімділік көрсеткіші бойынша хэш-кестеге негізделген балама тәсілді қарастырып, екі алгоритмнің есептеу күрделілігін салыстыру қажет» деген мазмұнда берілуі мүмкін.

LLM мұндай диагностикалық кері байланысты табиғи тілде қалыптастыруға көмектесе алады. Алайда оның мазмұны LLM-нің еркін қорытындысына емес, бақылау деректерін талдау нәтижесінде анықталған ақпараттық белгілер мен олардың бағалау көрсеткіштерімен сәйкестігіне негізделуі тиіс. Қажет болған жағдайда оқытушы программалық кодты, тестілеу нәтижелерін, кодқа енгізілген өзгерістерді және семантикалық талдау нәтижелерін қарап, қалыптастырылған кері байланысты нақтылай немесе толықтыра алады. Осылайша LLM-негізіндегі талдау мен кері байланыс оқытушының педагогикалық шешімін алмастырмай, оны қосымша ақпаратпен қамтамасыз етеді.

Жоғарыда сипатталған бақылау әдістерін практикалық тұрғыдан жүзеге асыруда бағалау рубрикасы қолданылады. Рубрикада 2.1-бөлімде негізделген бағалау көрсеткіштері олардың деңгейлік сипаттамаларымен сәйкестендіріледі. Нақты тапсырманың мақсаты мен мазмұнына қарай бағалау көрсеткіштерінің барлығы немесе олардың белгілі бір бөлігі таңдалады. Сондықтан әрбір программалау тапсырмасында барлық көрсеткіштерді бірдей қолдану міндетті емес. Таңдалған көрсеткіштер студенттің тапсырманы орындау нәтижелері мен

әрекеттерін бағалауға және оның программалаудан білім жағдайын интерпретациялауға негіз болады.

Программалау есептері мен жобаларын бағалауда 1.3-бөлімде айқындалған алты бағалау критерийіне негізделген рубрика қолданылады. Рубрикада әр критерийдің көрсеткіштері студенттің тапсырманы орындау нәтижелері мен әрекеттерін сипаттайтын деңгейлік дескрипторлар арқылы нақтыланады. Осы бөлімде рубрика бойынша жетістіктерді сипаттау үшін төмен, орта және жоғары деңгейлер қабылданды. Нақты тапсырманың мақсаты мен мазмұнына қарай бағалау критерийлерінің барлығы немесе белгілі бір бөлігі таңдалады. Олардың бағалаудағы үлесі мен дескрипторлары тапсырманың талаптарына сәйкес алдын ала белгіленіп, студенттерге ұсынылады.

Программалаудан білімді бақылаудың бағалау критерийлері бойынша деңгейлік дескрипторлар 12-кестеде берілген. Онда 5-кестеде жалпы түрде берілген көрсеткіштердің төмен, орта және жоғары деңгейлердегі сипаттамасы келтірілген.

Кесте 12 – Программалаудан білімді бақылаудың бағалау критерийлері бойынша деңгейлік дескрипторлар

Бағалау критерийі	Төмен деңгей	Орта деңгей	Жоғары деңгей
Тапсырма мазмұнын түсіну	Есептің мақсатын, кіріс және шығыс деректерін анықтауда қателеседі; шектеулерін ескермейді	Есептің мақсатын, кіріс және шығыс деректерін дұрыс анықтайды, бірақ шектеулерін немесе олардың шешімге әсерін толық түсіндірмейді	Есептің мақсатын, кіріс және шығыс деректерін, шектеулерін дұрыс анықтайды; олардың шешімге әсерін түсіндіреді
Алгоритмдік шешім	Алгоритм қадамдарының қисынды ретін сақтамайды; таңдаған құрылымдары есепті шешуге сәйкес келмейді	Есепті шешуге сәйкес алгоритм құрады, бірақ құрылымдарды таңдауын немесе тәсілінің тиімділігін толық негіздемейді	Алгоритм қадамдарын қисынды ұйымдастырады; құрылымдарды есепке сай таңдайды; тәсілінің тиімділігін негіздейді
Функционалдық дұрыстық	Программа негізгі тестілік жағдайларда есеп талабына сай нәтиже бермейді; студент қате шыққан жағдайды түсіндіруде қиналады	Программа негізгі тестілік жағдайларда дұрыс нәтиже береді, бірақ шеттік жағдайлардың бір бөлігін өңдемейді; студент қате шыққан жағдайды ішінара түсіндіреді	Программа негізгі және шеттік жағдайларда есеп талабына сай нәтиже береді; студент тексеру барысында анықталған сәтсіз жағдайларды түсіндіреді
Код сапасы	Код құрылымы бірізді емес; атаулар түсініксіз немесе конструкциялар орынсыз қолданылған; студент құрылымдық таңдауын негіздей	Кодтың негізгі құрылымы бірізді, атаулардың басым бөлігі түсінікті; студент құрылымдық таңдауын ішінара	Код құрылымы бірізді; атаулар түсінікті; конструкциялар орынды қолданылған; студент құрылымдық таңдауын негіздейді

12-кестенің жалғасы

Бағалау критерийі	Төмен деңгей	Орта деңгей	Жоғары деңгей
	алмайды	негіздейді, бірақ жекелеген конструкцияларды қолдануда кемшіліктер бар	
Қатені анықтау және түзету	Қатені және оның себебін өздігінен анықтай алмайды; енгізген өзгерісі қатені жоймайды немесе нәтижені қайта тексермейді	Қатені анықтап, түзетеді, бірақ оның себебін немесе түзету тәсілін толық негіздемейді; нәтижені қайта тексеруі жеткіліксіз	Қатені және оның себебін анықтайды; түзетуді негіздейді; өзгерістен кейін нәтижені қайта тексереді
Шешімді түсіндіру және негіздеу	Алгоритм қадамдары мен код бөліктерінің қызметін түсіндіруде қателеседі; таңдаған шешімінің орындылығын негіздей алмайды	Алгоритмнің негізгі қадамдары мен код бөліктерінің қызметін түсіндіреді, бірақ олардың байланысын немесе шешімінің орындылығын толық негіздемейді	Алгоритм қадамдарын және код бөліктерінің қызметін өзара байланыста түсіндіреді; таңдаған шешімінің орындылығын негіздейді

Рубриканы қолдануда программалық код, тестілеу нәтижелері, код өзгерістерінің тарихы және студенттің түсіндірмелері өзара байланыста қарастырылады. Алгоритмнің тиімділігі алгоритмдік шешім критерийі аясында, ал шеттік жағдайларды өңдеу функционалдық дұрыстық критерийі аясында бағаланады. Критерийлердің педагогикалық мазмұны сақталып, дескрипторлар тапсырманың нақты талаптарына сәйкес нақтыланады. Бағалау нәтижелерін педагогикалық тұрғыдан түсіндіру оқытушының қатысуымен жүзеге асырылады.

Бұл бөлімде рубриканы қолданудың әдістемелік тәртібі ұсынылады. Критерийлердің сандық салмақтары мен деңгейлердің сандық шекаралары мұнда бекітілмейді. Олардың негізділігін, рубриканың сенімділігін және интеллектуалдық бағалау нәтижелерінің оқытушы бағасымен сәйкестігін 2.3-бөлімдегі эксперименттік жұмыс барысында тексеру көзделеді.

Рубрика әртүрлі бақылау әдістері арқылы алынған нәтижелерді ортақ критерийлер бойынша бағалауға мүмкіндік береді. Осы негізде бақылау әдістері сабақтың мақсаты мен тапсырманың мазмұнына қарай өзара ұштастырылады. Мысалы, практикалық сабақта қысқа диагностикалық тапсырмалар арқылы жекелеген білім компоненттері тексеріліп, кейін әртүрлі деңгейдегі программалау есептері ұсынылады. Зертханалық жұмыста тестілеу мен қатені түзету әрекеттері, ал жобалық жұмыста кезеңдік нәтижелер мен тапсырманы орындау барысы бақыланады. GenAI қолданылған жағдайда студенттің ұсынылған шешімді тексеруі, түзетуі және негіздеуі де тиісті критерийлер бойынша бағаланады.

Оқу жұмысының әртүрлі формалары мен тапсырманы орындау жағдайларында интеллектуалдық бақылау әдістерін қолдану мүмкіндіктері 13-кестеде берілген.

Кесте 13 – Программалауды оқытудың әртүрлі формаларында интеллектуалдық бақылау әдістерін қолдану

Оқу формасы	Негізгі әдіс	Деректер	Диагностикалық фокус
Дәріс/семинардан кейінгі жедел тексеру	Операциялық-диагностикалық	Қысқа жауап, код фрагменті, түсіндірме	Түсіну, жеке білім компонентіндегі қиындық
Практикалық сабақ	Көпдеңгейлі тапсырмалар + программалау есебін бақылау	Код, тест, алгоритм түсіндірмесі	Алгоритм, дұрыстық, код сапасы
Зертханалық жұмыс	Тестілеу және қатені түзету + үдерістік бақылау	Тесттер, қателер журналы, код нұсқалары	Қатені анықтау және түзету, дұрыстық
Жобалық жұмыс	Бес кезеңдік жобалық бақылау	Схема, код, commits, tests, қорғау	Көпкомпонентті кезеңдік профиль
GenAI қолданылатын тапсырма	AI нәтижесін верификациялау	GenAI ұсынған шешім, студент енгізген өзгерістер, тест нәтижелері, шешімді негіздеу	Түсіну, сын тұрғысынан тексеру, түзету, негіздеу
Аралық бақылау	Гибридті критерийлік бақылау	Бірнеше тапсырма нәтижелері	Білім компоненттері бойынша жинақталған профиль

Интеллектуалдық модель негізінде ұсынылған бақылау әдістерін цифрлық ортада іске асыру мақсатында «Bilim AI» веб-платформасы әзірленді [56]. Платформа программалаудан студенттердің білімін бақылау барысында тапсырмаларды орындау нәтижелері мен үдерісі туралы деректерді жинақтауға және талдауға, белгіленген критерийлер бойынша бақылау нәтижелерін қалыптастыруға және оларды оқытушыға ұсынуға арналған. Ол студенттің бастапқы білім деңгейін диагностикалауға, деңгейіне сәйкес программалау тапсырмаларын қалыптастыруға, программалық кодты тексеруге, қателерді анықтап түсіндіруге және жекелендірілген кері байланыс беруге мүмкіндік береді. Генеративті ЖИ студентке программалау ұғымдарын қадамдық түсіндіру және қате туындаған жағдайда көмек көрсету үшін ЖИ-ассистент ретінде пайдаланылады. Студенттің тапсырмаларды орындау нәтижелері мен оқу прогресі дерекқорда сақталады.

Платформаның интеллектуалдық-талдау компонентінде пайдаланылатын үлкен тілдік модельді таңдау үшін программалауды оқытуда және интеллектуалдық білім беру жүйелерінде қолдануға болатын заманауи

генеративті ЖИ модельдеріне салыстырмалы талдау жүргізілді. Салыстыру барысында Llama 3, GPT-4, GPT-3.5, Gemini Pro және Claude 3 модельдерінің функционалдық мүмкіндіктері, программалық кодты генерациялау сапасы, логикалық пайымдау қабілеті, мультимодалдылықты қолдауы және білім беру жүйелерінде қолданылу тиімділігі қарастырылды.

Модельдерді бағалау халықаралық benchmark-жүйелердің нәтижелері негізінде жүргізілді. Benchmark-жүйе – жасанды интеллект модельдерінің сапасын, дәлдігін және функционалдық мүмкіндіктерін бағалауға арналған стандартталған тесттер жиынтығы. Негізгі көрсеткіштер ретінде модельдің программалық кодты дұрыс генерациялау қабілетін бағалайтын HumanEval және білім деңгейі мен логикалық пайымдау мүмкіндігін бағалайтын MMLU benchmark-жүйелерінің нәтижелері алынды. Сонымен қатар модельдердің кодты түсіндіру және қателерді анықтау сапасы, күрделі есептерді логикалық шешу қабілеті (reasoning), жауап беру жылдамдығы, мультимодалдылығы және локалды іске қосу мүмкіндігі ескерілді. Салыстырмалы талдау нәтижелері 14-кестеде келтірілген.

Кесте 14 – Программалауды оқытуда қолдануға арналған генеративті ЖИ модельдерінің салыстырмалы сипаттамасы

Критерий	Llama 3	GPT-4	GPT-3.5	Gemini Pro	Claude 3
Әзірлеуші	Meta	OpenAI	OpenAI	Google	Anthropic
Модель түрі	Open-source LLM	Коммерциялық LLM	Коммерциялық LLM	Мультимодальды LLM	Мультимодальды LLM
Кодты генерациялау және түсіндіру сапасы	Жоғары	Өте жоғары	Орташа	Жоғары	Өте жоғары
Жұмыс жылдамдығы	Жоғары	Орташа	Жоғары	Жоғары	Орташа
Локалды іске қосу мүмкіндігі	Бар	Жоқ	Жоқ	Жоқ	Жоқ
Мультимодалдылықты қолдау	Шектелген	Бар	Жартылай	Бар	Бар
HumanEval benchmark-жүйесі	81,7%	87,1%	~48–67%	71,9%	84,9%
MMLU benchmark-жүйесі	82%	86,5%	70%	85,9%	86,8%
Негізгі артықшылықтары	Ашық архитектурасы және локалды орналастыру мүмкіндігі	Логикалық пайымдау мен программалық код сапасының жоғары болуы	Жоғары жылдамдық және қолжетімділік	Кеңейтілген мультимодалдылық мүмкіндіктері	Терең талдау және толық түсіндіру мүмкіндігі

14-кестенің жалғасы

Критерий	Llama 3	GPT-4	GPT-3.5	Gemini Pro	Claude 3
Негізгі кемшіліктері	Күрделі логикалық пайымдау міндеттерінде GPT-4 моделінен төмен	Жабық архитектура және жоғары құны	Жауаптардың дәлдік деңгейі төменірек	Кей жағдайларда жауаптардың тұрақсыздығы байқалады	Жоғары есептеу ресурстарын талап етеді

14-кестеден GPT-4 және Claude 3 модельдерінің программалық кодты генерациялау мен түсіндіру сапасы бойынша жоғары көрсеткіштерге ие екені көрінеді. Алайда GPT-4 моделінің жабық архитектурасы мен жоғары құны, Claude 3 моделінің жоғары есептеу ресурстарын талап етуі, сондай-ақ бұл модельдерді локалды іске қосу мүмкіндігінің болмауы оларды білім беру платформасына енгізуді шектейді. Llama 3 моделі жоғары жұмыс жылдамдығымен, ашық архитектурасымен және локалды орналастыру мүмкіндігімен ерекшеленеді, ал HumanEval және MMLU көрсеткіштері бойынша коммерциялық модельдерге жақын нәтиже көрсетеді. Осы ерекшеліктер ескеріліп, «Bilim AI» платформасында генеративті талдау Llama отбасының Llama 3.3 70B моделі арқылы жүзеге асырылды [134].

Платформаны әзірлеуде келесі архитектуралық қағидалар басшылыққа алынды. Жүйенің негізгі программалау тілі ретінде AI/ML модельдерімен интеграциялауға қолайлы Python тілі таңдалды. Модульдік архитектура жүйені жеке функционалдық модульдерге бөлуге мүмкіндік береді, ал «интерфейс → қолданбалы логика → деректер қабаты» түріндегі қабаттық архитектура олардың атқаратын қызметін ажыратады. Клиент, сервер және сыртқы GenAI сервистері арасындағы деректер алмасу REST API арқылы қамтамасыз етіледі. Асинхронды жұмыс режимі LLM-ге сұраныс жіберілген кезде сервердің басқа сұраныстарды қатар өңдеуіне мүмкіндік береді. Мұндай архитектура жүйені кеңейтуге, тестілеуге және масштабтауға мүмкіндік береді. Платформаны әзірлеуде қолданылған технологиялар 15-кестеде берілген.

Кесте 15 – «Bilim AI» веб-платформасын әзірлеуде қолданылған технологиялар

Технология	Нұсқасы	Мақсаты
Python	3.12	Негізгі программалау тілі
FastAPI	0.115	Веб-фреймворк (бэкенд)
SQLAlchemy	2.0	ORM кітапханасы
SQLite	3	Реляциялық дерекқор
Pydantic	2.x	Деректерді валидациялау
Jinja2	3.x	HTML шаблондау жүйесі
Uvicorn	0.30	ASGI веб-сервер
Groq API	–	Llama 3.3 70B моделіне қол жеткізу
python-jose	3.x	JWT аутентификация
passlib	1.7	Құпиясөздерді хэштеу
aiosmtplib	3.x	Email жіберу
HTML5/CSS3/JS	–	Фронтенд интерфейсі

«Bilim AI» веб-платформасы https://bilim-al-nxoe.onrender.com серверінде орналастырылған. Платформа 2024–2025 және 2025–2026 оқу жылдарында Абай атындағы Қазақ ұлттық педагогикалық университетінде 6В01507 – Информатика және 6В01509 – Информатика және робототехника білім беру бағдарламалары бойынша «Программалауға кіріспе», «Алгоритмдер және бағдарлама жасау негіздері», «Алгоритмдер және мәліметтер құрылымы», «Программалау II (Python)» пәндерінің оқу үдерісіне енгізілді.

2.2-бөлімде интеллектуалдық модельді программалаудан білімді бақылауда қолдану әдістері нақтыланды. Бақылау оқу мақсатына сәйкес тапсырмаларды іріктеу, студенттің орындау нәтижелері мен әрекеттері туралы деректерді жинау және оларды бағалау критерийлері мен көрсеткіштеріне сәйкестендіріп түсіндіру арқылы жүзеге асырылады. Әдістер нақты платформаға тәуелді емес, техникалық құралдар тапсырма мен оқу ортасына қарай таңдалады.

Программалық код, тестілеу нәтижелері, код өзгерістерінің тарихы және студенттің түсіндірмелері өзара байланыста бағаланады. GenAI қолданылғанда студенттің ұсынылған шешімді түсінуі, тексеруі, түзетуі және негіздеуі ескеріледі. Бағалау нәтижелерін педагогикалық тұрғыдан түсіндіру мен қорытынды шешім қабылдау оқытушының қатысуымен жүзеге асырылады.

Диагностикалық кері байланыс анықталған қиындықтарға сәйкес келесі тапсырманы таңдауға және оқытуды түзетуге негіз болады. Ұсынылған әдістердің нәтижелілігін, рубриканың сенімділігін және интеллектуалдық бағалаудың оқытушы бағасымен сәйкестігін 2.3-бөлімде эксперименттік тексеру көзделеді.

2.3 Программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін эксперименттік тексеру және оның тиімділігін бағалау

Алдыңғы бөлімдерде программалаудан студенттердің білімін интеллектуалдық бақылаудың теориялық негіздері айқындалып, оның қағидалары мен бағалау көрсеткіштері негізделді, программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі әзірленді және осы модель негізінде бақылауды ұйымдастыру әдістері ұсынылды. Ұсынылған модельдің педагогикалық тұрғыдан негізділігін тәжірибеде анықтау оның компоненттері мен олардың арасындағы байланыстардың программалаудан білімді бақылау үдерісінде іске асырылуын эксперименттік тексеруді және алынған нәтижелер негізінде модельдің тиімділігін бағалауды қажет етеді. Осыған байланысты зерттеудің келесі кезеңінде педагогикалық эксперимент ұйымдастырылды.

Педагогикалық эксперименттің мақсаты – программалаудан студенттердің білімін бақылаудың әзірленген интеллектуалдық моделін эксперименттік тексеру және оның тиімділігін бағалау.

Қойылған мақсатқа сәйкес педагогикалық эксперименттің міндеттері:

– программалаудан студенттердің білімін бақылаудың бастапқы жағдайын зерттеу және қолданылатын бағалау тәсілдерінің диагностикалық мүмкіндіктері мен шектеулерін анықтау;

– әзірленген интеллектуалдық модельдің компоненттерін программалаудан студенттердің білімін бақылау үдерісінде кешенді іске асыру;

– интеллектуалдық модель негізінде программалаудан студенттердің білімін бақылауды ұйымдастыру және эксперимент барысында алынған нәтижелерді талдау;

– интеллектуалдық модель негізінде қалыптастырылған бағалау нәтижелерін оқытушының референттік бағаларымен салыстыру;

– эксперимент нәтижелерін салыстырмалы және математикалық-статистикалық талдау негізінде интеллектуалдық модельдің тиімділігін бағалау.

Зерттеудің ғылыми болжамы: егер программалаудан студенттердің білімін бақылау педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысы негізінде жүзеге асырылса, онда студенттің программалаудан білім жағдайын неғұрлым толық анықтауға мүмкіндік туады, өйткені интеллектуалдық бақылау студенттің программалау тапсырмасын орындаудың соңғы нәтижесімен ғана шектелмей, оның программалау әрекетін сипаттайтын деректерді бағалау көрсеткіштері негізінде кешенді талдауға және педагогикалық тұрғыдан интерпретациялауға мүмкіндік береді.

Зерттеу болжамын тексеру және қойылған міндеттерді жүзеге асыру мақсатында педагогикалық эксперимент үш кезеңде жүргізілді: диагностикалық, сараптамалық-негіздеуші және эксперименттік-тексеру. Диагностикалық кезеңде программалаудан студенттердің білімін бақылаудың бастапқы жағдайы мен қолданыстағы бағалау тәсілдерінің мүмкіндіктері және шектеулері анықталды. Сараптамалық-негіздеуші кезеңде интеллектуалдық модельдің бағалау мазмұны, критерийлері мен көрсеткіштері сараптамалық

тұрғыдан негізделді. Эксперименттік-тексеру кезеңінде әзірленген интеллектуалдық модель нақты студенттік программалау жұмыстары негізінде іске асырылып, алынған нәтижелер оқытушының сараптамалық бағалау нәтижелерімен салыстырылды және модельдің тиімділігі математикалық-статистикалық талдау негізінде бағаланды.

Эксперименттің әр кезеңінде зерттеу мақсаты мен міндеттеріне сәйкес дербес іріктемелер қамтылды.

Диагностикалық кезеңде студенттер мен профессор-оқытушылар құрамынан барлығы 360 респондент қатысты. Кезеңнің мақсаты программалаудан студенттердің білімін бақылаудың бастапқы жағдайын және қолданыстағы бағалау тәсілдерінің мүмкіндіктері мен шектеулерін анықтау болды. Алынған нәтижелер GenAI технологияларын пайдалану жағдайында программалаудан білімді бақылауды жетілдіру қажеттілігін негіздеуге пайдаланылды.

Сараптамалық-негіздеуші кезеңде бағалау мазмұны мен критерийлерін сараптамалық тұрғыдан негіздеуге барлығы 162 респондент қатысты, олардың ішінде IT бағытындағы білім алушылар – 109, жоғары оқу орындарының профессор-оқытушылар құрамы – 31, IT саласының мамандары – 22. Бұл бағалау мазмұны мен бағалау критерийлерінің маңыздылығын білім алушы, оқытушы және кәсіби сала маманы тұрғысынан саралауға мүмкіндік берді.

Эксперименттік-тексеру кезеңінде Қазақстанның үш жоғары оқу орны – Абай атындағы Қазақ ұлттық педагогикалық университеті, Қазақ ұлттық қыздар педагогикалық университеті және Торайғыров университетінің 1–4 курсында «Физика–Информатика», «Информатика» және «Ақпараттық жүйелер» білім беру бағдарламалары бойынша білім алатын 187 студент қатысты. Кезеңнің мақсаты әзірленген программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін нақты оқу үдерісінде эксперименттік тексеру және оның тиімділігін бағалау болды. Модель студенттердің программалау тапсырмаларын орындау барысында алынған деректер негізінде апробацияланып, алынған нәтижелер оның тиімділігін бағалауға пайдаланылды.

Эксперименттік жұмыстар AP23490592 «Генеративті жасанды интеллект технологиясының программалаудан ЖОО студенттерінің білімін бақылау әдістемесіне әсерін зерттеу» гранттық жобасы аясында жүргізілді. Зерттеу барысында алынған эмпирикалық материалдар программалаудан студенттердің білімін бақылауда GenAI технологияларын қолданудың мүмкіндіктерін айқындауға, бағалау мазмұны мен критерийлерін негіздеуге және әзірленген интеллектуалдық модельдің тиімділігін эксперименттік тексеруге пайдаланылды.

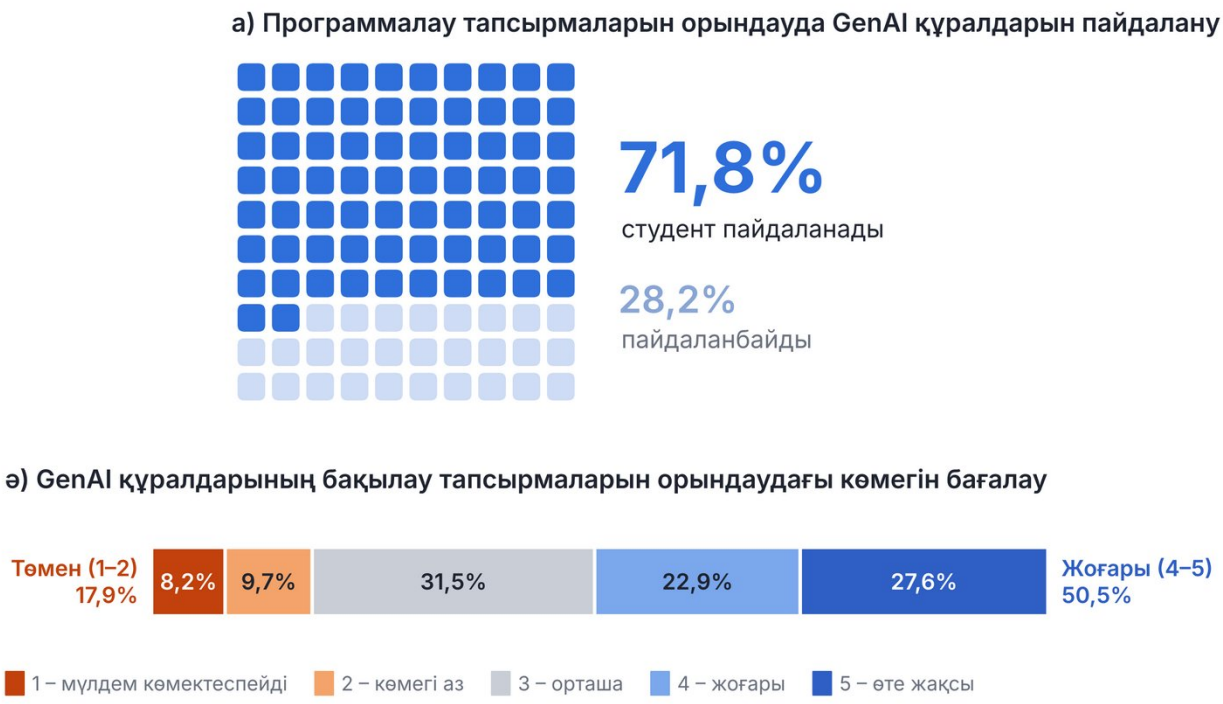
Диагностикалық кезеңде программалаудан студенттердің білімін бақылаудың бастапқы жағдайын және қолданыстағы бақылау тәсілдерінің мүмкіндіктері мен шектеулерін анықтау мақсатында студенттер мен профессор-оқытушылар құрамына құрылымдалған сауалнама жүргізілді. Диагностикаға барлығы 360 респондент, оның ішінде 340 студент және 20 оқытушы қатысты. Студенттерге арналған сауалнама программалаудан білімді бақылауда қолданылатын тапсырмаларды, оларға қанағаттану деңгейін, программалау тапсырмаларын орындауда GenAI құралдарын пайдалану тәжірибесін, олардың

программалау дағдыларына әсерін және бақылау тапсырмаларын орындаудағы көмегін бағалауды қамтыды (Қосымша А). Оқытушыларға арналған сауалнама программалаудан студенттердің білімін бақылауда қолданылатын әдістер мен оны ұйымдастыру тәсілдерін, сондай-ақ оқу материалдарын дайындауда GenAI құралдарын пайдалану тәжірибесін анықтауға бағытталды (Қосымша Ә). Сандық деректер сипаттамалық статистика әдістерімен өңделді; категориялық көрсеткіштер абсолюттік және пайыздық үлестер арқылы сипатталды. Оқытушылар сауалнамасындағы GenAI көмегін бағалау үшін медиана мен квартильаралық аралық та есептелді.

Студенттердің бастапқы диагностикасы программалаудан білімді бақылауда практикалық тапсырмалардың басым қолданылатынын көрсетті. Респонденттердің 83,5%-ы практикалық тапсырмаларды, 46,2%-ы жобалық жұмыстарды, 35,9%-ы тест тапсырмаларын, 31,5%-ы ауызша сауалнаманы, 27,1%-ы кодты тексеруді, ал 14,4%-ы интербелсенді тапсырмаларды көрсеткен. Бірнеше жауап нұсқасын таңдауға мүмкіндік берілгендіктен, көрсеткіштердің жиынтық үлесі 100%-дан асады. Бақылау тапсырмаларына қанағаттану деңгейін бес балдық шкала бойынша бағалау нәтижесінде орташа көрсеткіш 4,18 балды құрады. Студенттердің 77,4%-ы бақылау тапсырмаларын 4 немесе 5 балмен бағалаған. Бұл көрсеткіштер қолданыстағы бақылау тапсырмаларының студенттер тарапынан жалпы оң қабылданатынын сипаттайды.

Сонымен қатар студенттердің программалау тапсырмаларын орындауда GenAI құралдарын пайдалану тәжірибесі анықталды. Нәтижесінде респонденттердің 71,8%-ы GenAI құралдарын пайдаланатынын, ал 28,2%-ы пайдаланбайтынын көрсеткен. GenAI құралдарының бақылау тапсырмаларын орындаудағы көмегін бес балдық шкала бойынша бағалау кезінде студенттердің 8,2%-ы 1 балл, 9,7%-ы 2 балл, 31,5%-ы 3 балл, 22,9%-ы 4 балл, ал 27,6%-ы 5 балл берген. GenAI көмегін 4 немесе 5 балмен бағалаған студенттердің жиынтық үлесі 50,5%-ды, орташа көрсеткіш шамамен 3,52 балды құрады. Бұл нәтижелер студенттердің басым бөлігінің программалау тапсырмаларын орындауда GenAI құралдарын пайдалану тәжірибесі бар екенін, алайда олардың көмегіне берілген бағалардың әркелкі екенін көрсетті (9-сурет).

Студенттермен қатар программалаудан білімді бақылаудың бастапқы жағдайын оқытушылар тұрғысынан анықтау мақсатында 20 оқытушыға сауалнама жүргізілді. Оқытушылардың жауаптарын талдау программалаудан студенттердің білімін бақылауда жобалық және практикалық тапсырмалардың басым қолданылатынын көрсетті: олардың әрқайсысының үлесі 30,6%-ды құрады. Тест тапсырмаларының үлесі 24,5% болды; кодты тексеру мен ауызша сұрау сауалнама да қолданылғанымен, олардың үлесі төмен болды.



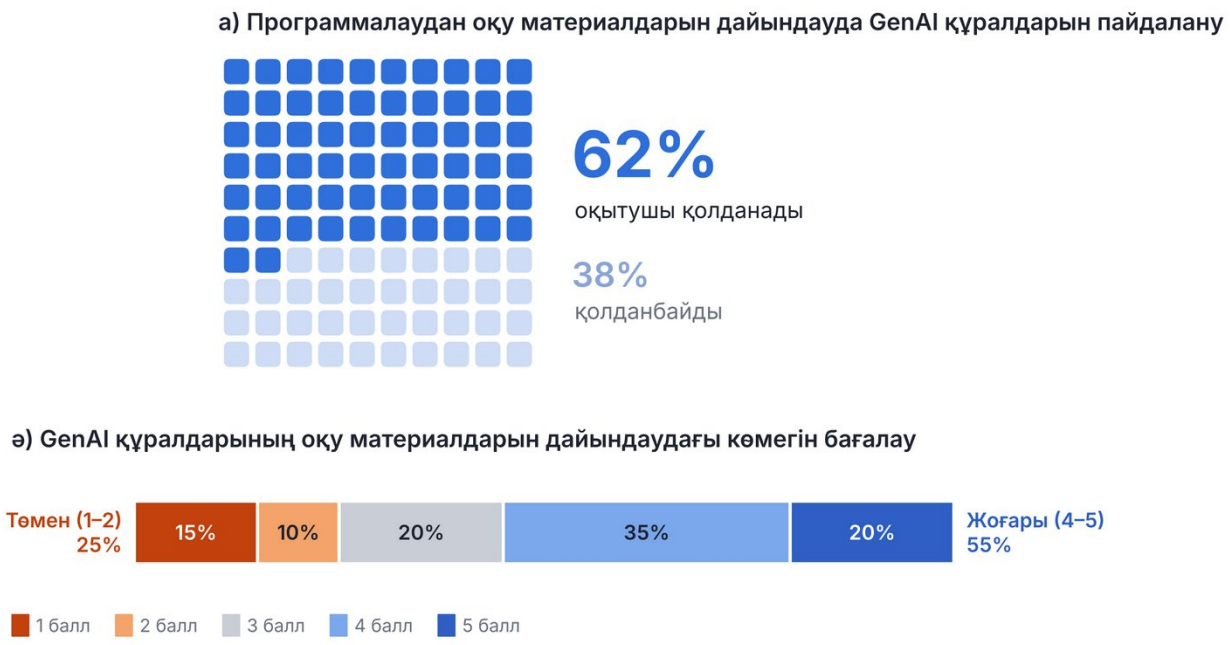
Сурет 9 – Студенттердің программалау тапсырмаларын орындауда GenAI құралдарын пайдалану тәжірибесі және олардың көмегін бағалауы

Білімді бақылауды ұйымдастыру тәсілдерін талдау барысында топтық талқылаулардың үлесі 31,8%, онлайн программалау платформаларының үлесі 25% болғаны анықталды. Сонымен қатар кодтау жарыстары, тәлімгерлік және жұптық программалау, өзін-өзі бағалау және жұмыстар портфолиосы сияқты тәсілдер де қолданылған. Бұл нәтижелер программалаудан білімді бақылауда дәстүрлі тапсырмалармен қатар студенттердің практикалық әрекеті мен өзара әрекеттесуіне негізделген әртүрлі ұйымдастыру тәсілдерінің қолданылатынын көрсетті (10-сурет).



Сурет 10 – Оқытушылардың программалаудан білімді бақылауда қолданатын әдістері және бақылауды ұйымдастыру тәсілдері

Оқытушылардың GenAI құралдарын пайдалану тәжірибесін талдау олардың 62%-ы программалаудан оқу материалдарын дайындауда аталған құралдарды қолданатынын, ал 38%-ы қолданбайтынын көрсетті. GenAI құралдарының программалаудан оқу материалдарын дайындаудағы көмегін бес балдық шкала бойынша бағалау кезінде оқытушылардың 15%-ы 1 балл, 10%-ы 2 балл, 20%-ы 3 балл, 35%-ы 4 балл және 20%-ы 5 балл берген. Бағалаудың медианалық мәні 4,00 балды, квартильаралық аралығы 2,00–4,00 балды құрады. Бұл нәтижелер оқытушылардың басым бөлігінің оқу материалдарын дайындауда GenAI құралдарын қолдану тәжірибесі бар екенін, алайда олардың көмегіне берілген бағалардың әркелкі екенін көрсетті (11-сурет).



Сурет 11 – Оқытушылардың программалаудан оқу материалдарын дайындауда GenAI құралдарын пайдалану тәжірибесі және олардың көмегін бағалауы

Диагностикалық кезеңнің нәтижелері студенттердің қолданыстағы бақылау тапсырмаларына қанағаттану деңгейінің жоғары екенін және оқытушылар тәжірибесінде практикалық және жобалық тапсырмалардың қолданылатынын көрсетті. Сонымен қатар студенттердің 71,8%-ының программалау тапсырмаларын орындауда, ал оқытушылардың 62%-ының оқу материалдарын дайындауда GenAI құралдарын пайдалануы бұл технологиялардың программалауды оқыту мен оқу әрекетіне енгенін айқындады.

Алайда GenAI құралдарын қолдану фактісі мен олардың көмегіне берілген субъективті баға студенттің тапсырманы қаншалықты дербес орындағанын, программалық шешімнің логикасын қаншалықты түсінетінін және алынған нәтижені өз бетінше негіздей алатынын тікелей анықтауға мүмкіндік бермейді. Осыған байланысты программалаудан білімді бақылауды тек тапсырманың соңғы нәтижесімен шектемей, студенттің тапсырманы орындау үдерісін, шешім құру логикасын, программалық кодты түсінуі мен түсіндіруін және GenAI құралдарын пайдалану ерекшеліктерін сипаттайтын деректермен толықтыру қажеттілігі айқындалды. Бұл қажеттілік зерттеудің келесі – сараптамалық-негіздеуші кезеңінде программалаудан білімді интеллектуалдық бақылаудың бағалау мазмұнын, критерийлері мен көрсеткіштерін сараптамалық тұрғыдан негіздеуге негіз болды.

Сараптамалық-негіздеуші кезеңнің мақсаты – программалаудан студенттердің білімін интеллектуалдық бақылауда қолданылатын бағалау мазмұны, критерийлері мен көрсеткіштерінің негізділігін әртүрлі қатысушылар топтарының бағалауы негізінде анықтау болды.

Диагностикалық кезең нәтижелері GenAI құралдарын пайдалану жағдайында программалаудан білімді бақылауды студенттің тапсырманы

орындауының соңғы нәтижесімен ғана шектемей, оның программалау әрекетін сипаттайтын деректермен толықтыру қажеттігін көрсетті. Осыған байланысты сараптамалық-негіздеуші кезеңде бақылау мазмұны мен бағалау критерийлерінің маңыздылығын сараптамалық тұрғыдан анықтауға назар аударылды.

Сараптамалық-негіздеуші кезеңге қатысушыларды іріктеу программалаудан білімді бақылаудың білім алушы, педагогикалық және кәсіби қырларын қамту қажеттілігіне негізделді. Осыған сәйкес IT бағытындағы білім алушылар, жоғары оқу орындарының профессор-оқытушылар құрамы және IT саласының мамандары қамтылды. Мұндай құрам программалаудан білімді бақылауды оқу үдерісі мен кәсіби талаптар тұрғысынан кешенді қарастыруға мүмкіндік берді.

Сараптамалық зерттеуде жобалық программалау ұйымдастырылды, өйткені жобалық тапсырмалар студенттің дайын программалық нәтижесімен қатар оны әзірлеу барысындағы әрекеттерін де қарастыруға жағдай жасайды. Осы контекст бақылау мазмұны мен бағалау критерийлерін программалық шешімді әзірлеу үдерісімен байланыстыра сараптауға негіз болды.

Сараптамалық-негіздеуші кезеңде қатысушыларға бақылау мазмұны мен бағалау критерийлерінің маңыздылығын анықтауға арналған сауалнама жүргізілді (Қосымша Б). Сауалнама нәтижелерін өңдеуде қатысушылардың даярлық деңгейі мен тәжірибесіндегі айырмашылықтарды ескеру үшін жеке салмақ коэффициенттері қолданылды. Оларды есептеуде профессор-оқытушылар құрамы үшін ғылыми дәрежесі, академиялық лауазымы мен жұмыс өтілі; IT бағытындағы білім алушылар үшін білім беру деңгейі, оқу курсы және олимпиадалар мен конкурстардағы жетістіктері; IT мамандары үшін кәсіби деңгейі мен жұмыс тәжірибесі ескерілді. Осы көрсеткіштер бойынша жинақталған балдар ең төмен және ең жоғары мүмкін мәндер аралығында нормаланып, әр қатысушының бағасын қорытынды нәтижелерді есептеуде ескеруге арналған жеке салмақ коэффициенті анықталды. Нәтижесінде орташа салмақ коэффициенті IT бағытындағы білім алушыларда – 0,387, профессор-оқытушылар құрамында – 0,472, IT мамандарында – 0,499, ал жалпы іріктеме бойынша – 0,419 болды. Бұл мәндер топтар арасындағы даярлық деңгейі мен тәжірибесіндегі айырмашылықтарды ескеруге мүмкіндік берді.

Сараптамалық бағалаудың мазмұндық негізі ретінде студенттердің программалау әрекетін кешенді қарастыруға мүмкіндік беретін екі жобалық программалау тапсырмасы әзірленді: «Қатысу деңгейін бақылау жүйесі (Computer Vision)» және «Студенттің уақытын ақылды жоспарлау» веб-қосымшасы (Қосымша В).

Программалау бойынша жобалық тапсырманы орындау бір ғана қорытынды программалық өнімді алумен шектелмей, өзара байланысты бірнеше әрекет кезеңдерін қамтиды. Осыған сәйкес жобалық тапсырманы орындау үдерісі бес кезеңге құрылымдалды: есептің мазмұнын түсіну және қойылымын анықтау (Analysis), шешімді жоспарлау (Planning), программалық шешімді іске асыру (Implementation), оның орындалуын бақылау (Control) және нәтижені қорғау (Defense). Әр кезеңде студент әрекетін сипаттайтын бағалау

критерийлері ұсынылып, қатысушылардан олардың маңыздылығын бағалау сұралды.

Сауалнама нәтижелерін талдау программалау әрекетінің бес кезеңі бойынша бағалау критерийлерінің маңыздылығына берілген бағаларда қатысушылар топтары арасында айырмашылықтар бар екенін көрсетті. Программалау әрекетінің кезеңдері бойынша қатысушылардың бағалау нәтижелері 16-кестеде берілген.

Кесте 16 – Программалау әрекетінің кезеңдері бойынша сараптамалық бағалау нәтижелері

Программалау әрекетінің кезеңдері	IT бағытындағы білім алушылар	Профессор-оқытушылар құрамы	IT мамандары	Жалпы
Есептің мазмұнын түсіну және қойылымын анықтау (Analysis)	0,70	0,51	0,73	0,66
Шешімді жоспарлау (Planning)	0,72	0,54	0,76	0,69
Программалық шешімді іске асыру (Implementation)	0,70	0,57	0,77	0,69
Орындалуын бақылау (Control)	0,72	0,57	0,73	0,69
Нәтижені қорғау (Defense)	0,74	0,59	0,76	0,71
Барлық кезеңдер бойынша	0,72	0,55	0,75	0,69

16-кесте нәтижелері бойынша есептің мазмұнын түсіну және қойылымын анықтау, шешімді жоспарлау, программалық шешімді іске асыру, оның орындалуын бақылау және нәтижені қорғау кезеңдері бойынша жалпы бағалау мәндері 0,66–0,71 аралығында болды. Ең жоғары мән нәтижені қорғау кезеңінде (0,71) анықталды. Қатысушылар топтары бойынша IT мамандары мен IT бағытындағы білім алушылардың бағалау мәндері профессор-оқытушылар құрамының бағаларымен салыстырғанда жоғары болды.

Кезеңдер бойынша алынған жалпы нәтижелермен қатар, әр кезеңде студенттің программалау әрекетін бағалауға ұсынылған критерийлердің маңыздылығы талданды. Қатысушылар топтары бойынша жоғары бағаланған критерийлер 17-кестеде берілген.

Кесте 17 – Қатысушылар топтары бойынша жоғары бағаланған критерийлер

Программалау әрекетінің кезеңі	Қатысушылар тобы	Жоғары бағаланған критерий	Бағалау мәні
Есептің мазмұнын түсіну және қойылымын анықтау	IT бағытындағы білім алушылар	Бастапқы талаптардың дұрыстығы	0,72
	ПОҚ	Есепті қою логикасы	0,56
	IT мамандары	Бастапқы талаптардың дұрыстығы	0,82
Шешімді жоспарлау	IT бағытындағы білім алушылар	Есептің дұрыс қойылуы	0,74
	ПОҚ	Жоспардың құрылымы мен ұйымдастырылуы	0,56
	IT мамандары	Есептің дұрыс қойылуы	0,83
Программалық шешімді іске асыру	IT бағытындағы білім алушылар	Кодтың құрылымы мен оқылымдылығы	0,73
	ПОҚ	Программаның дұрыстығы	0,61
	IT мамандары	Программаның дұрыстығы	0,61
Орындалуын бақылау	IT бағытындағы білім алушылар	Нәтиженің дұрыстығы	0,75
	ПОҚ	Нәтиженің дұрыстығы	0,59
	IT мамандары	Нәтиженің дұрыстығы	0,79
Нәтижені қорғау	IT бағытындағы білім алушылар	Дұрыстық және сенімділік	0,76
	ПОҚ	Дұрыстық және сенімділік	0,62
	IT мамандары	Дұрыстық және сенімділік	0,83

17-кесте нәтижелері бағалау критерийлерінің маңыздылығына қатысты қатысушылар топтарының көзқарастарында айырмашылықтармен қатар ортақ ұстанымдардың да болғанын көрсетті. Есептің мазмұнын түсіну және қойылымын анықтау кезеңінде IT бағытындағы білім алушылар мен IT мамандары бастапқы талаптардың дұрыстығын, ал профессор-оқытушылар құрамы есепті қою логикасын жоғары бағалады. Шешімді жоспарлау кезеңінде білім алушылар мен IT мамандары есептің дұрыс қойылуына, профессор-оқытушылар құрамы жоспардың құрылымы мен ұйымдастырылуына басымдық берді. Программалық шешімді іске асыру кезеңінде білім алушылар кодтың құрылымы мен оқылымдылығын, ал профессор-оқытушылар құрамы мен IT мамандары программаның дұрыстығын жоғары бағалады. Орындалуын бақылау және нәтижені қорғау кезеңдерінде үш топтың бағалаулары сәйкес

келіп, тиісінше нәтиженің дұрыстығы және программалық шешімнің дұрыстығы мен сенімділігі басым критерийлер ретінде анықталды.

Программалау әрекетін бағалаудың негізгі критерийлерімен қатар, сауалнамада GenAI құралдарын қолдану ерекшеліктеріне де жеке назар аударылды. Бұл ретте GenAI-ді қолдану фактісін ғана емес, оның программалау әрекетінің қай кезеңінде, қандай мақсатта және қалай пайдаланылғанын ескеру көзделді. Осыған сәйкес GenAI қолдану программалау әрекетінің бес кезеңінің әрқайсысында жеке критерий ретінде ұсынылып, қатысушылар оның маңыздылығын 0–100% шкаласы бойынша бағалады. Критерийлер GenAI-ді есепті нақтылау мен идеяларды қалыптастыруда, шешімді жоспарлауда, программалық шешімді әзірлеуде, тестілеу мен қателерді түзетуде саналы қолдануды, сондай-ақ нәтижені қорғау барысында GenAI көмегімен не және қандай мақсатта орындалғанын түсіндіруді қамтыды. GenAI құралдарын қолдану критерийінің бес кезең бойынша бағалау нәтижелері 18-кестеде берілген.

Кесте 18 – Программалау әрекетінің кезеңдері бойынша GenAI құралдарын қолдануды бағалау нәтижелері

Программалау әрекетінің кезеңі	GenAI қолданудың бағаланатын сипаты	Орташа мән
Есептің мазмұнын түсіну және қойылымын анықтау	Есепті нақтылау, идеяларды қалыптастыру және талаптарды айқындауда GenAI-ді саналы қолдану	0,64
Шешімді жоспарлау	Талдау, идеяларды қалыптастыру және шешім құрылымын жоспарлауда GenAI-ді қолдану	0,66
Программалық шешімді іске асыру	GenAI-ді саналы қолдану, ұсынылған шешімді түсіну және бейімдеу	0,66
Орындалуын бақылау	Тестілеу, қателерді талдау және түзетуде GenAI-ді саналы қолдану	0,65
Нәтижені қорғау	GenAI-дің қалай және қандай мақсатта қолданылғанын түсіндіру	0,68

18-кесте нәтижелері бойынша GenAI құралдарын қолдану критерийінің бес кезеңдегі орташа мәндері 0,64–0,68 аралығында болды. Ең жоғары мән нәтижені қорғау кезеңінде (0,68) анықталды. Бұл кезеңде студенттің GenAI құралдарын қолдану фактісінен гөрі, оларды қалай және не үшін пайдаланғанын түсіндіре алуы бағаланды. Шешімді жоспарлау және программалық шешімді іске асыру кезеңдерінде көрсеткіш 0,66-ны құрап, GenAI-ді талдау мен шешім құруда саналы пайдалану, сондай-ақ ұсынылған шешімді түсіну және бейімдеу ескерілді. Орындалуын бақылау кезеңінде (0,65) GenAI-ді тестілеу, қателерді талдау және түзету барысында саналы пайдалану қарастырылды.

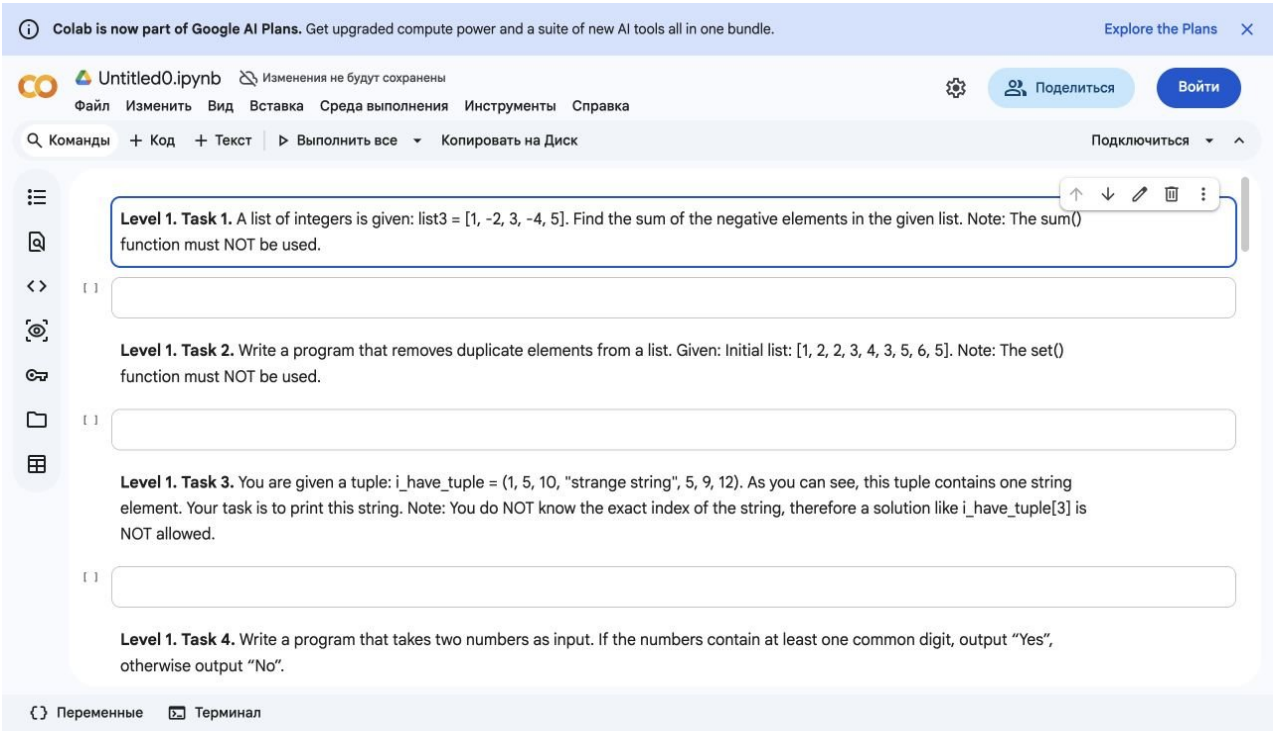
Осылайша, сараптамалық-негіздеуші кезең нәтижелері программалаудан білімді интеллектуалдық бақылауда студенттің дайын программалық нәтижесімен қатар, жобалық тапсырманы орындау кезеңдерінде көрінетін

шешімді әзірлеу әрекеттерін, қателерді анықтау мен түзету үдерісін, шешімді негіздеуін және GenAI құралдарын қолдану сипатын да ескеру қажеттігін көрсетті. Сараптамалық бағалау нәтижелерін теориялық зерттеу қорытындыларымен сабақтастыру есептің мазмұнын түсіну, алгоритмдік ойлау мен шешімнің тиімділігі, функционалдық дұрыстық пен шекаралық жағдайларды ескеру, код сапасы, шешімді түсіндіру, сондай-ақ қателерді анықтау, диагностикалау және түзету көрсеткіштерін программалау әрекетін бағалау тұрғысынан нақтылауға мүмкіндік берді. Сонымен қатар GenAI құралдарын пайдалану студенттің дербестігімен, ұсынылған шешімді түсінуі мен сыни талдауымен, оны бейімдеу және негіздеу қабілетімен байланысты қарастырылды. Нәтижесінде программалаудан білімді интеллектуалдық бақылаудың бағалау мазмұны мен критерийлері нақтыланып, оларды әзірленген модельдің критерийлік-бағалау компонентінде қолдануға және GenAI құралдарын пайдалану туралы деректерді интеллектуалдық бақылауда ескеруге негіз қаланды. Осы негізде келесі эксперименттік-тексеру кезеңінде нақтыланған бағалау мазмұны мен критерийлері енгізілген интеллектуалдық модель нақты студенттік программалау жұмыстары негізінде апробацияланып, оның тиімділігі бағаланды.

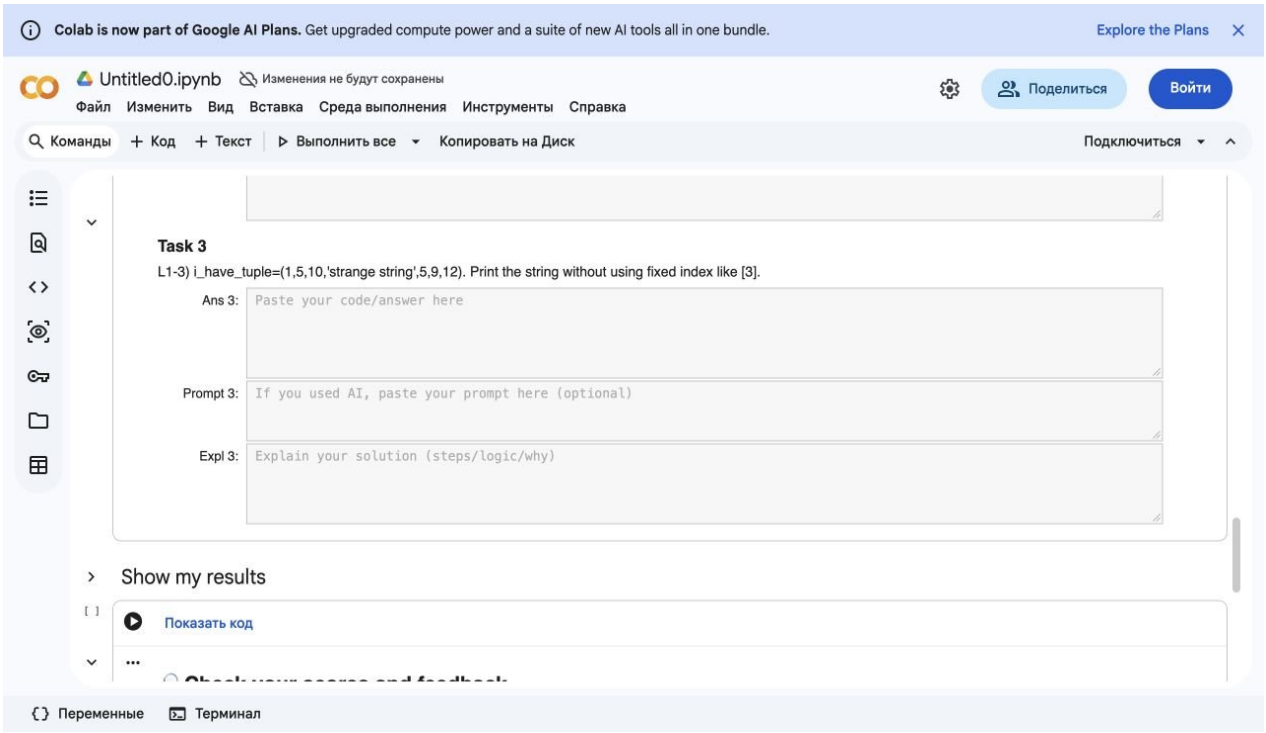
Эксперименттік-тексеру кезеңінде программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі білім алушылардың программалау тапсырмаларын орындау барысында жинақталған нақты деректер негізінде апробациядан өткізілді. Студенттердің программалық кодымен қатар, шешімнің логикасын сипаттайтын жазбаша түсіндірмелері және GenAI құралдары қолданылған жағдайда оларға берілген сұраныстар жинақталды. Аталған материалдар студенттік жұмыстарды бастапқы және интеллектуалдық модель негізіндегі тәсілдермен бағалаудың эмпирикалық негізін құрады.

Эксперименттік тексеру үшін үш күрделілік деңгейін қамтитын программалау тапсырмалары әзірленді (Қосымша Г). Тапсырмаларды орындау барысында студенттерге GenAI құралдарын көмекші құрал ретінде пайдалануға рұқсат берілді. Бағалауға қажетті материалдар Google Colab ортасындағы құрылымдалған тіркеу формасы арқылы жинақталды. Әр тапсырма бойынша студенттің программалық кодын немесе жауабын, GenAI қолданылған жағдайда оған берілген сұранысты және шешімнің жазбаша түсіндірмесін енгізуге арналған жеке өрістер қарастырылды. Тіркеу формасы 12-суретте берілген

а) Программалау тапсырмалары



ә) Студенттің жұмыстарын тіркеу формасы



Сурет 12 – Google Colab ортасында студенттік жұмыстарды тіркеу формасы

Форманың мұндай құрылымы студенттің программалық кодын, шешімнің жазбаша түсіндірмесін және GenAI қолданылуы туралы ұсынылған мәліметтерді өзара байланысты талдауға мүмкіндік берді. «Submit» батырмасы басылған кезде енгізілген деректер Google Sheets API арқылы орталық кестеге жіберілді. Сонымен қатар студенттердің тапсырмаларды орындау барысындағы

әрекеттерін сипаттайтын деректер арнайы тіркеу формасы арқылы жинақталды (Қосымша Ғ). Онда тапсырмаларды орындау жолының түсіндірмесі, кездескен қиындықтар, оқытушыға қойылған сұрақтар және GenAI құралдары қолданылған жағдайда оларға берілген нақты сұраныстар (prompt) тіркелді. Осылайша жинақталған деректер жиыны студенттік жұмыстардың функционалдық дұрыстығын, шешімді түсіндіру сапасын және GenAI құралдарын пайдалану ерекшеліктерін бағалауға негіз болды.

Жинақталған студенттік жұмыстарды бағалау ұйымдастырылды. Оқытушылар жұмыстарды 0–100 балдық шкала бойынша бағалап, әр жұмысқа қойылған тұтас баға кейінгі салыстыру үшін референттік мән ретінде қабылданды.

Студенттердің тапсырмаларды орындау нәтижелері GPT, Claude және Gemini үлкен тілдік модельдері арқылы екі тәсілмен бағаланды. Бірінші тәсілде программалық шешімнің функционалдық дұрыстығына негізделген бастапқы бағалау жүргізілді. Екінші тәсілде әзірленген интеллектуалдық модель негізінде функционалдық дұрыстық, шешімді түсіндіру сапасы және GenAI құралдарын пайдалану деңгейі біріктіріліп, бағалау жүзеге асырылды. Екі тәсілде де бірдей программалық кодтар, жазбаша түсіндірмелер және ұсынылған GenAI сұраныстары қамтылған деректер жиыны пайдаланылды. Бұл бағалау нәтижелерін ортақ бастапқы деректер негізінде салыстыруға мүмкіндік берді.

Интеллектуалдық модель негізіндегі бағалауда функционалдық дұрыстық программалық шешімнің тапсырма талаптарына сәйкестігін, алгоритмнің қисындылығын және орындалу нәтижесінің дұрыстығын сипаттады. Шешімді түсіндіру сапасы студенттің алгоритмнің негізгі қадамдарын, таңдалған тәсілдің себептерін және кодтың жұмыс істеу логикасын жазбаша негіздеуіне қарай бағаланды. GenAI құралдарын пайдалану деңгейі студент ұсынған сұраныстар негізінде қарастырылып, дайын шешімді қолдану, оны тапсырмаға бейімдеу және алынған нәтижені сыни талдау ерекшеліктерін қамтыды. Бұл ретте GenAI-мен өзара әрекеттесу жазбасының болмауы аталған құралдардың қолданылмағанын тікелей білдірмейді.

Екі бағалау тәсілі бойынша алынған нәтижелер оқытушының референттік бағаларымен салыстырылды. Бағалар арасындағы сызықтық және рангілік байланысты, абсолюттік ауытқу шамасын және жүйелі айырманың бағытын анықтау үшін Pearson және Spearman корреляция коэффициенттері, орташа абсолюттік қате (MAE) және орташа айырма (Mean Difference) көрсеткіштері қолданылды.

Аталған бағалау компоненттерінің қорытынды бағадағы үлесін ғылыми негіздеу үшін программалау пәндерін жүргізетін жоғары оқу орындарының 31 оқытушысының қатысуымен сараптамалық сауалнама жүргізілді. Сараптамалық бағалау функционалдық дұрыстық, шешімді түсіндіру сапасы және GenAI құралдарын пайдалану деңгейінен тұратын интеллектуалдық модельдің бағалау компоненттерінің салмақтық мәндерін анықтауға бағытталды. Сонымен қатар сараптамалық бағалау нәтижелері GenAI құралдарын пайдалану деңгейі мен шешімді түсіндіру сапасын сипаттайтын деңгейлік шкалалардың сандық мәндерін негіздеуге пайдаланылды. Сарапшылар пікірлерін жинақтауда олардың академиялық лауазымы мен оқытушылық тәжірибесіне сәйкес белгіленген салмақ коэффициенттері

ескерілді. Сараптамалық бағалардың салмақталған орташа мәні (1) формула бойынша анықталды.

(1)

$\bar{x}w$ – салмақталған орташа баға;
x_i – i-сарапшының бағасы;
w_i – сарапшының салмақ коэффициенті;
m = 31 – сарапшылар саны.

Үш бағалау компонентіне берілген пайыздық үлестердің қосындысы жекелеген жауаптарда 100%-ға дәл сәйкес келмегендіктен, әр сарапшының үлестері алдымен нормаланып, кейін (1) формула бойынша жинақталды (2):

(2)

p_{ij} – i-сарапшының j-компонентке берген бастапқы пайыздық үлесі;
p^*_{ij} – нормаланған үлес;
j=1,2,3 – үш компоненттің индексі.

Нормаланған сараптамалық бағаларды салмақталған орташа мән негізінде жинақтау нәтижесінде функционалдық дұрыстықтың салмағы – 36,26%, шешімді түсіндіру сапасының салмағы – 36,41%, ал GenAI құралдарын пайдалану деңгейінің салмағы – 27,33% болып анықталды. Анықталған салмақтық мәндер интеллектуалдық модельде студенттің программалаудан білімін кешенді бағалау кезінде аталған бағалау компоненттерінің қорытынды нәтижеге қосатын үлесін есептеуге негіз болды.

Осыған сәйкес студенттің тапсырмаларды орындау нәтижелері бойынша қорытынды бағасы (3) формуламен есептелді:

(3)

Мұндағы S – қорытынды баға; F – функционалдық дұрыстық бойынша баға; E – шешімді түсіндіру сапасы бойынша баға; A – GenAI құралдарын пайдалану деңгейі бойынша баға. Әр компонент 0–100 балдық шкалада бағаланды. Бұл формула үш компоненттің бағаларын сараптамалық тұрғыдан анықталған салмақтарына сәйкес бір қорытынды көрсеткішке біріктірді.

Интеллектуалдық модель негізіндегі бағалауда GPT, Claude және Gemini модельдеріне бірдей компоненттер, салмақ коэффициенттері және бағалау нұсқаулары ұсынылды. Студенттердің тапсырмаларды орындау нәтижелері модельдердің веб-интерфейстеріне Excel форматында жүктеліп, бағалау нәтижелері де осы форматта алынды. Әр модель әр бағалау тәсілі бойынша бір рет бағалау жүргізді.

Студенттік жұмыстарды екі бағалау тәсілі бойынша өңдеу нәтижелерінің сәйкестігін анықтау үшін оқытушының бағасы референттік мән ретінде қабылданды. Бастапқы және интеллектуалдық модель негізіндегі бағалау нәтижелері оқытушының референттік бағаларымен салыстырылды. Бағалау нәтижелерін статистикалық тұрғыдан салыстыруда Pearson және Spearman корреляция коэффициенттері, орташа абсолюттік қате (Mean Absolute Error,

MAE) және орташа айырма (Mean Difference) көрсеткіштері қолданылды. Бұл көрсеткіштер бағалау нәтижелерінің өзара байланысын ғана емес, олардың референттік бағадан ауытқу шамасы мен бағытын да кешенді сипаттауға мүмкіндік берді.

Оқытушының референттік бағалары мен үлкен тілдік модельдер (LLM) қалыптастырған бағалар арасындағы сызықтық байланысты анықтау үшін Pearson корреляция коэффициенті келесі формула бойынша есептелді (4):

(4)

Мұндағы X_i – i-студенттің тапсырмаларды орындау нәтижелеріне оқытушы қойған референттік баға; Y_i – сол нәтижелерге LLM қалыптастырған баға; $\bar{X}$ және $\bar{Y}$ – сәйкес бағалардың орташа мәндері; n = 187 – салыстыруға енгізілген студенттік жазбалар саны.

Оқытушы мен LLM бағаларының рангілік сәйкестігін анықтау үшін Spearman корреляция коэффициенті қолданылды (5):

(5)

мұндағы R_X – оқытушы бағаларының рангілері; R_Y – LLM бағаларының рангілері. Spearman коэффициенті бағалардың абсолюттік мәндерінен гөрі студенттік жұмыстардың бағалау нәтижелері бойынша орналасу реттілігінің қаншалықты сәйкес келетінін анықтауға мүмкіндік береді. Тең бағаларға орташа ранг берілді.

LLM қалыптастырған бағалардың оқытушының референттік бағаларынан абсолюттік ауытқу шамасын анықтау үшін орташа абсолюттік қате (MAE) есептелді (6):

(6)

MAE мәнінің төмен болуы LLM қалыптастырған бағалардың оқытушының референттік бағаларына абсолюттік тұрғыдан жақын екенін сипаттайды.

Сонымен қатар бағалардың жүйелі ауытқу бағытын анықтау үшін орташа айырма (Mean Difference) есептелді (7):

(7)

Орташа айырманың оң мәні LLM бағаларының оқытушының референттік бағаларынан орта есеппен жоғары, ал теріс мәні төмен болғанын көрсетеді. Көрсеткіштің нөлге жақындауы бағалаудағы жүйелі айырманың азаюын сипаттайды.

Осылайша, Pearson коэффициенті LLM мен оқытушы бағалары арасындағы сызықтық байланысты, Spearman коэффициенті бағалардың рангілік сәйкестігін, MAE олардың абсолюттік ауытқу шамасын, ал Mean Difference бағалардың жүйелі түрде жоғары немесе төмен қойылу бағытын сипаттады. Аталған көрсеткіштер бастапқы және интеллектуалдық модель негізіндегі бағалау тәсілдерінде GPT, Claude және Gemini модельдері

қалыптастырған нәтижелер үшін есептелді. Алынған салыстырмалы нәтижелер 19-кестеде берілген.

Бағалау тәсілдері арасындағы айырмашылықтар зерттеуде қосымша статистикалық тексерулер арқылы бағаланды: Pearson коэффициенттерінің өзгерісі Williams тестімен, Spearman коэффициенттерінің өзгерісі тәуелді корреляцияларға арналған Fisher z негізіндегі тестпен, ал абсолюттік қателердің айырмашылығы жұптық t-тестпен тексерілді. Mean Difference (MD) көрсеткіші модель бағалары мен оқытушының референттік бағалары арасындағы жүйелі ауытқудың бағыты мен шамасын сипаттау үшін пайдаланылды.

Кесте 19 – Функционалдық дұрыстық және интеллектуалдық модель негізіндегі бағалау нәтижелерінің оқытушының референттік бағасымен сәйкестігі

Көрсеткіш	GPT функционалдық дұрыстық	GPT интеллектуалдық модель	Claude функционалдық дұрыстық	Claude интеллектуалдық модель	Gemini функционалдық дұрыстық	Gemini интеллектуалдық модель
Pearson r	0,5754	0,6081	0,4784	0,6470	0,5787	0,6020
Spearman r_s	0,6014	0,5886	0,4493	0,6901	0,6082	0,6337
MAE	13,72	12,87	18,32	7,34	16,12	11,44
Mean Difference	+10,02	−11,63	+15,67	+4,12	+12,78	+8,09

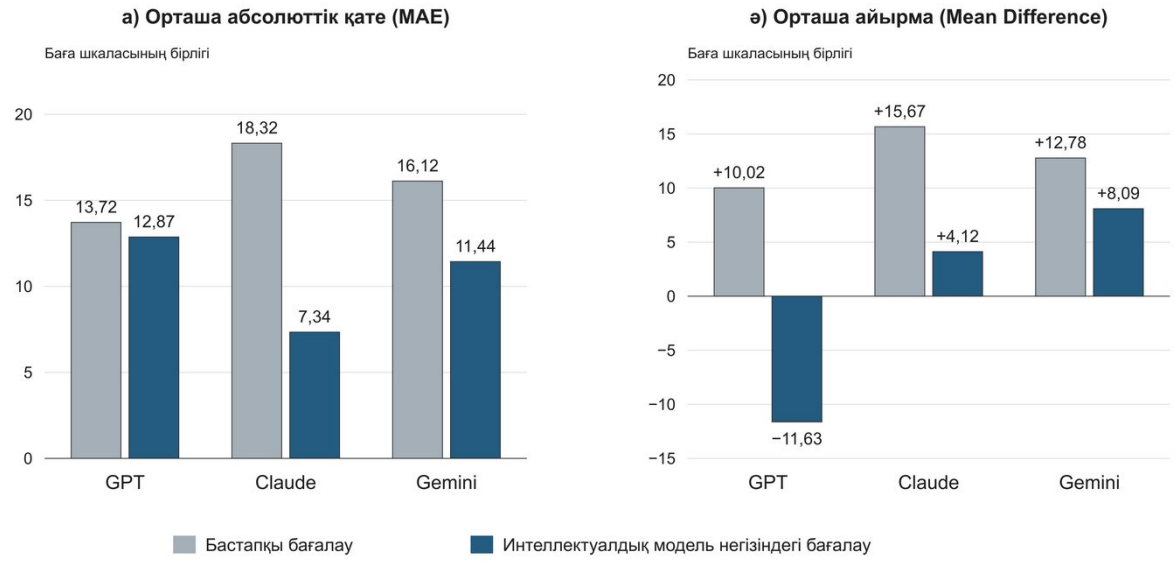
19-кестедегі нәтижелер бағалау құрылымын кеңейтудің үш тілдік модельде әртүрлі өзгерістермен сипатталғанын көрсетті. Claude моделінде Pearson коэффициенті 0,4784-тен 0,6470-ге, Spearman коэффициенті 0,4493-тен 0,6901-ге жоғарылады. MAE 18,32-ден 7,34-ке дейін төмендеп, абсолюттік қате шамамен 59,9%-ға азайды. MD +15,67-ден +4,12-ге жақындады. Жүргізілген статистикалық тексерулер бойынша екі корреляция коэффициентінің өсуі және абсолюттік қатенің төмендеуі статистикалық мәнді болды ($p < 0,001$).

Gemini моделінде Pearson коэффициенті 0,5787-ден 0,6020-ге, Spearman коэффициенті 0,6082-ден 0,6337-ге жоғарылады. Алайда бұл өзгерістер статистикалық мәнді болған жоқ: сәйкесінше $p = 0,654$ және $p = 0,592$. MAE 16,12-ден 11,44-ке төмендеп, абсолюттік қате шамамен 29,0%-ға азайды; бұл айырмашылық статистикалық мәнді болды ($p < 0,001$). MD +12,78-ден +8,09-ға төмендеді. Демек, Gemini бойынша оң нәтиже ең алдымен абсолюттік бағалау қатесінің азаюынан көрінді.

GPT моделінде Pearson коэффициенті 0,5754-тен 0,6081-ге жоғарылап, MAE 13,72-ден 12,87-ге төмендеді, ал Spearman коэффициенті 0,6014-тен 0,5886-ға азайды. Бұл өзгерістер статистикалық мәнді болған жоқ: Pearson үшін $p = 0,524$, Spearman үшін $p = 0,791$, абсолюттік қате үшін $p = 0,457$. MD +10,02-ден −11,63-ке өзгеріп, жүйелі ауытқу жоғары бағалаудан төмен бағалау

бағытына ауысты. Сондықтан GPT бойынша бағалау дәлдігі жақсарды деген біржақты қорытынды жасауға негіз жоқ.

Бағалау тәсілдері арасындағы абсолюттік және жүйелі ауытқулардың өзгерісі 13-суретте көрсетілген.



Сурет 13 – Бастапқы және интеллектуалдық модель негізіндегі бағалау нәтижелерінің абсолюттік және жүйелі ауытқуларын салыстыру

Ескерту: а) MAE мәнінің төмен болуы абсолюттік қатенің аз екенін білдіреді; ә) MD мәнінің нөлге жақындауы орташа жүйелі ауытқудың азаюын, ал таңбасы оның бағытын көрсетеді.

Эксперимент нәтижелері әзірленген интеллектуалдық модель негізіндегі бағалаудың тиімділігін көрсетті. Функционалдық дұрыстықпен шектелетін бастапқы бағалаумен салыстырғанда, функционалдық дұрыстықты, шешімді түсіндіру сапасын және GenAI құралдарын пайдалану деңгейін кешенді ескеру Claude және Gemini модельдерінде оқытушының референттік бағаларымен сәйкестіктің жоғарылауына және бағалау ауытқуларының азаюына мүмкіндік берді. GPT моделінде жекелеген көрсеткіштер бойынша оң өзгерістер байқалғанымен, барлық статистикалық көрсеткіштер бойынша бір бағыттағы жақсару анықталған жоқ. Осылайша, эксперимент нәтижелері әзірленген интеллектуалдық модельді программалаудан студенттердің білімін бақылауда қолданудың тиімділігін көрсетті және оның нәтижелілігі модель компоненттерінің кешенді іске асырылуымен қатар, интеллектуалдық талдауда қолданылатын үлкен тілдік модельдің ерекшеліктеріне де байланысты екенін айқындады.

Алынған нәтижелер программалаудан білімді бақылауды студенттің тапсырманы орындауының соңғы нәтижесімен ғана шектемей, функционалдық дұрыстықты, шешімді түсіндіру сапасын және GenAI құралдарын пайдалану деңгейін өзара байланыста ескеру студенттің программалаудан білім жағдайын кешенді бағалауға мүмкіндік беретінін көрсетті.

2-тарау бойынша қорытынды

Екінші бөлімде программалаудан студенттердің білімін бақылаудың интеллектуалдық моделін жасау, оны оқу үдерісінде іске асыру және эксперименттік тексеру мәселелері қарастырылды. Бірінші бөлімде анықталған теориялық қағидалар, бағалау критерийлері мен көрсеткіштер негізінде программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі жасалды. Модель педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттерден тұрады. Компоненттердің өзара байланысы бақылау мақсатын анықтаудан бастап студенттің программалау әрекеті туралы деректерді жинау, оларды талдау, бағалау нәтижелерін интерпретациялау және педагогикалық шешім қабылдауға дейінгі бақылау үдерісінің бірізділігін қамтамасыз етеді.

Модельді іске асыруда студенттің программалау тапсырмасын орындауы барысында алынатын деректер маңызды орын алады. Оларға программалық код, тестілеу және орындалу нәтижелері, жіберілген қателер мен оларды түзету әрекеттері, тапсырманы орындау тарихы, алгоритмдік түсіндірме және генеративті жасанды интеллектпен өзара әрекеттесу деректері жатады. Бұл деректер студенттің тапсырманы орындаудағы соңғы нәтижесімен қатар, программалау әрекетінің орындалу барысын да сипаттайды. Оларды автоматтандырылған тестілеу, статикалық және динамикалық талдау, үдерістік аналитика және үлкен тілдік модельдер негізіндегі семантикалық талдау арқылы өңдеу бағалау көрсеткіштері бойынша студент білімінің жай-күйін анықтауға мүмкіндік береді.

Интеллектуалдық модель негізінде программалаудан студенттердің білімін бақылауды ұйымдастыру әдістері әзірленді. Бақылау үдерісі педагогикалық мақсатты анықтау, критерийлер мен көрсеткіштерді белгілеу, программалау тапсырмасын ұсыну, студент әрекетінен алынған деректерді жинау, оларды интеллектуалдық талдау, көрсеткіштер бойынша бағалау, алынған нәтижелерді педагогикалық тұрғыдан интерпретациялау және кері байланыс беру әрекеттерінің өзара байланысы негізінде жүзеге асырылады. Мұндай ұйымдастыру студенттің программалаудан білімін бір ғана қорытынды баллмен шектемей, оның программалау әрекетінің жекелеген көрсеткіштері бойынша нәтижесін анықтауға және оқытушыға студенттің білім жағдайы туралы толық ақпарат беруге бағытталған.

Зерттеу нәтижелерін практикалық тұрғыдан қолдану мақсатында программалаудан студенттердің білімін бақылауда пайдалануға арналған «Bilim AI» бағдарламалық өнімі әзірленді [56]. Бағдарламалық өнім ұсынылған бақылау тәсілдерін цифрлық ортада қолдануға мүмкіндік беретін практикалық құрал ретінде қарастырылды. Бағдарламалық өнімде генеративті талдау Llama 3.3 70B үлкен тілдік моделі арқылы жүзеге асырылады [134].

Ұсынылған интеллектуалдық модельді эксперименттік тексеру бірнеше кезеңде жүргізілді. Диагностикалық кезеңде программалаудан білімді бақылаудың қолданыстағы жағдайы мен студенттердің программалау тапсырмаларын орындау ерекшеліктері зерттелді. Сараптамалық-негіздеуші кезеңде программалау әрекетін бағалауға қолданылатын көрсеткіштердің

мазмұны мен маңыздылығы нақтыланды. Эксперименттік-тексеру кезеңінде студенттердің программалау тапсырмаларын орындау нәтижелері мен үдерісі туралы деректер негізінде интеллектуалдық бақылаудың мүмкіндіктері тексеріліп, алынған нәтижелер оқытушы бағалауымен салыстырылды. Эксперимент нәтижелерін сандық және сапалық талдау ұсынылған модельдің программалаудан студенттердің білімін бақылауда қолданылу мүмкіндігін көрсетті.

ҚОРЫТЫНДЫ

Жүргізілген зерттеу жоғары оқу орнында студенттердің программалаудан білімін бақылаудың қазіргі жағдайын зерделеуге және оны интеллектуалдық модель негізінде ұйымдастыруға бағытталды. Теориялық талдау программалаудан білімді бақылауда студент орындаған тапсырманың соңғы нәтижесімен ғана шектелмей, есепті түсіну, алгоритм құру, кодты жазу, тексеру, қателерді анықтау мен түзету және қабылданған шешімді түсіндіру сияқты программалау әрекеттерін де ескеру қажеттігін көрсетті.

Автоматтандырылған бақылау жүйелері программалық кодтың функционалдық дұрыстығын тексеруге және жедел кері байланыс беруге мүмкіндік береді. Алайда интеллектуалдық оқу құралдары мен генеративті жасанды интеллект технологияларының дамуы жағдайында дайын программалық код студенттің білім деңгейін толық сипаттайтын жалғыз көрсеткіш бола алмайды. Сондықтан программалаудан білімді бақылауда программалық кодпен бірге тапсырманы орындау тарихын, жіберілген қателер мен оларды түзету әрекеттерін, алгоритмдік түсіндірмені және генеративті ЖИ-мен өзара әрекеттесу деректерін пайдалану қажеттігі айқындалды.

Зерттеу нәтижесінде программалаудан білімді интеллектуалдық бақылаудың педагогикалық мақсаттылық және валидтілік; көпкомпоненттілік және көпдеректілік; нәтиже мен үдерістің бірлігі; критерийлік негізділік және түсіндірмелілік; педагогикалық шешім мен интеллектуалдық қолдаудың бірлігі қағидалары анықталды. Осы қағидалар негізінде студенттің есептің мазмұнын түсінуін, алгоритмдік шешім құруын, программалық шешімнің функционалдық дұрыстығын, код сапасын, қателерді анықтау мен түзетуін, шешімді түсіндіруі мен негіздеуін бағалауға арналған критерийлер мен көрсеткіштер нақтыланды.

Анықталған қағидалар, бағалау критерийлері мен көрсеткіштер негізінде педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттерден тұратын программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі жасалды. Модель студенттің программалау әрекетінен алынған әртүрлі деректерді жинақтау, интеллектуалдық талдау, бағалау және алынған нәтижелерді педагогикалық тұрғыдан интерпретациялау үдерістерін өзара байланыстырады.

Интеллектуалдық модель негізінде программалаудан білімді бақылауды ұйымдастыру әдістері әзірленіп, зерттеу нәтижелерін практикалық қолдануға арналған «Bilim AI» бағдарламалық өнімі жасалды. Ұсынылған модель диагностикалық, сараптамалық-негіздеуші және эксперименттік-тексеру кезеңдері арқылы тексерілді. Эксперимент барысында студенттердің программалау тапсырмаларын орындау нәтижесі мен үдерісі туралы деректер талданып, интеллектуалдық бақылау нәтижелері оқытушы бағалауымен салыстырылды.

Зерттеу нәтижелері бойынша мынадай негізгі қорытындылар жасалды:

– Программалаудан студенттердің білімін бақылаудың қазіргі жағдайы талданып, интеллектуалдық оқу құралдарын қолдану жағдайындағы бақылаудың ерекшеліктері айқындалды. Студенттің білімін бағалауда

тапсырманың соңғы нәтижесімен бірге программалау әрекетінің үдерісін сипаттайтын деректерді ескеру қажеттігі негізделді.

– Программалаудан білімді интеллектуалдық бақылаудың қағидалары анықталып, студенттің программалау әрекетінің нәтижесі мен үдерісін бағалауға арналған критерийлер мен көрсеткіштер нақтыланды.

– Педагогикалық-мақсаттық, критерийлік-бағалау, деңгейлік-тапсырма, интеллектуалдық-талдау және нәтижелік-интерпретациялық компоненттердің өзара байланысына негізделген программалаудан студенттердің білімін бақылаудың интеллектуалдық моделі жасалды.

– Интеллектуалдық модель негізінде программалаудан студенттердің білімін бақылауды ұйымдастыру әдістері әзірленіп, оларды практикалық қолдануға арналған «Bilim AI» бағдарламалық өнімі жасалды.

– Интеллектуалдық модель эксперименттік тұрғыдан тексеріліп, программалаудан студенттердің білімін объективті және кешенді бақылауда қолдану тиімділігі бағаланды.

Жүргізілген зерттеу нәтижелері қойылған міндеттердің шешілгенін және зерттеу мақсатына қол жеткізілгенін көрсетеді. Алдағы зерттеулерде интеллектуалдық модельді әртүрлі программалау тілдері мен оқу курстарына бейімдеу, генеративті жасанды интеллектпен өзара әрекеттесу деректерін талдау тәсілдерін жетілдіру және бақылау нәтижелері негізінде студентке дараланған педагогикалық қолдау көрсету мәселелерін қарастыруға болады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 «Білім туралы» Қазақстан Республикасының Заңы (27.07.2007 ж. № 319 III; 01.03.2025 ж.).https://adilet.zan.kz/kaz/docs/Z070000319_

2 «Жоғары және жоғары оқу орнынан кейінгі білім берудің мемлекеттік жалпыға міндетті стандарттарын бекіту туралы», ҚР Ғылым және жоғары білім министрінің 2022 жылғы 20 шілдедегі №2 бұйрығы https://adilet.zan.kz/kaz/docs/V2200028916

3 «Жоғары және (немесе) жоғары оқу орнынан кейінгі білім беру ұйымдарында оқытудың кредиттік технологиясы бойынша оқу процесін ұйымдастыру қағидаларын бекіту туралы», ҚР БҒМ 2011 жылғы 20 сәуірдегі №152 бұйрығы https://adilet.zan.kz/kaz/docs/V2300033173

4 «Жоғары және (немесе) жоғары оқу орнынан кейінгі білім беру ұйымдары қызметінің үлгілік қағидаларын бекіту туралы», ҚР БҒМ 2018 жылғы 30 қазандағы №595 бұйрығы https://adilet.zan.kz/kaz/docs/V1800017657

5 Қазақстан Республикасында жоғары білім мен ғылымды дамытудың 2023-2029 жылдарға арналған тұжырымдамасын бекіту туралы: ҚР Үкіметінің 2023 жылғы 28 наурыздағы № 248 қаулысы. – URL: https://adilet.zan.kz/kaz/docs/P2300000248.

6 Жасанды интеллектті дамытудың 2024-2029 жылдарға арналған тұжырымдамасын бекіту туралы: ҚР Үкіметінің 2024 жылғы 24 шілдедегі № 592 қаулысы. – URL: https://adilet.zan.kz/kaz/docs/P2400000592.

7 Қазақстан Республикасының «Жасанды интеллект туралы» Заңы: 2025 жылғы 17 қарашадағы № 230-VIII ҚРЗ. – URL: https://adilet.zan.kz/kaz/docs/Z2500000230 .

8 Бекежанова А.А., Бидайбеков Е.Ы. Определение содержания обучения будущих учителей информатики объектно-ориентированному программированию с использованием инфографики // Вестник КазНПУ имени Абая. Серия «Физико-математические науки». – 2021. – Т. 75. – №3. DOI: 10.51889/2021-3.1728-7901.22.3

9 Нурбекова Ж.К. Фундаментальное и опережающее обучение программированию студентов по специальности «Информатика»: автореф. ... док. пед. наук: 13.00.02. – Алматы, 2007. – 40 с.

10 Жангисина Г.Д. Методологические основы преподавания языков программирования в высшей школе: автореф. ... док. пед. наук: 13.00.02. – Караганда, 2002. – 55 с.

11 Нугманова С.А. Обучение теории параллельных вычислений будущих учителей информатики на основе системы упражнений и задач: дис. … канд. пед. наук: 13.00.02. – Алматы, 2009. – 157 с.

12 Хакимова Т. Методические и содержательные аспекты использования структурного подхода при изучении языков программирования студентами-математиками: автореф. … канд. пед. наук: 13.00.02. – Алматы, 1999. – 22 с.

13 Кожахметов С.Т. Математико-методическое обеспечение обучения программированию на основе объектно-ориентированного подхода при подготовке специалистов по информатике: 13.00.02: дис. ... канд. пед. наук. – Алматы, 1999. – 129 с.

14 Байбақтина А.Т. Болашақ информатика оқытушыларына программалау негіздерін оқытуды жобалар әдісі бойынша жетілдіру: 13.00.02: пед. ғыл. канд. …автореф. – Алматы, 2007. – 20 б.

15 Байгушева К.М. Методика обучения будущих информатиков программированию для Интернета: автореф. ... канд. пед. наук: 13.00.02. – Алматы, 2007. – 21 с.

16 Бекежанова А.А. Болашақ информатика мұғалімдеріне инфографика құралдарын пайдаланып объектіге-бағытталған программалауды оқыту әдістемесі: 6D011100: док. PhD. ... дис. – Алматы, 2024.

17 Аймичева Г.И. Формирование логической компетенции по проектированию мобильных приложений у студентов по специальности «Информатика»: 6D011100: док. PhD. ... дис. – Алматы, 2025. – 192 с.

18 Гриншкун В.В. Инструментальная поддержка преподавания алгоритмических языков на базе древовидного представления информации // Вестник научно-педагогического центра. – Алматы, 1994. – № 15. – С. 7.

19 Шкарбан Ф.В. Методика обучения основам объектно-ориентированного программирования бакалавров прикладной информатики с использованием визуальных учебных сред: 13.00.02: дис. ... канд. пед. наук. – Волгоград, 2018. – 212 с.

20 Левченко И.В., Садыкова А.Р., Абушкин Д.Б., Карташова Л.И., Кондратьева В.А., Моисеев В.П. Особенности подготовки по программированию будущих учителей информатики // Вестник Российского университета дружбы народов. Серия: Информатизация образования. – 2021. – Т. 18. – № 4. – С. 337–346. – DOI: 10.22363/2312-8631-2021-18-4-337-346.

21 Федотенко М.А. Совершенствование методики обучения объектно-ориентированному программированию будущих учителей информатики посредством разработки образовательных мобильных приложений: 5.8.2: дис. ... канд. пед. наук. – Москва, 2021. – 205 с.

22 Rodrigues G., Monteiro A.F., Osório A. Introductory Programming in Higher Education: A Systematic Literature Review // Third International Computer Programming Education Conference (ICPEC 2022). Open Access Series in Informatics. – 2022. – Vol. 102. – P. 4:1–4:17. – DOI: 10.4230/OASIcs.ICPEC.2022.4.

23 Calderon I., Silva W., Feitosa E. Active Learning Methodologies for Teaching Programming in Undergraduate Courses: A Systematic Mapping Study // Informatics in Education. – 2024. – Vol. 23. – No. 2. – P. 279–322. – DOI: 10.15388/infedu.2024.11.

24 Shoufan A. Rethinking Programming Education: A Lecture-Free Approach // IEEE Access. – 2025. – Vol. 13. – P. 34758–34767. – DOI: 10.1109/ACCESS.2025.3542389.

25 Luxton-Reilly A., Simon, Albluwi I., Becker B.A., Giannakos M., Kumar A.N., Ott L., Paterson J., Scott M.J., Sheard J., Szabo C. Introductory programming: a systematic literature review // Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018 Companion). – New York: ACM, 2018. – P. 55-106. – DOI: 10.1145/3293881.3295779.

26 Medeiros R.P., Ramalho G.L., Falcão T.P. A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education // IEEE Transactions on Education. – 2019. – Vol. 62, No. 2. – P. 77–90. – DOI: 10.1109/TE.2018.2864133

27 Kalelioğlu F., Keskinkılıç F. Difficulties of Learning Programming in Higher Education: A Systematic Literature Review of Recent Research // Ankara University Journal of Faculty of Educational Sciences. – 2026. – Vol. 59, No. 2. – P. 771–835. – DOI: 10.30964/auebfd.1703365.

28 Robins A., Rountree J., Rountree N. Learning and teaching programming: a review and discussion // Computer Science Education. – 2003. – Vol. 13, No. 2. – P. 137-172. – DOI: 10.1076/csed.13.2.137.14200.

29 McCracken M., Almstrum V., Diaz D., Guzdial M., Hagan D., Kolikant Y.B.-D., Laxer C., Thomas L., Utting I., Wilusz T. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students // ACM SIGCSE Bulletin. – 2001. – Vol. 33, No. 4. – P. 125-180. – DOI: 10.1145/572139.572181.

30 Абдильдинова Г.М. Методика создания и использования электронного средства контроля знаний студентов по программированию на основе теории экспертных систем: автореф. ... канд. пед. наук: 13.00.02. – Алматы, 2010. – 21 с.

31 Токжигитова Н.К. Визуалды программалау бойынша болашақ информатика мамандарының оқу жетістіктерін мульти-критериалды бағалаудың әдіснамасы: 6D011100: PhD дис. – Астана, 2017. – 135 б.

32 А.Е. Сағымбаева, Ж.К. Нурбекова, А.Е. Жаксылыков. GenAI дәуірінде болашақ IT мамандарының программалау дағдыларын бағалау әдістері // Қазақстан математиктерінің бірінші конгресі : тезистер жинағы. – Шымкент, 2026. – Б. 377–379

33 Ala-Mutka K.M. A survey of automated assessment approaches for programming assignments // Computer Science Education. – 2005. – Vol. 15, No. 2. – P. 83-102. – DOI: 10.1080/08993400500150747.

34 Keuning H., Heeren B., Jeuring J. Strategy-based feedback in a programming tutor // Proceedings of the Computer Science Education Research Conference (CSERC ’14). – New York: ACM, 2014. – P. 43–54. – DOI: 10.1145/2691352.2691356.

35 Ihantola P., Ahoniemi T., Karavirta V., Seppälä O. Review of recent systems for automatic assessment of programming assignments // Proceedings of the 10th Koli Calling International Conference on Computing Education Research. – New York: ACM, 2010. – P. 86-93. – DOI: 10.1145/1930464.1930480.

36 Paiva J.C., Leal J.P., Figueira Á. Automated assessment in computer science education: a state-of-the-art review // ACM Transactions on Computing Education. – 2022. – Vol. 22, No. 3. – Article 34. – DOI: 10.1145/3513140.

37 Combéfis S. Automated code assessment for education: review, classification and perspectives on techniques and tools // Software. – 2022. – Vol. 1, No. 1. – P. 3-30. – DOI: 10.3390/software1010002.

38 Messer M., Brown N.C.C., Kölling M., Shi M. Automated grading and feedback tools for programming education: a systematic review // ACM Transactions

on Computing Education. – 2024. – Vol. 24, No. 1. – Article 10. – DOI: 10.1145/3636515.

39 Педагогика: учебное пособие для студентов педагогических вузов и педагогических колледжей / под ред. П.И. Пидкасистого. — М.: Педагогическое общество России, 1998. — 640 с.

40 Подласый И.П. Педагогика. Новый курс. Кн. 1: Общие основы. Процесс обучения. — М.: ВЛАДОС, 1999. — 576 с.

41 Басова Н.В. Педагогика и практическая психология. — Ростов н/Д: Феникс, 2000. — С. 251–255.

42 Black P., Wiliam D. Developing the theory of formative assessment // Educational Assessment, Evaluation and Accountability. – 2009. – Vol. 21. – P. 5–31. DOI: 10.1007/s11092-008-9068-5.

43 Harlen W. Assessment of Learning. – London: SAGE Publications, 2007. – 176 p. DOI:10.4135/9781446214695.

44 Messick S. Validity of psychological assessment: validation of inferences from persons' responses and performances as scientific inquiry into score meaning // American Psychologist. – 1995. – Vol. 50, No. 9. – P. 741-749. – DOI: 10.1037/0003-066X.50.9.741.

45 Anderson L.W., Krathwohl D.R. (Eds.). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. – New York: Longman, 2001. – 352 p. // https://davidpassmore.net/courses/data/Anderson-Krathwohl.pdf

46 Сағымбаева А.Е. Болашақ информатика мұғалімдерін оқушылардың білімін бақылау мен бағалауға дайындаудың теориялық-әдістемелік негіздері: 13.00.02: пед. ғыл. док. дис. – Алматы, 2010. – 280 б.

47 Corbett A.T., Anderson J.R. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge // User Modeling and User-Adapted Interaction. – 1994. – Vol. 4. – P. 253–278. DOI: 10.1007/BF01099821.

48 Piech C., Bassen J., Huang J., Ganguli S., Sahami M., Guibas L.J., Sohl-Dickstein J. Deep Knowledge Tracing // Advances in Neural Information Processing Systems. – 2015. – Vol. 28. DOI: 10.48550/arXiv.1506.05908.

49 Shi Y., Chi M., Barnes T., Price T.W. Code-DKT: A Code-Based Knowledge Tracing Model for Programming Tasks // Proceedings of the 15th International Conference on Educational Data Mining. – 2022. – P. 50–61. 10.48550/arXiv.2206.03545

50 Сағымбаева А.Е., Жақсылықов А.Е., Шекербекова Ш.Т., Жамкеева А.Б. Генеративті жасанды интеллект технологиясының программалаудан ЖОО студенттерінің білімін бақылаудағы рөлі // Абай атындағы ҚазҰПУ Хабаршысы. «Физика-математика ғылымдары» сериясы. – 2024. – № 3 (87). – Б. 320-330. – DOI: 10.51889/2959-5894.2024.87.3.027.

51 Nurbekova Z., Sagimbayeva A., Zhaxylykov A., Bidaibekov E. GenAI potential and challenges in programming knowledge control // International Journal of Innovative Research and Scientific Studies. – 2025. – Vol. 8, No. 5. – P. 157–167. – DOI: 10.53894/ijirss.v8i5.8577.

52 Kohen-Vacs D., Usher M., Jansen M. Integrating generative AI into programming education: student perceptions and the challenge of correcting AI errors

// International Journal of Artificial Intelligence in Education. – 2025. – Vol. 35, No. 5. – P. 3166-3184. – DOI: 10.1007/s40593-025-00496-4.

53 Zhang Z., Dong Z., Shi Y., Price T., Matsuda N., Xu D. Students' Perceptions and Preferences of Generative Artificial Intelligence Feedback for Programming // Proceedings of the AAAI Conference on Artificial Intelligence. – 2024. – Vol. 38, No. 21. – P. 23250–23258. – DOI: 10.1609/aaai.v38i21.30372.

54 Silva P., Costa E. Assessing Large Language Models for Automated Feedback Generation in Learning Programming Problem Solving // Proceedings of the Innovation and Responsibility in AI-Supported Education Workshop. Proceedings of Machine Learning Research. – 2025. – Vol. 273. – P. 116–124.

55 Нурбекова Ж.К., Сагимбаева А.Е., Жаксылыков А.Е., Абильдинова Г.М., Рахимжанов Е.Е. Многоуровневая семантическая информационная система контроля результатов обучения: авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізіліміне мәліметтерді енгізу туралы куәлік № 2542. – 31.11.2017.

56 Жаксылыков А.Е., Нурбекова Ж.К., Сагимбаева А.Е., Ануарбек Д.Б., Bilim AI: авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізіліміне мәліметтерді енгізу туралы куәлік № 76071. – 02.07.2026.

57 Cheng L. Diagnostic Assessments // The TESOL Encyclopedia of English Language Teaching / eds. J. I. Liontas et al. – John Wiley & Sons, 2019. DOI: 10.1002/9781118784235.eelt0493.

58 Энциклопедия профессионального образования: в 3 т. Т. 1: А–Л / рук. авт. коллектива, науч. и лит. ред. С. Я. Батышев. – М.: Российская академия образования; Ассоциация «Профессиональное образование», 1998. – 567 с.

59 Бидайбеков Е.Ы., Нурбекова Ж.К., Сагимбаева А.Е. Информационный подход в оценивании качества знаний // Качество педагогического образования: проблемы и перспективы развития: матер. междунар. науч.-практ. конф. – Алматы, 2004. – С. 810–816.

60 Бидайбеков Е.Ы., Балыкбаев Т.О., Ибрагимова Н.Ж. Методические основы измерения результатов обучения школьников по информатике. – Алматы, 2007. – 154 с.

61 Леонтьев А.Н. Деятельность. Сознание. Личность. – М.: Политиздат, 1975. – 304 с.

62 Айтпаева А.К. Деятельностный подход – практико-ориентированная тактика подготовки будущих педагогов // Scientific Collection «InterConf». – 2021. – № 49. – С. 231–236. – DOI: 10.51582/interconf.7-8.04.2021.022.

63 Scanlon E., Issroff K. Activity theory and higher education: evaluating learning technologies // Journal of Computer Assisted Learning. – 2005. – Vol. 21, No. 6. – P. 430-439. – DOI: 10.1111/j.1365-2729.2005.00153.x.

64 Нурбекова Ж.К. Теоретико-методологические основы обучения программированию: монография. – Павлодар, 2004. – 225 с.

65 Malik S.I., Mathew R., Al-Sideiri A., Jabbar J., Al-Nuaimi R., Tawafak R.M. Enhancing problem-solving skills of novice programmers in an introductory programming course // Computer Applications in Engineering Education. – 2022. – Vol. 30, No. 1. – P. 174-194. – DOI: 10.1002/cae.22450.

66 Deek F.P., Turoff M., McHugh J.A. A common model for problem solving and program development // IEEE Transactions on Education. – 1999. – Vol. 42, No. 4. – P. 331-336. – DOI: 10.1109/13.804541.

67 Kurniawan O., Jégourel C., Lee N.T.S., De Mari M., Poskitt C.M. Steps before syntax: helping novice programmers solve problems using the PCDIT framework // Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS 2022). – 2022. – P. 982-991. – DOI: 10.24251/HICSS.2022.121.

68 Jadud M.C. Methods and tools for exploring novice compilation behaviour // Proceedings of the Second International Workshop on Computing Education Research (ICER '06). – New York: ACM, 2006. – P. 73-84. – DOI: 10.1145/1151588.1151600.

69 Villamor M.M. A review on process-oriented approaches for analyzing novice solutions to programming problems // Research and Practice in Technology Enhanced Learning. – 2020. – Vol. 15. – Article 8. – DOI: 10.1186/s41039-020-00130-y.

70 Acuña R., Bansal A. Assessing student programming process using automated reasoning // 2023 IEEE Frontiers in Education Conference (FIE). – IEEE, 2023. – DOI: 10.1109/FIE58773.2023.10342936.

71 Yang S., Baird M., O'Rourke E., Brennan K., Schneider B. Decoding debugging instruction: a systematic literature review of debugging interventions // ACM Transactions on Computing Education. – 2024. – Vol. 24, No. 4. – Article 45. – P. 1-44. – DOI: 10.1145/3690652.

72 Lehtinen T., Lukkarinen A., Haaranen L. Students struggle to explain their own program code // Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21). – New York: ACM, 2021. – Vol. 1. – P. 206-212. – DOI: 10.1145/3430665.3456322.

73 Bennedsen J., Caspersen M.E. Assessing process and product: a practical lab exam for an introductory programming course // Innovation in Teaching and Learning in Information and Computer Sciences. – 2007. – Vol. 6, No. 4. – P. 183-202. – DOI: 10.11120/ital.2007.06040183.

74 Wang Y., Li H., Feng Y., Jiang Y., Liu Y. Assessment of programming language learning based on peer code review model: implementation and experience report // Computers & Education. – 2012. – Vol. 59, No. 2. – P. 412-422. – DOI: 10.1016/j.compedu.2012.01.007.

75 Barczak A.L.C., Mathrani A., Han B., Reyes N.H. Automated assessment system for programming courses: a case study for teaching data structures and algorithms // Educational Technology Research and Development. – 2023. – Vol. 71, No. 6. – P. 2365-2388. – DOI: 10.1007/s11423-023-10277-2.

76 Zhaxylykov A.E., Nurbekova Zh., Sagimbayeva A., Shekerbekova S. Comparative Analysis of GenAI Models in Problem Solving on the LeetCode Platform // 2025 10th International Conference on Computer Science and Engineering (UBMK). – Istanbul, Türkiye: IEEE, 2025. – P. 131–136. DOI: 10.1109/UBMK67458.2025.11206938.

77 Keuning H., Jeuring J., Heeren B. A systematic literature review of automated feedback generation for programming exercises // ACM Transactions on Computing Education. – 2018. – Vol. 19, No. 1. – Article 3. – P. 1-43. – DOI: 10.1145/3231711.

78 Livaja Mušac P., Nakić J., Sović Kržić A. Automatic assessment tools for grading coding assignments: a systematic literature review // Applied Sciences. – 2026. – Vol. 16, No. 11. – Article 5658. – DOI: 10.3390/app16115658.

79 Wu M., Mosse M., Goodman N., Piech C. Zero shot learning for code education: rubric sampling with deep learning inference // Proceedings of the AAAI Conference on Artificial Intelligence. – 2019. – Vol. 33, No. 1. – P. 782-790. – DOI: 10.1609/aaai.v33i01.3301782.

80 Mendonça P.C., Quintal F., Mendonça F. Evaluating LLMs for automated scoring in formative assessments // Applied Sciences. – 2025. – Vol. 15, No. 5. – Article 2787. – DOI: 10.3390/app15052787.

81 Sarsa S., Denny P., Hellas A., Leinonen J. Automatic generation of programming exercises and code explanations using large language models // Proceedings of the 2022 ACM Conference on International Computing Education Research (ICER ’22). – New York: ACM, 2022. – Vol. 1. – P. 27-43. – DOI: 10.1145/3501385.3543957.

82 Phung T., Pădurean V.-A., Singh A., Brooks C., Cambronero J., Gulwani S., Singla A., Soares G. Automating human tutor-style programming feedback: leveraging GPT-4 tutor model for hint generation and GPT-3.5 student model for hint validation // Proceedings of the 14th Learning Analytics and Knowledge Conference (LAK’24). – New York: ACM, 2024. – P. 12-23. – DOI: 10.1145/3636555.3636846.

83 Becker B.A., Denny P., Finnie-Ansley J., Luxton-Reilly A., Prather J., Santos E.A. Programming is hard – or at least it used to be: educational opportunities and challenges of AI code generation // Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE 2023). – New York: ACM, 2023. – Vol. 1. – P. 500-506. – DOI: 10.1145/3545945.3569759.

84 Prather J., Denny P., Leinonen J., Becker B.A., Albluwi I., Craig M., Keuning H., Kiesler N., Kohn T., Luxton-Reilly A., MacNeil S., Petersen A., Pettit R., Reeves B.N., Savelka J. The robots are here: navigating the generative AI revolution in computing education // Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR ’23). – New York: ACM, 2023. – P. 108-159. – DOI: 10.1145/3623762.3633499.

85 Liu D., Fan G., Pan L. Tool, tutor, or crutch?: a grounded theory of cognitive scaffolding and offloading in AI-assisted programming education // International Journal of STEM Education. – 2026. – Vol. 13. – Article 10. – DOI: 10.1186/s40594-025-00592-w.

86 А.Е. Сағымбаева, А.Е. Жаксылыков Білімді бақылаудың заманауи ақпараттандыру құралдарын салыстырмалы талдау // Л.Н. Гумилев атындағы еуразия ұлттық университеті «Интеллектуалдық ақпараттық және коммуникациялық технологиялар - «қазақстан-2050» стратегиясы аясында үшінші индустриалды революцияны жүзеге асырудың құралы» атты III халықаралық ғылыми – практикалық конференция материалдары. – Астана, 2016. – Б. 121–128

87 А.Е. Сағымбаева, А.Е. Жаксылыков. Білімді компьютерлік бақылауда көпдеңгейлі ақпараттық модельді қолдану жайлы // «Математикалық модельдеу мен ақпараттық технологиялар білімде және ғылымда» атты VII Халықаралық ғылыми-әдістемелік конференция материалдары. – Алматы, 2015. – Б. 641–644

88 Сағымбаева А.Е., Жақсылықов Ә.Е. Білімді бақылауды компьютерлік көпдеңгейлі ақпараттық модель негізінде іске асырудың қажеттілігі // Абай атындағы ҚазҰПУ Хабаршысы. «Физика-математика ғылымдары» сериясы. – 2015. – №3(51). – Б. 259–263.

89 Сағымбаева А.Е., Жақсылықов Ә.Е. Оқыту нәтижелерін бақылаудың көпдеңгейлі семантикалық ақпараттық моделінің компьютерлік жүйесі // Қазақстан Педагогикалық Ғылымдар Академиясының Хабаршысы. – 2017. – №3. – Б. 64–71.

90 Sagimbaeva A.E., Abildinova G.M., Zhaxylykov A.E. Issue of the Application of the Multi-Level Semantic Information System for Learning Outcomes Monitoring // ПМУ Хабаршысы. Педагогикалық серия. – 2017. – №4. – P. 302–315.

91 Сағымбаева А.Е., Жақсылықов Ә.Е. Білім алушылардың оқу материалын меңгеру деңгейлеріне салыстырмалы талдау // Абай атындағы ҚазҰПУ Хабаршысы. «Физика-математика ғылымдары» сериясы. – 2017. – № 3 (59). – Б. 257-261.

92 А.Е. Сағымбаева, А.Е. Жаксылыков. Білімді бақылаудың көп деңгейлі ақпараттық моделін компьютерде іске асыру мәселесі // «Заманауи ғылым және білімнің өзекті мәселелері мен инновация тенденциялары» атты халықаралық ғылыми-тәжірибелік конференция материалдары. – Түркістан, 2017. – Б. 510–514

93 Liang J., Hare R., Chang T., Xu F., Tang Y., Wang F.-Y., Peng S. Student modeling and analysis in adaptive instructional systems // IEEE Access. – 2022. – Vol. 10. – P. 59359-59372. – DOI: 10.1109/ACCESS.2022.3178744.

94 Eglington L.G., Pavlik P.I. How to optimize student learning using student models that adapt rapidly to individual differences // International Journal of Artificial Intelligence in Education. – 2023. – Vol. 33. – P. 497-518. – DOI: 10.1007/s40593-022-00296-0.

95 VanLehn K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems // Educational Psychologist. – 2011. – Vol. 46, No. 4. – P. 197-221. – DOI: 10.1080/00461520.2011.611369.

96 Найманова Д.С., Ткач Г.М., Даутова А.З., Жаксылыков А.Е. IT мамандығы студенттерінің оқу үлгерімі мен мансаптық траекториясын интеллектуалды болжау моделі // Абай атындағы ҚазҰПУ Хабаршысы. «Физика-математика ғылымдары» сериясы. – 2025. – № 3 (91). – Б. 275-286. – DOI: 10.51889/2959-5894.2025.91.3.025.

97 Nurbekova Zh., Naimanova D., Tkach G., Dautova A., Zhaxylykov A., Rakhimbayev A. Intelligent system for analyzing the career achievements of IT specialists: development and implementation of an AI-based software solution // 2025 10th International Conference on Computer Science and Engineering (UBMK). – Istanbul: IEEE, 2025. – P. 1270-1274. – DOI: 10.1109/UBMK67458.2025.11206957.

98 Codeforces. Contest rules: judging solutions. – URL: https://codeforces.com/blog/entry/4088 (қаралған күні: 09.09.2026).

99 Codeforces. API help: Submission object. – URL: https://codeforces.com/apiHelp/objects#Submission (қаралған күні: 09.09.2026).

100 Orvalho P., Janota M., Manquinho V. GitSEED: A Git-backed Automated Assessment Tool for Software Engineering and Programming Education // SIGCSE Virtual 2024: Proceedings of the 2024 ACM Virtual Global Computing Education Conference. – Association for Computing Machinery, 2024. – P. 165–171. – DOI: 10.1145/3649165.3690106.

101 HackerRank. Test cases in coding questions. – URL: https://support.hackerrank.com/articles/3245197419-test-cases-in-coding-questions (қаралған күні: 09.09.2026).

102 HackerRank. Viewing a candidate’s detailed test report. – URL: https://support.hackerrank.com/articles/8263794320-viewing-a-candidate%27s-detailed-test-report (қаралған күні: 09.09.2026).

103 Virtual Programming Lab for Moodle (VPL). Documentation, version 4.4.2: Introduction. – URL: https://vpl.dis.ulpgc.es/documentation/vpl-4.4.2/introduction.html (қаралған күні: 09.09.2026).

104 Virtual Programming Lab for Moodle (VPL). Documentation, version 4.4.2: Automated program assessment using BIOTES. – URL: https://vpl.dis.ulpgc.es/documentation/vpl-4.4.2/biotes.html (қаралған күні: 09.09.2026).

105 Xiang C., Wang Y., Zhou Q., Yu Z. Graph semantic similarity-based automatic assessment for programming exercises // Scientific Reports. – 2024. – Vol. 14, No. 1. – Article 10530. – DOI: 10.1038/s41598-024-61219-8.

106 Theeb A., Abdullah N.A., Fenjan A., Keikhosrokiani P. A lightweight intelligent feedback system for student code submissions using large language models // Discover Education. – 2026. – Vol. 5. – Article 371. – DOI: 10.1007/s44217-026-01400-5.

107 Shute V.J. Focus on formative feedback // Review of Educational Research. – 2008. – Vol. 78, No. 1. – P. 153-189. – DOI: 10.3102/0034654307313795.

108 Загвязинский В.И. Теория обучения: современная интерпретация: учебное пособие для вузов. – М.: Академия, 2001. – 192 с.

109 Бабаев С.Б., Оңалбек Ж.К. Жалпы педагогика: оқулық. — Алматы: Нұр-пресс, 2005. — 228 б.

110 Lister R. et al. A Multi-National Study of Reading and Tracing Skills in Novice Programmers // ACM SIGCSE Bulletin. — 2004. — Vol. 36, № 4. — P. 119–150.

111 Wu Y., Zhu H., Wang C., Song F., Zhu H., Chen Y., Zheng Q., Tian F. Programming knowledge tracing based on heterogeneous graph representation // Knowledge-Based Systems. – 2024. – Vol. 300. – Article 112161. – DOI: 10.1016/j.knosys.2024.112161.

112 Nurbekova Z., Sagimbayeva A., Aimicheva G., Zhaksylykov A., Nazyrova A., Artykbayev N. Improving Programming Learning Outcomes for AI-Driven Students using a Digital Assessment Environment // International Journal of Engineering Education. – 2026. – Vol. 42, No. 4. – P. 1116–1130.

113 Ferrara S., Qunbar S. Validity Arguments for AI-Based Automated Scores: Essay Scoring as an Illustration // Journal of Educational Measurement. – 2022. – Vol. 59, No. 3. – P. 288–313. https://doi.org/10.1111/jedm.12333Digital Object Identifier (DOI)

114 Khan W., Topham L., Jones N., Atherton P., Al-Shabandar R., Kolivand H., et al. Auto-assessment of assessment: A human-in-the-loop AI framework addressing policy gaps in academic assessment // PLOS ONE. – 2026. – Vol. 21, No. 4. – e0346815. – DOI: 10.1371/journal.pone.0346815.

115 Авдарсоль С. Информатикадан критериалдық тәсіл негізінде оқушылардың функционалдық сауаттылығын бағалау жүйесін дамыту: философия докторы (PhD) дәрежесін алу үшін дайындалған диссертация: 6D011100 – Информатика. – Алматы: Абай атындағы Қазақ ұлттық педагогикалық университеті, 2022. – 126 б.

116 Biggs J. Enhancing teaching through constructive alignment // Higher Education. – 1996. – Vol. 32, No. 3. – P. 347-364. – DOI: 10.1007/BF00138871.

117 Baartman L.K.J., Bastiaens T.J., Kirschner P.A., van der Vleuten C.P.M. Evaluating assessment quality in competence-based education: a qualitative comparison of two frameworks // Educational Research Review. – 2007. – Vol. 2, No. 2. – P. 114-129. – DOI: 10.1016/j.edurev.2007.06.001.

118 Chen H.-M., Nguyen B.-A., Dow C.-R. Code-quality evaluation scheme for assessment of student contributions to programming projects // Journal of Systems and Software. – 2022. – Vol. 188. – Article 111273. – DOI: 10.1016/j.jss.2022.111273

119 Zh. Nurbekova, A. Sagimbaeva, A. Zhaxylykov, N. Artykbayev. Programming Knowledge Assessment in the GenAI Era: Research, Challenges, and Methodology // Fourth scientific-practical conference with international participation digital transformation of education - issues and solutions, Bulgaria, Ruse, 23–24 April 2026. – 2026. – URL: https://www.conf-dte.bg/proceeding-abstracts.html#abstracts-2025

120 А.Е. Сагимбаева, А.Е. Жаксылыков. Вопросы моделирование ситуации в компьютерном тестировании // Материалы всероссийской молодежной конференции с международным участием «Инновационные средства и методы обучения в условиях ИКТ». – Красноярск (РФ), 2015. – С. 142–147

121 Ш.Т. Шекербекова, А.Е. Сағымбаева, А.Е. Жаксылыков. Программалау курсынан білімді бағалау критерийлері // «Математикалық модельдеу мен ақпараттық технологиялар білімде және ғылымда» атты X халықаралық ғылыми-әдістемелік конференция материалдары. – Алматы, 2025. – Б. 914–918. – ISBN 978-601-311-284-8

122 Fuller U., Johnson C.G., Ahoniemi T., Cukierman D., Hernán-Losada I., Jackova J., Lahtinen E., Lewis T.L., Thompson D.M., Riedesel C., Thompson E. Developing a computer science-specific learning taxonomy // ACM SIGCSE Bulletin. – 2007. – Vol. 39, No. 4. – P. 152-170. – DOI: 10.1145/1345375.1345438.

123 А.Е. Сағымбаева, А.Б. Жамкеева, М.Н. Кемелбек. Программалаудан студенттердің білімін бағалау критерийлері мен дескрипторларын жасау мәселелері // Х.Досмұхамедов атындағы Атырау университетінің Хабаршысы. – Атырау, – 2025. №77(2). –Б. 131-140. https://doi.org/10.47649/vau.25.v77.i2.11

124 Ш.Т. Шекербекова, А.Е. Сағымбаева, А.Е. Жаксылыков, Н.Е. Артықбаев, А.К. Нагметов. «Программалаудан бағалау критерийлерін жасаудағы GenAI технологиясының мүмкіндіктері», 351–360 б., DOI: 10.51889/2959-5894.2025.90.2.031.

125 Сағымбаева А.Е., Жамкеева А.Б., Жаксылыков А.Е., Қожан А.Е. Студенттердің программалау есептерін шешу көрсеткішін бағалау моделі // Абай атындағы ҚазҰПУ-нің Хабаршысы. «Физика-математика ғылымдары» сериясы. – 2026. – №1 (93). – Б. 300–310. – DOI: 10.51889/2959-5894.2026.93.1.027

126 Sheard J., Simon, Carbone A., Chinn D., Clear T., Corney M., D'Souza D., Fenwick J., Harland J., Laakso M.-J., Teague D. How difficult are exams? A framework for assessing the complexity of introductory programming exams // Proceedings of the Fifteenth Australasian Computing Education Conference (ACE 2013). – Sydney: Australian Computer Society Inc., 2013. – Vol. 136. – P. 145–154.

127 Masapanta-Carrión S., Velázquez-Iturbide J.Á. Evaluating instructors' classification of programming exercises using the revised Bloom's taxonomy // Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2019). – ACM, 2019. – P. 541–547. DOI: 10.1145/3304221.3319748

128 Zhang Y., Paquette L., Pinto J.D., Fan A.X. Utilizing programming traces to explore and model the dimensions of novices' code-writing skill // Computer Applications in Engineering Education. – 2023. – Vol. 31, No. 4. – P. 1041-1058. – DOI: 10.1002/cae.22622.

129 Sun X., Liu Q., Zhang K., Shen S., Zhuang Y., Guo Y. LGS-KT: integrating logical and grammatical skills for effective programming knowledge tracing // Neural Networks. – 2025. – Vol. 185. – Article 107164. – DOI: 10.1016/j.neunet.2025.107164.

130 Tusquellas N., López-Villanueva D., Palau R., Santiago R. Educational Conceptual Model Design Research Methodology // UTE Teaching & Technology (Universitas Tarraconensis). – 2025. – № 2. – e4103. DOI: 10.17345/ute.2025.4103.

131 Poličar P.G., Špendl M., Curk T., Zupan B. Automated assignment grading with large language models: insights from a bioinformatics course // Bioinformatics. – 2025. – Vol. 41, Suppl. 1. – P. i21-i29. – DOI: 10.1093/bioinformatics/btaf196.

132 Beck K. Test-driven development: by example. – Boston: Addison-Wesley, 2003. – 220 p.

133 Hattie J., Timperley H. The power of feedback // Review of Educational Research. – 2007. – Vol. 77, No. 1. – P. 81-112. – DOI: 10.3102/003465430298487.

134 Grattafiori A., Dubey A., Jauhri A., Pandey A. et al. The Llama 3 herd of models // arXiv preprint arXiv:2407.21783. – 2024. – DOI: 10.48550/arXiv.2407.21783.

ҚОСЫМША А

Студенттерге арналған сауалнама

Мақсаты: программалаудан студенттердің білімін бақылаудың бастапқы жағдайын, қолданылатын бақылау тапсырмаларын және студенттердің программалау тапсырмаларын орындауда GenAI құралдарын пайдалану тәжірибесін анықтау.

Нұсқаулық: Сұрақтарға өз тәжірибеңізге сәйкес жауап беріңіз. Сауалнама нәтижелері ғылыми-зерттеу мақсатында жинақталған түрде пайдаланылады.

1. Жоғары оқу орнының атауы?

Абай атындағы Қазақ ұлттық педагогикалық университеті
Торайғыров университеті
Қазақ ұлттық қыздар педагогикалық университеті

2. Сіздің мамандығыңыз?

«6В01507 – Информатика»
«6В01509 – Информатика және робототехника»
«6В06102 - Ақпараттық жүйелер»
«6В01506 - Физика-информатика»
6В01506 — Информатика
6В01514 — Информатика және робототехника
6В06102 — Ақпараттық жүйелер

3. Нешінші курс студентісіз?

1 курс
2 курс
3 курс
4 курс

4. Программалаудан біліміңізді бақылауда қандай тапсырмалар қолданылады? (бірнеше жауапты таңдауға болады)

Тест тапсырмалары; жобалық жұмыстар; практикалық тапсырмалар; кодты тексеру; ауызша сауалнама; интербелсенді тапсырмалар; басқа: ____.

5. Программалаудан білімді бақылауда қолданылатын тапсырмаларға қаншалықты қанағаттанасыз?

1 – мүлдем қанағаттанбаймын; 2 – қанағаттану деңгейі төмен; 3 – орташа; 4 – қанағаттанамын; 5 – толық қанағаттанамын.

6. Программалау тапсырмаларын орындау барысында GenAI құралдарын (ChatGPT, Gemini, GitHub Copilot және т.б.) пайдаланасыз ба?

Иә; Жоқ.

7. GenAI құралдарының программалау дағдыларыңызға әсерін қалай бағалайсыз?

Айқын әсер етті; әсері байқалды, бірақ шамалы; айтарлықтай әсер еткен жоқ; әсері болмады

8. GenAI құралдарының программалау бойынша бақылау тапсырмаларын орындаудағы көмегін қалай бағалайсыз?

1 – мүлдем көмектеспейді; 2 – көмегі аз; 3 – көмегі орташа; 4 – көмегі жоғары; 5 – өте жақсы көмектеседі.

ҚОСЫМША Ә

Оқытушыларға арналған сауалнама

Оқытушылардың программалаудан білімін бақылауда бастапқы жағдайын диагностикалау

1. Жоғары оқу орнының атауы?

_____

2. Білім беру бағдарламасының атауы, өзіңіздің ғылыми дәрежеңіз

_____

3. Программалауға байланысты қандай пәндерді оқытасыз

_____

4. Программалаудан студенттердің білімін бақылауда қандай әдістерді қолданасыз?

- Тест тапсырмаларын
- Жобалық жұмыстарды
- Практикалық тапсырмаларды
- Кодты тексеруді
- Ауызша сауалнамаларды
- Интербелсенді тапсырмаларды
- Басқа _____

5. Программалаудан студенттердің білімін бақылауды қалай ұйымдастырасыз?

- Топтық талқылаулар
- Онлайн программалау платформалары
- Кодтау жарыстары
- Өзін-өзі бағалау
- Тәлімгерлік және жұптық программалау
- Жұмыстардың портфолиосы
- Басқа _____

6. Программалаудан дәріс материалдарын дайындауда GenAI құралдарын пайдаланасыз ба? (мысалы, ChatGPT, Gemini және т.б.)

- Ия
- Жоқ

7. Программалаудан практикалық тапсырмаларды дайындау үшін GenAI құралдарын (мысалы, ChatGPT, Gemini, GitHub Copilot, Tabnine, Codeium және т.б.) қолданасыз ба??

_____

8. Өз тәжірибеңізде нақты қандай GenAI құралдарын пайдаланасыз?

9. GenAI құралдарының кәсіби құзыреттілігіңізді дамытуға және практикалық қызметіңізге әсерін қалай бағалайсыз?

- Толықтай әсері болды
- Әсері байқалды, бірақ шамалы
- Айтарлықтай әсер еткен жоқ
- Әсері болмады

10. GenAI-дің программалаудан оқу материалдарын дайындаудағы үлесін қалай бағалайсыз? (1 — Мүлдем көмектеспейді, 5 — Өте жақсы көмектеседі)

- 1
- 2
- 3
- 4
- 5

ҚОСЫМША Б

Студенттерге, профессор-оқытушыларға және IT-сарапшыларға арналған сауалнама

ЖОО-да жобалық-бағдарланған оқыту үдерісіне GenAI-дің әсерін программалау курсы мысалында зерттеу

Құрметті респонденттер!

Сіздерді программалау курсы мысалында жасанды интеллектінің жоғары білім берудегі жобалық-бағдарланған оқыту үдерісіне әсерін зерттеуге бағытталған сауалнамаға қатысуға шақырамыз.

Сіздің қосқан үлесіңіз студенттік жобаларды сүйемелдеу әдістемесін оңтайландыруға және жобалаудың барлық кезеңдеріндегі бағалау сапасын арттыруға мүмкіндік береді.

Сауалнаманы толтыруға шамамен 5–10 минут уақытыңыз кетеді. Бөлген уақытыңыз бен қатысуыңыз үшін шынайы алғысымызды білдіреміз. Алынған деректер статистикалық өңдеу мақсатында жалпыланған түрде пайдаланылатын болады.

Тәуекелдер мен артықшылықтар: бұл сауалнамаға қатысу қандай да бір тәуекел немесе қолайсыздық туғызбайды деп есептейміз. Сіздің атыңыз көрсетілмейді және ешбір жағдайда берген жауаптарыңызбен байланыстырылмайды. Сауалнамаға қатысудан тікелей пайда алмауыңыз мүмкін болғанымен, оған қатысу Сіздің алдағы ғылыми-зерттеу және кәсіби қызметіңізге оң ықпал етеді деп үміттенеміз.

Қатысушылардың құқықтары: сауалнамаға қатысуыңыз толығымен ерікті. Егер қандай да бір уақытта сауалнамаға қатысуды жалғастырғыңыз келмейді деп шешсеңіз, ешқандай салдарсыз өз келісіміңізді кері қайтарып алуға құқығыңыз бар. Сауалнаманы толтырудың кез келген кезеңінде қатысудан бас тартуыңыз Сіздің кәсіби қызметіңізге ешқандай әсер етпейді. Сауалнама сұрақтарына жауап беру арқылы Сіз ұсынылған ақпаратпен мұқият танысқаныңызды және зерттеуге қатысуға өз келісіміңізді бергеніңізді растайсыз.

1. Рөлі:

Білім алушы (бакалавриат – магистратура – докторантура)

Профессор-оқытушылар құрамы

IT-сарапшылар

Білім алушы туралы мәліметтер (бакалавриат – магистратура – докторантура)

2. Сіз білім алатын ЖОО-ның түрі мен мәртебесін көрсетіңіз:

Зерттеу университеті

Педагогикалық университет

Техникалық университет

Жекеменшік университет

Мемлекеттік университет

Ұлттық университет

Басқа _____

3. Қазіргі білім алу деңгейіңізді таңдаңыз:
Бакалавриат
Магистратура
Докторантура
Басқа _____

4. Қазіргі оқу курсыңызды таңдаңыз:
1-курс
2-курс
3-курс
4-курс
5-курс

5. Программалау бойынша олимпиадаларға, хакатондарға немесе басқа да жарыстарға қатыстыңыз ба?
Иә, халықаралық деңгейде
Иә, республикалық деңгейде
Иә, университет деңгейінде
Жоқ
Басқа:

Профессор-оқытушылар құрамы туралы мәліметтер:
6. ЖОО-ның түрі мен мәртебесін көрсетіңіз:
Зерттеу университеті
Педагогикалық университет
Техникалық университет
Жекеменшік университет
Мемлекеттік университет
Ұлттық университет
Басқа_____

7. Ғылыми/академиялық дәрежеңізді көрсетіңіз:
Ғылым докторы
Ғылым кандидаты
PhD
Магистр
Басқа_____

8. Академиялық лауазымыңызды көрсетіңіз:
Оқытушы
Аға оқытушы
Қауымдастырылған профессор (доцент)
Профессор
Зерттеуші/ғылыми қызметкер

Басқа_____

9. Жұмыс өтіліңізді көрсетіңіз (жылмен):
1 жылдан аз
1–3 жыл
3–5 жыл
5–10 жыл
10–20 жыл
20 жылдан астам

IT-сарапшылар туралы мәліметтер:
10. IT саласындағы кәсіби деңгейіңізді таңдаңыз
Junior
Middle
Senior
Lead / Architect
Басқа _____

11. IT саласындағы лауазымыңызды көрсетіңіз
Software Developer / Engineer (программалық жасақтаманы әзірлеуші / инженер)
QA Engineer / Tester (сапаны қамтамасыз ету инженері / тестілеуші)
Data Analyst / Data Scientist (деректер талдаушысы / деректер ғылымы маманы)
DevOps / System Engineer (DevOps / жүйелік инженер)
Team Lead / Tech Lead (команда жетекшісі / техникалық жетекші)
Project / Product Manager (жоба / өнім менеджері)
AI / ML Engineer (жасанды интеллект / машиналық оқыту инженері)
Басқа _____

12. IT саласындағы жұмыс өтіліңізді көрсетіңіз (жылмен)
1 жылдан аз
1–3 жыл
3–5 жыл
5–10 жыл
10–20 жыл
20 жылдан астам

Профессор-оқытушылар құрамына қойылған сұрақ
13. IT пәндерін оқыту тәжірибеңізге сүйене отырып, программалау бойынша білім, білік және дағдыларды бағалау кезінде төмендегі компоненттердің үлес салмағын (%) бөліп көрсетіңіз.
Үш көрсеткіштің қосындысы 100%-ға тең болуы тиіс.

Кесте Б.1 – Бағалау компоненттерінің үлес салмағы

Бағалау компоненттері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Кодтың функционалдық дұрыстығы мен талаптарға сәйкестігі	○	○	○	○	○	○	○	○	○	○	○
ЖИ құралдарын және промпт-инжинирингті меңгеру деңгейі	○	○	○	○	○	○	○	○	○	○	○
Техникалық қорғау және шешімді негіздеу (алгоритм логикасын түсіну және сұрақтарға жауап беру)	○	○	○	○	○	○	○	○	○	○	○

Профессор-оқытушылар құрамына қойылған сұрақ

14. «ЖИ құралдарын және промпт-инжинирингті меңгеру» блогы бойынша балдарды (үлес салмағын) бөліңіз

Кесте Б.2 – «ЖИ құралдарын және промпт-инжинирингті меңгеру» блогы бойынша балдардың үлес салмағы

ЖИ-ді пайдалану деңгейі	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Copy-Paste: талдаусыз, үстірт пайдалану	○	○	○	○	○	○	○	○	○	○	○
Integrator: тапсырманы құрамдас бөліктерге жіктеу және алынған шешімдерді өз бетінше біріктіру	○	○	○	○	○	○	○	○	○	○	○
Mentor: шешімді жобалау және терең деңгейдегі промпт-инжиниринг	○	○	○	○	○	○	○	○	○	○	○

Профессор-оқытушылар құрамына қойылған сұрақ

15. Студенттің түсіндіру сапасын ескере отырып, оған ең жоғары балдың (0–100%) қанша пайызын беру керек деп есептейсіз?

Кесте Б.3 – Студенттің түсіндіру сапасы үшін берілетін балдың үлесі

Түсіндіру деңгейі	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Базалық: синтаксисті түсіндіреді, бірақ программаның жалпы логикасын түсіндіре алмайды	○	○	○	○	○	○	○	○	○	○	○
Орта: программаның логикасы мен таңдалған шешімді түсіндіреді, бірақ есеп шарты өзгерген жағдайда қиналады	○	○	○	○	○	○	○	○	○	○	○
Жоғары: кодты терең түсінеді, шешім таңдауын негіздейді, оны өзгерген жағдайға бейімдей алады және балама шешімдерді салыстырады	○	○	○	○	○	○	○	○	○	○	○

16. 1-кезең – Ұйымдастыру (инициация)

Жобаның бастау кезеңіндегі критерийлердің маңыздылығын бағалаңыз (мәселе мен мақсатты тұжырымдау, күтілетін нәтижені, шектеулерді және шешімнің бастапқы тұжырымдамасын анықтау).

Кесте Б.4 – Жобаның бастау (инициация) кезеңіндегі критерийлердің маңыздылығы

Бағалау критерийлері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Мәселе мен мақсатты қою логикасы (мақсат пен күтілетін нәтиженің мәселеден қаншалықты қисынды туындайтыны)	○	○	○	○	○	○	○	○	○	○	○
Бастапқы тұжырымдаманың креативтілігі (шынайылықты сақтай отырып, шешім идеясының бірегейлігі)	○	○	○	○	○	○	○	○	○	○	○
Жоба сипаттамасының құрылымдылығы (жобаны іске асыру кезеңдерінің реттілігіне сипаттаманың сәйкестігі)	○	○	○	○	○	○	○	○	○	○	○
Таңдалған бағыттың тиімділігі (тұжырымдаманың қолда бар ресурстарды ұтымды пайдалана отырып, күтілетін нәтижеге жеткізу мүмкіндігі)	○	○	○	○	○	○	○	○	○	○	○
Бастапқы талаптар мен шектеулердің дұрыстығы (есеп шарттарының сәйкестігі, табыстылық критерийлерінің дұрыстығы)	○	○	○	○	○	○	○	○	○	○	○
Бастамашылық кезеңінде GenAI-ді пайдалану (тапсырманы нақтылау, идеяларды генерациялау және талаптарды формализациялау үшін ЖИ-ді саналы түрде қолдану)	○	○	○	○	○	○	○	○	○	○	○

17. 2-кезең – Жоспарлау

Жобаны жоспарлау кезеңіндегі критерийлердің маңыздылығын бағалаңыз (міндеттерді тұжырымдау, тәсілдерді таңдау, ресурстарды бөлу және табыстылық критерийлерін анықтау).

Кесте Б.5 – Жобаны жоспарлау кезеңіндегі критерийлердің маңыздылығы

Бағалау критерийлері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Шешімді жобалау логикасы (таңдалған стратегия мен алгоритмдердің негізділігі)	○	○	○	○	○	○	○	○	○	○	○
Тұжырымдама мен идеялардың креативтілігі (есепті шешу тәсілінің бірегейлігі)	○	○	○	○	○	○	○	○	○	○	○
Жоба жоспарының құрылымдылығы (кезеңдердің, міндеттер мен рөлдердің нақтылығы)	○	○	○	○	○	○	○	○	○	○	○
Жоспарлаудың тиімділігі (мерзімдер мен ресурстардың шынайылығы)	○	○	○	○	○	○	○	○	○	○	○
Есептің дұрыс қойылуы (жоба талаптары мен мақсаттарына сәйкестігі)	○	○	○	○	○	○	○	○	○	○	○
Жоспарлау кезінде GenAI-ді пайдалану (талдау, идеяларды қалыптастыру және шешім құрылымын әзірлеу үшін ЖИ-ді қолдану)	○	○	○	○	○	○	○	○	○	○	○

18. 3-кезең – Іске асыру

Жобаны іске асыру кезеңіндегі критерийлердің маңыздылығын бағалаңыз (программалық кодты жазу, функционалды іске асыру, өзгерістер енгізу және жұмыс істейтін нәтижеге қол жеткізу).

Кесте Б.6 – Жобаны іске асыру кезеңіндегі критерийлердің маңыздылығы

Бағалау критерийлері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Шешімді іске асыру логикасы (алгоритмдер мен таңдалған тәсілдің негізділігі)	○	○	○	○	○	○	○	○	○	○	○
Программалық шешімнің креативтілігі (іске асырудың стандартты емес және тиімді тәсілдері)	○	○	○	○	○	○	○	○	○	○	○
Кодтың құрылымы мен оқылымдылығы (модульділігі, түсініктілігі және ұйымдастырылуы)	○	○	○	○	○	○	○	○	○	○	○
Іске асыру тиімділігі (оңтайлылығы, өнімділігі және ресурстарды тиімді пайдалануы)	○	○	○	○	○	○	○	○	○	○	○
Программаның жұмыс істеу дұрыстығы (қателердің болмауы және тапсырма талаптарының орындалуы)	○	○	○	○	○	○	○	○	○	○	○
Әзірлеу барысында GenAI-ді пайдалану (ЖИ-ді саналы түрде қолдану, ұсынылған шешімдерді түсіну және бейімдеу)	○	○	○	○	○	○	○	○	○	○	○

19. 4-кезең – Мониторинг және бақылау

Жобаны мониторингтеу және бақылау кезеңіндегі критерийлердің маңыздылығын бағалаңыз (шешімнің сапасын тексеру, қателерді анықтау және түзету, талаптар мен көрсеткіштерге сәйкестігін бақылау).

Кесте Б.7 – Жобаны мониторингтеу және бақылау кезеңіндегі критерийлердің маңыздылығы

Бағалау критерийлері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Логикалық талдау және қателерді анықтау (қателер мен сәйкессіздіктердің себептерін анықтай білу)	○	○	○	○	○	○	○	○	○	○	○
Шешім іздеудегі креативтілік (қателерді түзету мен шешімді жетілдірудің стандартты емес тәсілдері)	○	○	○	○	○	○	○	○	○	○	○
Тексерудің құрылымдылығы (тестілеу мен бақылаудың жүйелілігі)	○	○	○	○	○	○	○	○	○	○	○
Түзетулердің тиімділігі (шешімді жетілдірудің оңтайлылығы мен жылдамдығы)	○	○	○	○	○	○	○	○	○	○	○
Нәтиженің дұрыстығы мен сенімділігі (талаптарға сәйкестігі және қателердің болмауы)	○	○	○	○	○	○	○	○	○	○	○

Б.7-кестенің жалғасы

Бағалау критерийлері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Тестілеу және жөндеу барысында GenAI-ді пайдалану (талдау және қателерді түзету үшін ЖИ-ді саналы түрде қолдану)	○	○	○	○	○	○	○	○	○	○	○

20. 5-кезең – Жобаны қорғау

Жобаны қорғау кезеңінде әрбір критерийдің қорытынды бағалауға қаншалықты ықпал етуі керектігін бағалаңыз (шешімді таныстыру, сұрақтарға жауап беру, қорытындылар мен ұсыныстарды негіздеу).

Кесте Б.8 – Жобаны қорғау кезеңіндегі критерийлердің маңыздылығы

Бағалау критерийлері	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
Шешімді логикалық негіздеу (аргументтер мен себеп-салдарлық байланыстарды негізді көрсету)	○	○	○	○	○	○	○	○	○	○	○
Тәсілдің креативтілігі (дұрыстықты сақтай отырып, стандартты емес идеяларды ұсыну)	○	○	○	○	○	○	○	○	○	○	○
Презентацияның/баяндаманың құрылымдылығы (мазмұнның анық логикасы мен баяндау реттілігі)	○	○	○	○	○	○	○	○	○	○	○
Ұсынылған шешімнің тиімділігі (орындалу уақыты мен жадты пайдалану, оңтайлылық және масштабталу мүмкіндігі)	○	○	○	○	○	○	○	○	○	○	○
Нәтиженің дұрыстығы мен сенімділігі (дұрыс нәтижелер, қателер мен шеттік жағдайларды өңдеу)	○	○	○	○	○	○	○	○	○	○	○
GenAI-ді саналы және ашық пайдалану (ЖИ көмегімен не және қандай мақсатта орындалғанын түсіндіре білу)	○	○	○	○	○	○	○	○	○	○	○

ҚОСЫМША В

Сараптамалық-негіздеуші кезеңде жасалған жобалық программалау тапсырмалары

1. Жобалық тапсырма 1. «Қатысу деңгейін бақылау жүйесі (Computer Vision)»

Жобаның мақсаты. Веб-камера арқылы студенттің экран алдында болуын және тест тапсыру барысында назар (концентрация) деңгейін автоматты түрде бақылап, талдайтын веб-қосымшаны әзірлеу.

Жобаның сипаттамасы. Жоба компьютерлік көру технологияларын пайдалана отырып, студенттің онлайн тестілеу немесе оқу сессиясы кезіндегі белсенділігін бағалауға бағытталған. Жүйе нақты уақыт режимінде бетті анықтап, экран алдындағы болу ұзақтығын есептейді және қорытынды аналитика ұсынады.

Негізгі функциялары (MVP):

- нақты уақыт режимінде адамның бетін анықтау (OpenCV, Haar Cascades немесе MediaPipe);
- қолданушы кадрдан шығып кеткен сәттерді тіркеу;
- экранға қарау уақытын автоматты түрде есептеу;
- сессия бойынша статистиканы сақтау (уақыт, белсенділік);
- сессия соңында қатысу деңгейінің қорытынды графигін генерациялау.
- Қолдану саласы:
- онлайн тестілеу жүйелері;
- қашықтан оқыту платформалары;
- академиялық адалдықты бақылау;
- оқу үдерісінің аналитикасы.

Қолданылатын технологиялар: Python, OpenCV, Flask, SQLite.

Күтілетін нәтиже. Жүйе студенттің оқу немесе тест сессиясындағы қатысу деңгейін пайыздық көрсеткіштер мен визуалды графиктер арқылы көрсетеді.

2. Жобалық тапсырма 2. «Студенттің уақытын ақылды жоспарлау» веб-қосымшасы

Жобаның мақсаты. Студенттердің оқу тапсырмаларын, дедлайндарын және уақыт ресурстарын тиімді басқаруға арналған веб-қосымшаны әзірлеу. Эксперименттік жоспар

Жобаның сипаттамасы. Жоба студенттердің күнделікті оқу жүктемесін жүйелеуге, тапсырмаларды уақытында орындауға және уақытты тиімді бөлуге көмектесетін веб-негізделген жоспарлау құралын құруды көздейді.

Негізгі функциялары (MVP):

- оқу тапсырмаларын қосу, өңдеу және жою;
- әр тапсырмаға дедлайн мен басымдық деңгейін тағайындау;
- апталық немесе айлық оқу жүктемесін визуализациялау;
- уақытты тиімді бөлу бойынша автоматты ұсыныстар беру;
- тапсырмалардың орындалу мәртебесін бақылау.
- Қолдану саласы:
- жеке оқу үдерісін жоспарлау;

- университеттер мен колледждер;
- онлайн және офлайн оқу формалары.

Қолданылатын технологиялар: Python + Flask, HTML/CSS, SQLite.

Күтілетін нәтиже. Қосымша студентке жеке оқу жоспарын құруға мүмкіндік беріп, оқу уақытының тиімділігін арттыруға бағытталады.

Жобалық тапсырмаларды орындауды ұйымдастыру

Жобалық тапсырмаларды орындау барысында студенттер 3–4 адамнан тұратын шағын топтарға біріктіріледі. Әр топта жоба архитекторы (Software Architect), әзірлеуші (Developer) және жоба менеджері (Project Manager) рөлдері бөлінеді.

Жоба архитекторы (Software Architect) программалық жүйенің жалпы архитектурасы мен техникалық шешімдерін жобалауға жауап береді. Оның негізгі функцияларына бизнес-талаптарды техникалық мүмкіндіктермен сәйкестендіру, архитектуралық тәсілді таңдау, жобаның модульдері мен компоненттерінің өзара байланысын анықтау, қолданылатын құралдарды белгілеу, API мен деректер форматтарын анықтау және жобаның бастапқы жоспарға сәйкестігін бақылау жатады.

Әзірлеуші (Developer) жобаның функционалдығын бекітілген архитектураға сәйкес программалық түрде іске асырады. Оның функцияларына тиімді әрі оқылатын код жазу, негізгі алгоритмдер мен функцияларды құрастыру, мәліметтер қорын жобалау және басқару, кодтың дұрыс жұмыс істеуін тексеру және командаластардың кодының сапасын бақылау кіреді.

Жоба менеджері (Project Manager) командалық жұмысты ұйымдастыру мен үйлестіруге жауап береді. Ол жобаны жеке тапсырмаларға декомпозициялайды, рөлдер мен жауапкершілікті бөледі, мерзімдер мен бақылау нүктелерін жоспарлайды, тапсырмалардың орындалуын бақылайды, команда ішіндегі коммуникацияны ұйымдастырады және тәуекелдерді басқарады.

Жобалық тапсырмалар GitHub веб-сервисі негізінде ұйымдастырылады. Бұл орта студенттердің бірлесіп жұмыс жасауына және жобаның орындалу үдерісін бақылауға мүмкіндік береді. Қосымша көмекші құрал ретінде ChatGPT, Copilot және Gemini сияқты GenAI жүйелерін қолдануға рұқсат етіледі.

GenAI құралдары жобаны орындауда көмекші ресурс ретінде қарастырылады. Студенттер оларды жобаны әзірлеудің кез келген кезеңінде пайдалана алады, алайда дайын шешімдерді немесе толық программалық кодты механикалық түрде көшіруге жол берілмейді. GenAI қолданылған әрбір жағдайда оның қай кезеңде, қандай мақсатта және не үшін пайдаланылғаны тіркеледі.

Жобалық тапсырмаларды орындау кезеңдері

Жобалық тапсырмаларды орындау үдерісі бес апталық кезеңдік цикл бойынша ұйымдастырылады. Әр кезең студенттің программалау әрекетінің белгілі бір бөлігін қамтиды.

1-кезең – Analysis (талдау). Жобалық тапсырманың талаптарын талдау, жобаның мақсатын нақтылау және күтілетін нәтижелерді анықтау.

2-кезең – Planning (жоспарлау). Шешім жоспарын құру, жобаның архитектурасын және негізгі модульдерін жобалау.

3-кезең – Implementation (іске асыру). Жобаның жоспарланған функционалдығын программалық түрде жүзеге асыру.

4-кезең – Control (бақылау). Программалық қателерді анықтау және түзету, әзірленген шешімнің тұрақтылығын тексеру.

5-кезең – Defense (қорғау). Әзірленген жобаны қорғау, орындалған программалық шешімді негіздеу және орындалған жұмыс бойынша рефлексия жүргізу.

ҚОСЫМША Г

Эксперименттік тексеруде қолданылған программалау тапсырмалары

I күрделілік деңгейі

1. Берілген n бүтін санының жұп немесе тақ екенін анықтаңыз.
2. Берілген үш санның ең үлкенін анықтаңыз.
3. Берілген n бүтін санының цифрларын кері ретпен шығарыңыз.

II күрделілік деңгейі

1. Берілген массивтегі тақ сандардың қосындысын анықтаңыз.
2. Берілген сөздің палиндром екенін анықтаңыз.
3. Берілген n натурал санының барлық бөлгіштерін анықтаңыз.

III күрделілік деңгейі

1. Берілген санның кез келген екі цифрының орнын ауыстыру арқылы палиндром сан алуға болатынын анықтаңыз.
2. Берілген сандар жиынынан ең жиі кездесетін санды анықтаңыз.
3. Берілген санның цифрларының орнын ауыстыру арқылы бастапқы саннан үлкен ең кіші санды анықтаңыз.

Мысал:
Input: 12543
Output: 13245

ҚОСЫМША Ғ

Эксперименттік тексеру барысында студенттердің программалау әрекеті туралы деректерді тіркеу

1. Жоғары оқу орнының атауы:
   - Абай атындағы Қазақ ұлттық педагогикалық университеті
   - Торайғыров университеті
   - Қазақ ұлттық қыздар педагогикалық университеті
2. Білім беру бағдарламасы:
   - 6В01507 – Информатика
   - 6В01509 – Информатика және робототехника
   - 6В06102 – Ақпараттық жүйелер
   - 6В01506 – Физика–Информатика
   - 6В01506 – Информатика
   - 6В01514 – Информатика және робототехника
3. Оқу курсы:
   - 1 курс
   - 2 курс
   - 3 курс
   - 4 курс

I күрделілік деңгейі

4. Тапсырмалар:
   1. Берілген n бүтін санының жұп немесе тақ екенін анықтаңыз.
   2. Берілген үш санның ең үлкенін анықтаңыз.
   3. Берілген n бүтін санының цифрларын кері ретпен шығарыңыз.

Программалық кодты жазыңыз:_____

I күрделілік деңгейіндегі тапсырмаларды орындау жолын түсіндіріңіз:

_____

II күрделілік деңгейі

5. Тапсырмалар:
   1. Берілген массивтегі тақ сандардың қосындысын анықтаңыз.
   2. Берілген сөздің палиндром екенін анықтаңыз.
   3. Берілген n натурал санының барлық бөлгіштерін анықтаңыз.

Программалық кодты жазыңыз:_____

II күрделілік деңгейіндегі тапсырмаларды орындау жолын түсіндіріңіз:

_____

III күрделілік деңгейі

6. Тапсырмалар:

1. Берілген санның кез келген екі цифрының орнын ауыстыру арқылы палиндром сан алуға болатынын анықтаңыз.

2. Берілген сандар жиынынан ең жиі кездесетін санды анықтаңыз.

3. Берілген санның цифрларының орнын ауыстыру арқылы бастапқы саннан үлкен ең кіші санды анықтаңыз.

Программалық кодты жазыңыз:_____

III күрделілік деңгейіндегі тапсырмаларды орындау жолын түсіндіріңіз:

_____

Тапсырмаларды орындау үдерісі туралы мәліметтер

7. Тапсырмаларды орындау барысында кездескен нақты қиындықтарды сипаттаңыз:

_____

8. Тапсырмаларды орындау барысында оқытушыға қойған сұрақтарыңызды жазыңыз:

_____

9. Тапсырмаларды орындау барысында GenAI құралдарын (мысалы, ChatGPT, GitHub Copilot, Google Gemini) пайдаланасаңыз, оларға берген нақты сұраныстарыңызды (prompt) жазыңыз:

_____