

Ш.Т. Шекербекова¹, А.Е. Сағымбаева¹, А.Е. Жаксылыков²,
Н.Е. Артықбаев¹, А.К. Нагметов^{1*}

¹ Абай атындағы Қазақ ұлттық педагогикалық университеті, Алматы қ., Қазақстан

² Торайғыров Университеті, Павлодар қ., Қазақстан

*e-mail: alibinagmetov@icloud.com

ПРОГРАММАЛАУДАН БАҒАЛАУ КРИТЕРИЙЛЕРІН ЖАСАУДАҒЫ GENAI ТЕХНОЛОГИЯСЫНЫҢ МҮМКІНДІКТЕРІ

Аңдатпа

Мақалада программалаудан студенттердің білімін бақылау тек теориялық білім деңгейін ғана емес, сонымен қатар студенттің практикалық дағдыларын, алгоритмдік ойлау қабілетін және программалау шешімдерінің тиімділігін бағалауға бағытталған деңгейлік тапсырмалар жүйесі қарастырылды. Сонымен қатар, бағалау критерийлерін құрастыруда GenAI (Generative Artificial Intelligence) технологиясының мүмкіндіктері жан-жақты сараланады. Зерттеу барысында репродуктивті, конструктивті және зерттеу сипатындағы тапсырмаларды бағалауға бағытталған объективті және бейімделген критерийлер жүйесін GenAI негізінде жасаудың жолдары ұсынылады. GenAI құралдары кодты автоматты тексеру, логикалық құрылымдарды сараптау, тест тапсырмаларын генерациялау және бағалау нәтижелерін талдауда тиімділік танытады. Сонымен қатар, оқытушының кәсіби рөлі бағалау мазмұнын анықтауда, GenAI ұсынған ұсыныстарды бейімдеуде және білім алушының даму деңгейіне сәйкес келетін шешім қабылдауда маңызды екені дәлелденді. Мақалада программалаудан деңгейлік тапсырмаларды бағалау критерийлері ұсынылып, GenAI технологиясы арқылы программалау білімін бағалаудың заманауи әдістемелік қырлары мен болашағы негізделді.

Түйін сөздер: программалау, репродуктивті тапсырма, конструктивті тапсырма, зерттеу сипатындағы тапсырма, бағалау критерийлері, GenAI.

Ш.Т. Шекербекова¹, А.Е. Сағымбаева¹, А.Е. Жаксылыков², Н.Е. Артықбаев¹, А.К. Нагметов¹

¹ Казахский национальный педагогический университет имени Абая, г. Алматы, Казахстан

² Университет Торайгырова, г. Павлодар, Казахстан

ВОЗМОЖНОСТИ ТЕХНОЛОГИИ GENAI В РАЗРАБОТКЕ КРИТЕРИЕВ ОЦЕНКИ ПО ПРОГРАММИРОВАНИЮ

Аннотация

В статье рассматривается система уровневых заданий для контроля знаний студентов по программированию, которая направлена не только на оценку теоретических знаний, но и на проверку практических навыков, алгоритмического мышления и эффективности программных решений. Также всесторонне анализируются возможности технологий GenAI (Generative Artificial Intelligence) при разработке критериев оценивания. Инструменты GenAI демонстрируют эффективность в автоматической проверке кода, анализе логических структур, генерации тестовых заданий и анализе результатов оценивания. Кроме того, доказано, что профессиональная роль преподавателя важна при определении содержания оценивания, адаптации предложений, сформированных GenAI, и принятии решений, соответствующих уровню развития обучающегося. В статье предлагаются критерии оценивания уровневых заданий по программированию, а также обосновываются современные методические аспекты и перспективы оценки знаний по программированию с использованием технологий GenAI.

Ключевые слова: программирование, репродуктивное задание, конструктивное задание, задание исследовательского характера, критерии оценивания, GenAI.

Sh.T. Shekerbekova¹, A.E. Sagymbaeva¹, A.E. Zhaksylykov², N.E. Artykbaev¹, A.K. Nagmetov¹

¹ Abai Kazakh National Pedagogical University, Almaty, Kazakhstan

² Toraigyrov University, Pavlodar, Kazakhstan

POTENTIAL OF GENAI TECHNOLOGY IN DESIGNING PROGRAMMING ASSESSMENT CRITERIA

Abstract

The article discusses a system of level assignments for monitoring students' knowledge of programming, which is aimed not only at assessing theoretical knowledge, but also at testing practical skills, algorithmic thinking, and the effectiveness of software solutions. It also comprehensively analyzes the capabilities of GenAI (Generative Artificial Intelligence) technologies in developing assessment criteria. GenAI tools demonstrate efficiency in automatic code checking, logical structure analysis, test task generation, and assessment results analysis. In addition, it has been proven that the professional role of the teacher is important in determining the content of assessment, adapting proposals generated by GenAI, and making decisions that correspond to the level of development of the student. The article proposes criteria for assessing level assignments in programming, and substantiates modern methodological aspects and prospects for assessing programming knowledge using GenAI technologies.

Keywords: programming, reproductive task, constructive task, research task, evaluation criteria, GenAI.

Негізгі ережелер

Бағалау критерийлерін құру барысында программалаудан берілетін тапсырмалар үш деңгейге бөлінуі қажет: репродуктивті деңгейде студент дайын алгоритмді қолданады, конструктивті деңгейде өзінің алгоритмін құрастырады, ал зерттеу сипатындағы деңгейде күрделі жобалық тапсырмаларды орындайды. Осы деңгейлерге сәйкес бағалау көрсеткіштері нақты, өлшенетін және GenAI технологиялары арқылы тексеруге ыңғайлы болуы тиіс. ChatGPT бағалау критерийлерін әзірлеуге, кодты тексеруге және тапсырмаға кері байланыс беруге мүмкіндік береді. Олар бағалау үдерісінің бейімділігі мен дәлдігін арттырып, студент жұмысының сапасын автоматты түрде талдауға септігін тигізеді. Сонымен қатар, бағалаудың объективтілігі мен сенімділігін қамтамасыз ету үшін GenAI оқушының жұмысын эталонмен салыстырып, синтаксистік және логикалық қателерді анықтап, нақты ұсыныстар ұсына алады.

Дегенмен, бағалау нәтижелерін қабылдау және шешім шығару барысында оқытушының рөлі өте маңызды. GenAI технологиясы көмекші құрал ретінде қарастырылып, оқытушы тарапынан академиялық адалдық пен әдістемелік талаптарға сәйкес сараланып қолданылуы тиіс. Бұл – әсіресе шығармашылық және жобалық тапсырмаларды бағалауда аса өзекті. Сонымен бірге, GenAI технологиясын тиімді қолдану үшін оқытушыларға арнайы әдістемелік даярлық қажет, себебі олар бағалау құралдарын дұрыс бағыттап, педагогикалық мақсатқа сәйкес пайдалануы тиіс. Осылайша, GenAI технологиясының мүмкіндіктерін тиімді қолдану программалаудан деңгейлік тапсырмаларды бағалауды сапалы ұйымдастыруға мүмкіндік береді. Бірақ ол оқытушыны толық алмастырмай, керісінше, оның бағалау іс-әрекетін қолдайтын интеллектуалдық серіктес ретінде қарастырылуы қажет.

Кіріспе

Ақпараттық технологиялар саласы қазіргі уақытта қарқынды даму үстінде. Заманауи технологиялардың енгізілуі мен инновациялардың пайда болуы программалаушылар мен IT мамандарына сұраныстың артуына себеп болып отыр. Сондықтан да, қазіргі кезде ЖОО IT мамандарына заманауи программалау тілдерін, веб-программалау, деректерді талдау және машиналық оқыту сияқты т.б. курстарды өздерінің оқу бағдарламаларына енгізуде [1].

Осы ретте, программалаудан студенттердің меңгерген білімін сапалы әрі объективті түрде бағалау кәсіби дайындықтың ажырамас бөлігі болып табылады. Программалау – тек теориялық біліммен ғана шектелмейтін, практикалық дағдылар мен алгоритмдік ойлау қабілетін талап ететін пән болғандықтан, оның оқу үдерісіндегі нәтижесін бақылау кешенді тәсілді қажет етеді. Студенттердің программалау саласындағы жетістігін дәл және әділ

бағалау үшін тапсырмалардың күрделілік деңгейі, оларды шешу жолдары және нәтиженің тиімділігі ескерілуі тиіс.

GenAI технологиясы бұл процесті оңтайландырудың жаңа мүмкіндіктерін ұсынып отыр: ол тапсырмаларды автоматты түрде тексеріп, қателерді көрсетіп, студенттің шешу тәсілін талдауға мүмкіндік береді. Бұл программалау пәні бойынша бақылаудың сапасын арттырумен қатар, бағалаудың объективтілігін қамтамасыз етеді. Заманауи білім беру жүйесінде GenAI автоматтандырылған бағалау жүйелерін дамыту, студенттердің программа кодын тексеру және кері байланыс беру мүмкіндік береді [2]. GenAI құралдары кодты жіберуді талдау, жылдам кері байланысты қамтамасыз ету және жіберілген қателерді анықтау арқылы студенттердің жұмысын бағалай алады. GenAI көмегімен жаңа білім беру құралдары студенттің жеке қажеттіліктеріне және оқу қарқынына бейімделе отырып, жекелендірілген оқу тәжірибесін жеңілдетеді [3]. GenAI оқыту мен оқу нәтижелерін жақсартып отырып, программалаудан берілген тапсырмаларды шешуде көмек көрсете алатынын көрсетеді [4].

Дегенмен, GenAI студенттердің программалау дағдыларын арттыра алғанымен, кодты құрудағы біржақтылыққа және жауапты пайдалануды қамтамасыз ету үшін этикалық нұсқаулардың қажеттілігіне қатысты алаңдаушылық тудырады. Сондықтан да, GenAI студенттердің дәстүрлі программалау дағдыларын алмастырмай, толықтыратын құрал ретінде қарастырылуы керек, бұл оқытушының бақылауының маңыздылығы екендігіне әкеледі [5].

Программалаудан студенттердің білімін бақылау тек теориялық білім деңгейін емес, сонымен қатар студенттің практикалық дағдыларын, алгоритмдік ойлау қабілетін және программалау шешімдерінің тиімділігін бағалауға бағытталуы тиіс. Программалаудан білімді бақылау жүйесінің мазмұны мыналарды қамтиды:

- синтаксис, тілдің құрылымы, негізгі операторлар мен деректер типтерін білу;
- берілген тапсырмалар бойынша код жаза алу, программалау ортасында жұмыс істей білу;
- мәселені талдау, тиісті алгоритмді таңдай білу және оны тиімді түрде жүзеге асыру;
- стандартты емес шешімдерді ұсына отырып, кодтың тиімділігін жақсарту.

Сонымен қатар, бағалау тапсырмалары әртүрлі деңгейде болуы қажет: репродуктивті (дайын алгоритмді қолдану), конструктивті (өз алгоритмін құру), зерттеу сипатындағы (күрделі жобалық тапсырмалар).

Сондықтан программалаудан білімді бақылауға берілетін деңгейлік тапсырмалардың түрлерімен, бағалау критерийлерінің нақтылығымен және технологиялық құралдардың тиімді қолданылуымен сипатталуы қажет.

Зерттеу әдіснамасы

Бұл зерттеуде программалаудан деңгейлік тапсырмалар негізінде бағалау критерийлерін әзірлеу үдерісіне GenAI технологияларын ықпалдастырудың педагогикалық және технологиялық мүмкіндіктері қарастырылды. Әдіснамалық негіз ретінде конструктивтік оқыту теориясы, Блум таксономиясының цифрлық үлгісі және білімді бағалаудың критериялды тәсілі алынды. Сонымен қатар, GenAI құралдарының (мысалы, ChatGPT) нақты тапсырмаларға қалай қолданылғаны бақылау және сипаттамалық әдістер арқылы талданды.

Зерттеу барысында сапалы контент талдау және жобалау әдістері қолданылды. Атап айтқанда, репродуктивті, конструктивті және зерттеу сипатындағы тапсырмалар үшін критерийлердің үлгілері жасалып, олар GenAI құралдарымен сәйкестендірілді. Бағалау критерийлерінің дұрыстығы Python тіліндегі код мысалдары мен Flask веб-қосымшаларына берілген тапсырмалар арқылы тестіленіп, сараптамалық талдау жүргізілді.

Зерттеу нәтижелері

Программалаудан студенттердің білімін бақылаудың деңгейлік тәсілі олардың программалаудың тұжырымдамаларын түсінуін және алған теориялық білімдерін практикалық жағдайдарда қолдануын жан-жақты бағалауға мүмкіндік береді.

Репродуктивті деңгей (дайын алгоритмді қолдану). Бұл деңгейде студент берілген тапсырманы дайын алгоритмдер мен құралдарды қолдана отырып орындайды. Мұнда шешімдер дайын түрде беріледі, ал студент оларды орындайды. Репродуктивті тапсырмалар есептерді шешу үшін бұрыннан бар алгоритмдерді пайдалануды қамтиды. Бұл әдісті пайдаланудың мақсаты студенттердің программалаудың негізгі құрылымдарымен танысуына және олардың бастапқы программалау дағдыларын қалыптастыруға көмектеседі [6].

Мысал. *Сұрыптау алгоритмі*: Дайын алгоритмдермен таңдау арқылы сұрыптау.

```
def selection_sort(arr):
    for i in range(len(arr)):
        min_idx = i
        for j in range(i+1, len(arr)):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
    return arr
```

Репродуктивті деңгейде сұрыптау алгоритмінің (мысалы, selection_sort) бағалау критерийін ChatGPT-дің көмегімен жасауға болады (кесте 1).

Кесте 1. Сұрыптау алгоритмінің бағалау критерийілері

Алгоритмді дұрыс таңдау	Студент тапсырма шартына сәйкес сұрыптау алгоритмін дұрыс таңдай білуі керек. Сұрыптауға selection_sort алгоритмі қолданылғаны маңызды болуы
Синтаксистік қателердің болмауы	Бағдарламаның Python тілінің синтаксисіне сай жазылуы
Логикалық дұрыстық	Алгоритм әртүрлі кіріс деректерімен жұмыс істеп, дұрыс нәтиже беруі қажет
Функцияны дұрыс құрылымдау	Алгоритм функция түрінде құрылып, параметр алып, нәтиже қайтарады
Түсіндірмелердің болуы	Кодтың әр бөлігі түсіндірілген болуы керек, қандай операцияның қайда орындалып жатқанын түсіндіретін болуы керек.
Массивтермен жұмыс	Сұрыптау барлық массив элементтеріне толық қолданылуы тиіс
Нәтижені визуалды шығару	Сұрыпталған нәтиже пайдаланушыға түсінікті түрде көрсетілуі тиіс
Түсініктемелер беру	Кодтағы негізгі бөліктерге қысқаша түсініктеме беру бағаланады

Конструктивті тапсырмалар студенттерден алгоритмдерді құруды талап етеді. Бұл әдісті пайдаланудың мақсаты студенттердің шығармашылықпен ойлауын қалыптастыру және есептерді шешу әдістерін тереңірек түсінуге ынталандыру. Студенттерге инновация мен сыни ойлауды дамытып, нақты процесті оңтайландыру үшін бірегей алгоритмді жасау тапсырылуы мүмкін [7]. Бұл деңгейде студент қандай да бір тапсырманы шешу үшін өз алгоритмін құрады. Бұл өз алдына жаңалық енгізуді немесе белгілі бір алгоритмдерді қолдануды талап етеді, бірақ бұл деңгейдің тапсырмасын орындағанда студенттің жаңа шешімдер мен тәсілдерді ойлап табу маңызды болып табылады.

Мысал. *Қарапайым калькулятор құру*. Студент математикалық амалдарды орындауға арналған калькулятор құрады, калькулятор қолданушының енгізген деректерін қабылдайды және оларды өңдейді.

```
def calculator():
    print("Қарапайым калькулятор")
    num1 = float(input("Бірінші санды енгізіңіз: "))
    operator = input("Операция таңдау (+, -, *, /): ")
```

```

num2 = float(input("Екінші санды енгізіңіз: "))
if operator == "+":
    return num1 + num2
elif operator == "-":
    return num1 - num2
elif operator == "*":
    return num1 * num2
elif operator == "/":
    return num1 / num2
else:
    return "Қате операция"
    
```

Бағалау критерийлері тапсырманың логикасын, синтаксистік дұрыстығын, қолданушымен өзара әрекеттесу сапасын, қате өңдеуді және құрылымдық ойлауды қамтуы қажет.

Жоғарыда келтірілген тапсырманы бағалау критерийін ChatGPT-дің көмегімен былай жасауға болады (кесте 2).

Кесте 2. Қарапайым калькулятор құру тапсырмасының бағалау критерийі

Студент қолданушыдан екі санды (num1, num2) енгізуді ұйымдастыра білуі керек. Бұл сандар нақты типте (float) жарияланып, input() арқылы қабылдануы қажет	- Айнымалы атаулары мағыналы; - Типтік түрлендіру (float()) қолданылған; - Қате енгізуді өңдеу қарастырылса – қосымша артықшылық
Қолданушыдан қандай операция орындағысы келетінін сұрап, оны енгізуге мүмкіндік береді (+, -, *, /)	- input() көмегімен операция таңдау мүмкіндігі бар; - Операторды енгізу кезінде бағыттаушы түсініктеме бар (мыс.: “Операция таңдау: +, -, *, /”)
Таңдалған операцияға сәйкес if, elif, else құрылымдары арқылы шарттық тексеру орындалады	- Әр арифметикалық операция (+, -, *, /) үшін бөлек шарт қарастырылған; - else блогы операцияның дұрыс еместігін өңдейді; - Логикалық қателер жоқ
Орындалатын операциялар математикалық жағынан нақты және дұрыс нәтижені қайтарады	- Әрбір амал өзіне тиесілі нәтиже қайтарады (return арқылы); - Айнымалылармен арифметикалық операциялар дұрыс орындалған.
Егер қолданушы екінші сан ретінде 0 енгізсе және / операциясын таңдаса, программа арнайы хабарлама беріп, қатені өңдейді	- if num2 == 0 шарты бар; - “0-ге бөлуге болмайды” деген хабарлама көрсетіледі; - Программа істен шықпай, қауіпсіз орындалады.
Егер қолданушы дұрыс емес оператор енгізсе (мыс., %, ^), программа “Қате операция” деген хабарлама беруі тиіс	- else блогы бар; - Қате енгізуді нақты және сыпайы өңдейді
Интерфейс қолданушыға нұсқаулық береді, түсінікті хабарламалар көрсетіледі	- Хабарламалар қазақша немесе нақты тілде жазылған; - Қолданушы не істеуі керектігін түсінеді; - Әр қадамда не сұралатыны нақты айтылған.
Программа синтаксистік қатесіз, соңына дейін орындалады	- Программа қате бермейді; - Барлық функциялар орындалады; - Код логикасы қарапайым әрі нақты

Зерттеу сипатындағы тапсырмалар (күрделі жобалық тапсырмалар) программалаудың тұжырымдамаларын кеңінен зерттеуді және қолдануды қамтитын күрделі жоба

тапсырмалары. Бұл әдісті пайдаланудың мақсаты студенттерді нақты әлемдегі қиындықтарға дайындайды және өз бетінше білім алуға ықпал етеді. Зерттеу сипатындағы тапсырмаларға студенттерден қауымдастық мәселелеріне арналған программалаық шешімдерді дайындау немесе инновациялық технологиялар бойынша зерттеулер жүргізуді талап ететін жобаларды жатқызуға болады [8]. Зерттеу сипатындағы тапсырмалар (күрделі жобалық тапсырмалар) тапсырмалары орындау барысында студент күрделі, көп аспектілі мәселелерді шешу үшін зерттеулер жүргізеді. Олар жаңа құралдар мен әдістерді іздестіреді, алгоритмдер мен шешімдер құрастырады, сол арқылы зерттеу немесе жобалық тапсырмаларды орындайды.

Мысалы. *Веб-қосымша құру*. Студент толыққанды веб-қосымша жасап шығады, мысалы, қолданушылардың тіркелу жүйесі, мәліметтер қорымен жұмыс істеу және қауіпсіздік шаралары ескеріледі.

```
from flask import Flask, render_template, request, redirect
app = Flask(__name__)
users = {}
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        if username not in users:
            users[username] = password
            return redirect('/login')
        else:
            return 'Пайдаланушы аты бұрыннан бар!'
    return render_template('register.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        if users.get(username) == password:
            return 'Қош келдіңіз!'
        else:
            return 'Қате логин немесе пароль!'
    return render_template('login.html')
if __name__ == "__main__":
    app.run(debug=True)
```

Зерттеу сипатында деңгейдегі тапсырмаларды бағалауда күрделілік, жүйелік шешімдер мен инновация негізгі критерийлер болып табылады. Студенттердің өз алгоритмдері мен шешімдерін табуы, жаңа әдістер мен құралдарды қолдануы, және нәтиже сапасын бағалау маңызды. Сонымен қатар, қауіпсіздік, қолданушы интерфейсі және қателіктерді өңдеу сияқты аспектілер де есепке алынуы тиіс. Бұл тапсырманы бағалау критерийлері студенттің Flask фреймворкі негізінде веб-қосымша жасау тапсырмасын орындау сапасын бағалауға арналған. Қосымшада тіркелу, жүйеге кіру, мәліметтермен жұмыс және қарапайым қауіпсіздік шаралары қарастырылады. Бұл тапсырманы бағалау критерийін ChatGPT-дің көмегімен былай жасауға болады (кесте 3.)

Кесте 3. Веб-қосымша құру тапсырмасының бағалау критерийі

Қолданушы қарапайымдылығы және қолдануға ыңғайлылығы	интерфейсінің және қолдануға	register.html және login.html беттері қарапайым әрі түсінікті түрде безендірілген
Тіркелу функциясының дұрыс жұмыс істеуі		Пайдаланушы деректері дұрыс қабылданып, қайталанатын пайдаланушыға ескерту беріледі
Жүйеге кіру функциясының сенімділігі		Тіркелген пайдаланушы жүйеге кіре алады, қате енгізгенде дұрыс ескерту шығарылады
Мәліметтер құрылымын дұрыс пайдалану		users сөздігі арқылы деректерді сақтау логикасы дұрыс қолданылған
Функциялардың логикалық құрылымы		Кодтың логикалық құрылымы нақты, тиімді ұйымдастырылған (route, condition блоктары т.б.)
Қауіпсіздік шараларының қолданылуы		Қарапайым қауіпсіздік (бірдей логинді қайта енгізуге шектеу, парольдің тексерілуі) ескерілген
Flask кітапханасын дұрыс қолдану		Flask-тің негізгі компоненттері (route, request, render_template) дұрыс және тиімді пайдаланылған
Қателерді өңдеу және қолданушыға жауап қайтару		Қате логин/пароль немесе тіркелу кезіндегі хабарламалар нақты әрі түсінікті
Кодтың тазалығы мен құрылымы		Код ықшам, артықшылықтан ада, оқуға жеңіл
Қосымша функционал қосу (қосымша бонустық ұпай)		Мысалы, логин сессиясы, құпиясөзді хэштеу, logout функциясы қосылған болса

Программалаудан деңгейлік бақылау тапсырмаларын пайдалану ұтымды болғанымен, кейбір оқытушылар күрделі тапсырмаларға көп көңіл бөлу программалауды жаңадан үйренушілерге шамадан тыс әсер етіп, олардың оқу үдерісіне кедергі келтіруі мүмкін деп санайды. Программалаудан студенттердің орындаған репродуктивті, конструктивті және зерттеу сипатындағы деңгейлік тапсырмаларын бағалаудың критерийлері келтірілді (Кесте 4).

Кесте 4. Программалаудан деңгейлік тапсырмаларды бағалау критерийлері

Критерийлер	Репродуктивті деңгей	Конструктивті деңгей	Зерттеу сипатындағы деңгей
Алгоритмдер мен әдістердің күрделілігі	Дайын алгоритмдер пайдаланылады (мысалы, Bubble Sort, Евклид алгоритмі)	Студент өз алгоритмдерін құрады (мысалы, Insertion Sort, калькулятор құру)	Күрделі және көпқырлы алгоритмдер мен шешімдер (мысалы, Random Forest, Web қосымшалар)
Тапсырманың күрделілігі	Тапсырмалар қарапайым және дайын алгоритмдермен шешілетін болады.	Тапсырмалар орташа күрделілікке ие және студенттер алгоритмдерді өз бетінше құрастырады.	Тапсырмалар күрделі, зерттеуді және жаңа әдістерді қолдануды талап етеді.
Қолданылатын құралдар мен әдістер	Дайын құралдар мен кітапханалар қолданылады.	Студенттің өзінің ойлап тапқан әдістері мен құралдары қолданылады.	Күрделі зерттеу әдістері, жаңа құралдар мен әдістер қолданылуы мүмкін (мысалы, машинамен оқыту).
Құрылымдық деңгей	Кодты түсінбеу немесе толық	Кодтың құрылымдық жағынан тиімділігі мен логикасы маңызды.	Кодтың күрделілігі жоғары, құрылымның

	<i>орындамау мүмкіндігі өте аз.</i>		<i>дұрыс және тиімді ұйымдастырылуы қажет.</i>
<i>Түсініктемелер мен құжаттама</i>	<i>Түсініктемелер мен құжаттама көбінесе қажет емес, себебі код дайын күйінде беріледі.</i>	<i>Кодты түсіндіру және түсініктемелер беру маңызды.</i>	<i>Жоба мен зерттеу нәтижелері туралы толық құжаттама және түсініктеме қажет.</i>
<i>Жаңа шешімдер мен тәсілдер</i>	<i>Жаңа шешімдер мен тәсілдер енгізілмейді.</i>	<i>Студент жаңа алгоритмдер мен шешімдерді өзі табады.</i>	<i>Жаңа шешімдер мен инновациялар ұсынылады, зерттеу жүргізіледі.</i>
<i>Қателіктерді өңдеу</i>	<i>Қателіктер өте сирек кездеседі, себебі алгоритм дайын түрде беріледі.</i>	<i>Қателіктерді дұрыс өңдеу, мысалы, қате енгізулер мен нәтижелер үшін шарттар мен тексерулер қосу маңызды.</i>	<i>Қателіктерді болдырмау үшін кеңейтілген қателіктерді өңдеу жүйесі мен тексерулер қолданылады.</i>
<i>Қолданушы интерфейсі</i>	<i>Қолданушы интерфейсі қарапайым болады (көп жағдайда жоқ).</i>	<i>Қолданушы интерфейсі студенттің өз тарапынан құрастырылады.</i>	<i>Қолданушы интерфейсі ыңғайлы, жүйенің тиімділігіне байланысты жоғары сапалы болуы тиіс.</i>
<i>Қолданушы енгізу және нәтижелер</i>	<i>Қолданушы енгізуі қарапайым, нәтижелер нақты көрсетіледі.</i>	<i>Қолданушыдан алынатын мәліметтер мен нәтижелер түсінікті болуы қажет.</i>	<i>Қолданушыдан алынатын деректердің сапасы мен нәтижелердің күрделілігі жоғары болуы мүмкін.</i>
<i>Шешімдердің қолданылуы</i>	<i>Алгоритмдер тек белгілі бір жағдайларда қолданылады.</i>	<i>Алгоритмдер кең ауқымды мәселелерге арналған, бірақ көбінесе белгілі бір жағдайларда қолданылады.</i>	<i>Алгоритмдер түрлі салаларға қатысты және кең ауқымды проблемаларды шешуге арналған.</i>

Дискуссия

Алдағы уақытта GenAI технологиясы бұл үдерісті оңтайландырудың жаңа мүмкіндіктерін ұсына алады, ол тапсырмаларды автоматты түрде тексеріп, қателерді көрсетіп, студенттің шешу тәсілін талдауға мүмкіндік береді. Бұл программалау пәні бойынша бақылаудың сапасын арттырумен қатар, бағалаудың объективтілігін қамтамасыз етеді.

Сонымен, программалаудан студенттердің білімді бақылауда деңгейлік тапсырмаларды қолдануды негіздей келе және жасалған деңгейлік бағалау критерийлерін жасау нәтижесінде репродуктивті, конструктивті және зерттеу сипатындағы (күрделі жобалық тапсырмалар) тапсырмалардың ChatGPT-дің көмегімен бағалау критерийлерін оқытушының өзі тексеріп отырғаны жөн деп саналады.

Себебі, GenAI-дың (мысалы, ChatGPT, Claude, Gemini) жасаған бағалау критерийінің дұрыстығын тексеру үшін бірнеше әдістемелік және практикалық әдістерді қолдануға болады. Алдымен, бағалау критерийлері программалауды оқу мақсаттарына сәйкес келуі, сонымен қатар студенттің білім деңгейін (репродуктивті, конструктивті, зерттеушілік) нақты өлшей алуы қажет. GenAI құрған критерийлерді оқытушының құрған критерийлерімен, академиялық еңбектердегі ұқсас бағалау жүйелерімен (мысалы, Bloom's Taxonomy, SOLO Taxonomy) салыстыру. Содан кейін пилоттық тестілеу жүргізу керек, ол үшін бірқатар студенттерге бірдей тапсырма беріп, оны GenAI жасаған критериймен және оқытушы жасаған бағалау критерийлерімен бағалап көріп, екі бағалау арасындағы сәйкестік деңгейін салыстырған жөн. Рефлексия жасау қажет, студенттермен кері байланыс GenAI ұсынған критерийді жетілдіруге

мүмкіндік береді. Бір тапсырманы бірнеше рет түрлі формада беріп (басқа сөздермен жазып, немесе кішігірім өзгеріс енгізіп), GenAI әр жолы бірдей не ұқсас баға бере ала ма, тексеріп көрген жөн.

Қорытынды

Зерттеу нәтижелері бойынша, программалаудан білімді бағалаудың тиімділігі тапсырмалардың мазмұны мен бағалау критерийлерінің нақтылығына тікелей байланысты. Зерттеу барысында GenAI технологиясының мүмкіндіктері репродуктивті, конструктивті және зерттеу сипатындағы деңгейлік тапсырмалардың бағалау критерийлерін құрастыру оң әсер ететіні анықталды. GenAI құралдары: репродуктивті деңгейде – дайын кодты түсіндіру, синтаксистік қатені табу, қарапайым тапсырмаларды тексеру; конструктивті деңгейде – алгоритмдік шешімдерді сараптау, логиканың дұрыстығын бағалау; зерттеушілік деңгейде – күрделі жобалық шешімдер мен архитектуралық құрылымдарды талдау мен салыстыру; сияқты функцияларды орындай отырып, тапсырмаларға дәл және бейімделген кері байланыс беруге қабілетті.

Сонымен қатар, GenAI ұсынған мүмкіндіктерді сараптап, оларды оқу мақсатына сәйкестендіру, критерийлерді оқытудың нақты контексіне бейімдеуде оқытушының рөлі басым. Оқытушы бағалаудың мазмұндық жағын анықтайтын, білім алушының даму деңгейін диагностикалауға және GenAI көмегін тиімді бағыттайтын негізгі тұлға болып табылады.

Қорытындылай келе, программалаудан деңгейлік тапсырмаларға негізделген бағалау критерийлерін GenAI технологиясы арқылы құру болашақ цифрлық педагогиканың өзекті бағыты болып табылады. Бұл тәсіл білім беру сапасын арттырып қана қоймай, оқытуды дараландыру, бейімдеу және цифрлық сауаттылықты қалыптастыруда маңызды рөл атқарады.

АЛҒЫС

Бұл зерттеуді Қазақстан Республикасы Ғылым және жоғары білім министрлігі Ғылым комитеті қаржыландырды (Грант № AP23490592)

Пайдаланылған дереккөздердің тізімі

[1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. – “Introduction to Algorithms” (3rd Edition). QA76.6.I5858 2009 005.1—dc22, ISBN 978-0-262-03384-8, p 1313.

[2] Сағымбаева А.Е., Жақсылықов А.Е., Шекербекова Ш.Т., Жамкеева А.Б. Генеративті жасанды интеллект технологиясының программалаудан ЖОО студенттерінің білімін бақылаудағы рөлі. ҚазҰПУ Хабаршы «Физика-математика ғылымдары» серия. – № 3 (87). – 2024. Б. 373-384. <https://doi.org/10.51889/2959-5894.2024.87.3.027>

[3] Silva, T. L. da, Vidotto, K. N. S., Tarouco, L. M. R., & Silva, P. F. da. (2024). Inteligência artificial generativa no ensino de programação: um mapeamento sistemático da literatura. *Renote*, 22(1), 262–272. <https://doi.org/10.22456/1679-1916.141553>

[4] NguyenDuc, A., Cabrero-Daniel, B., Arora, C., Przybyłek, A., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K., Saari, M. Y., Zhang, Z., Lê, H., Quan, T., & Abrahamsson, P. (2023). *Generative Artificial Intelligence for Software Engineering - a Research Agenda*. <https://doi.org/10.2139/ssrn.4622517>

[5] Atemkeng, M., Hamlomo, S., Welman, B., Oyentunji, N., Ataei, P., & Fendji, J. L. E. K. (2024). *Ethics of Software Programming with Generative AI: Is Programming without Generative AI always radical?* <https://doi.org/10.48550/arxiv.2408.10554>

[6] Ala-Mutka, K. (2005). A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2), 83–102. <https://doi.org/10.1080/08993400500150747>

[7] Nayak, S., Agarwal, R., & Khatri, S. P. (2022). Automated Assessment Tools for grading of programming Assignments: A review. *International Conference on Computational Collective Intelligence*, 1–4. <https://doi.org/10.1109/ICCCI54379.2022.9740769>

[8] Rajesh, S., Rao, V. R., & Thushara, M. G. (2024). *Comprehensive Investigation of Code Assessment Tools in Programming Courses*. <https://doi.org/10.1109/i2ct61223.2024.10543863>

References

- [1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. – “Introduction to Algorithms” (3rd Edition). QA76.6.I5858 2009 005.1—dc22, ISBN 978-0-262-03384-8, p 1313.
- [2] Sagymbaeva A.E., Zhaqsylyқov A.E., Shekerbekova Sh.T., Zhamkeeva A.B. (2024) Generativti zhasandy intellekt tehnologijasynyn programmalaudan ZhOO studentterinin bilimin bakylaudagy roli [The role of technologies of generative artificial intelligence in the control of knowledge of university students in programming]. *KazUPU Habarshy «Fizika-matematika gylymdary» serija. № 3 (87). 373-384.*
<https://doi.org/10.51889/2959-5894.2024.87.3.027> (In Kazakh)
- [3] Silva, T. L. da, Vidotto, K. N. S., Tarouco, L. M. R., & Silva, P. F. da. (2024). Inteligência artificial generativa no ensino de programação: um mapeamento sistemático da literatura. *Renote*, 22(1), 262–272.
<https://doi.org/10.22456/1679-1916.141553>
- [4] NguyenDuc, A., Cabrero-Daniel, B., Arora, C., Przybyłek, A., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K., Saari, M. Y., Zhang, Z., Lê, H., Quan, T., & Abrahamsson, P. (2023). Generative Artificial Intelligence for Software Engineering - a Research Agenda.
<https://doi.org/10.2139/ssrn.4622517>
- [5] Atemkeng, M., Hamlomo, S., Welman, B., Oyentunji, N., Ataei, P., & Fendji, J. L. E. K. (2024). Ethics of Software Programming with Generative AI: Is Programming without Generative AI always radical?
<https://doi.org/10.48550/arxiv.2408.10554>
- [6] Ala-Mutka, K. (2005). A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2), 83–102. <https://doi.org/10.1080/08993400500150747>
- [7] Nayak, S., Agarwal, R., & Khatri, S. P. (2022). Automated Assessment Tools for grading of programming Assignments: A review. *International Conference on Computational Collective Intelligence*, 1–4. <https://doi.org/10.1109/ICCCI54379.2022.9740769>
- [8] Rajesh, S., Rao, V. R., & Thushara, M. G. (2024). Comprehensive Investigation of Code Assessment Tools in Programming Courses. <https://doi.org/10.1109/i2ct61223.2024.10543863>