

Бағдарламалау мен математикалық ойлаудың өзара байланысы: теориялық талдау

ЖАРИЯЛАНДЫ 22.04.2026	ТІРЕК СӨЗДЕР абстракция, алгоритм, бағдарламалау, декомпозиция, есептеуіш ойлау, итерация, математикалық ойлау, модельдеу, тексеру	СІЛТЕМЕ https://bilimger.kz/188326/
---------------------------------	--	---

Алтынбек Жанна Нұрлыбекқызы

Л.Н.Гумилев атындағы ЕҰУ Механика-математика факультетінің алгебра және геометрия кафедрасының магистранты, Астана, Қазақстан

Ғылыми жетекші — **А.С.Науразбекова**

Аңдатпа

Аталған ғылыми мақалада бағдарламалау мен математикалық пайымдаудың өзара сабақтастығы теориялық тұрғыдан зерделенеді. Зерттеу барысында ұғымдық терминологиялық сараптама, әдеби шолу, мазмұндық талдау, салыстырмалы сараптама және логикалық құрылымдық модельдеу әдістері кешенді түрде пайдаланылды. Нәтижесінде екі салаға да ортақ когнитивті операциялар кешені айқындалды, олардың қатарына абстракция, декомпозиция, алгоритмдеу, жалпылау, дәлелдеу және тексеру амалдары жатқызылды. Бағдарламалау үдерісі математикалық идеяларды нақты орындалатын әрекеттерге айналдыратыны, сондай ақ қателерді анықтау мен түзету арқылы тексеру мәдениетін нығайтатыны негізделді. Сонымен қатар «модель — алгоритм — бағдарлама — тексеру — итерация» циклі пәндер тоғысындағы әмбебап механизм ретінде ұсынылды. Талқылау бөлімінде бағдарламалаудың математикалық ойлауды жетілдіруі тапсырма мазмұны мен рефлексиялық талдауға тікелей тәуелді екені дәлелденіп, пәнаралық дағдылар тасымалының негізгі шарттары сипатталды.

Кілт сөздер: бағдарламалау, математикалық ойлау, есептеуіш ойлау, алгоритм, абстракция, декомпозиция, модельдеу, тексеру, итерация.

Кіріспе. Қазіргі білім беру кеңістігінде бағдарламалау мен математика жеке дара салалар ретінде көрінгенімен, олардың ішкі табиғаты ортақ мүддеге қызмет етеді. Екі пән де адам бойындағы ойлау қабілетін жүйелеуге, нақты дәлелдерге сүйенуге және

күрделі мәселелерді кезеңдік шешімдерге жіктеуге бағытталған. Бағдарламалау мен математикалық пайымдаудың сабақтастығын сипаттауда «есептеуіш ойлау» (computational thinking) терминінің маңызы зор. Дж. Винг аталған ұғымды тек қана информатика шеңберімен шектемей, оны кез келген тұлғаға қажетті әмбебап құрал ретінде қарастырады. Оның пайымдауынша, күрделі түйткілдерді ұсақ бөлшектерге ыдырату, оларды белгілі бір қағидаларға бағындыру және қайталанатын әрекеттер арқылы нәтижеге жету адамның интеллектуалдық әлеуетін арттырады [1, 33 б.]. Сондай ақ С. Гровер мен Р. Пиа оның құрамына декомпозиция, үлгілерді тану және алгоритмдеу іспетті маңызды элементтерді жатқызады. Олар бұл дағдылардың бір пәннен екінші пәнге еркін ауысатынын (transfer), яғни бір салада игерілген жүйелілік екінші салада да жемісін беретінін дәлелдеген [2, 39 б.].

Осы байланыстың алтын арқауы ретінде алгоритм ұғымын атауға болады, себебі кез келген есепті шешуге арналған ретті қадамдар жиынтығы ретінде алгоритмдер математика мен бағдарламалау арасындағы берік көпір қызметін атқарады [3, 5 б.]. Мұнда математика мәселенің логикалық моделін ұсынса, бағдарламалау сол логиканы тірі жүйеге айналдырады. Теориялық білімді шынайы тәжірибемен ұштастыру мәселесінде К. Бреннан мен М. Резник ерекше пікір білдіріп, олар оқушының жетістігін тек соңғы нәтижемен емес, оған жету жолындағы ізденіс, қатені түзету және қайта құрастыру үдерісімен бағалауды ұсынады [4, 7 б.]. Мұндай тәсіл математикалық дәлелдеу мәдениетімен тікелей сабақтас. Е.К. Хеннер білім беру саласындағы есептеуіш ойлаудың негізі ретінде формальдау мен нәтижені тексеру мәдениетін көрсетсе [5, 21 б.], Л.М. Фридман математиканы оқытудағы ұғымдарды жүйелеу мен оқу әрекетін ұйымдастырудың іргелі қағидаларын айқындап берді [6, 58 б.]. Бағдарламалау ортасы С. Паперт атап өткендей, оқушыға өз идеяларын іс жүзінде сынап көруге және «ұстап көруге» мүмкіндік беретін бірегей конструктивті кеңістікке айналады [7, 192 б.].

Отандық ғылыми зерттеулерде де есептеуіш ойлаудың цифрлық дәуірдегі маңызы жан жақты талданған. Зерттеушілер алгоритмдік және модельдеу дағдыларының оқу үлгеріміне оң әсер ететінін алға тартады [8, 170 б.]. Ал бағдарламалауды оқыту логикалық пайымдау мен абстрактілі ойлауды жетілдірудің басты құралы екенін Э.М. Каган өз еңбектерінде негіздеген [9, 445 б.]. Ұсынылып отырған теориялық талдаудың түпкі мақсаты бағдарламалау әрекеті мен математикалық пайымдаудың әдістемелік тұтастығын дәлелдеу болып табылады. Біз олардың ортақ когнитивті операцияларын, атап айтқанда абстракция мен декомпозицияны ашып көрсетуді көздейміз. Бұл талдаудың нәтижелері алдағы бөлімдерде бағдарламалау тапсырмаларының математикалық қабілетті қалай шыңдайтынын және математикалық білімнің код жазудағы ұқыптылықты қалай арттыратынын түсіндіруге негіз болады [10, 145 б.].

Зерттеу әдістері. Зерттеу жұмысының мақсатына жету үшін теориялық және аналитикалық әдістер кешені кезең кезеңімен қолданылды. Алдымен бағдарламалау

мен математикалық пайымдау салаларына ортақ ұғымдық аппаратты нақтылау мақсатында анықтамалық терминологиялық талдау жүргізілді. Осы кезеңде «есептеуіш ойлау», «алгоритм», «абстракция», «декомпозиция», «модельдеу» және «тексеру» секілді іргелі категориялардың мазмұны сараланып, олардың өзара қисынды байланысы айқындалды. Зерттеудің теориялық өзегін құрау үшін іріктелген ғылыми мақалалар мен оқу әдістемелік еңбектерге контент талдау жасалды. Атап айтқанда, Дж. Вингтің есептеуіш ойлау тұжырымдамасы, С. Гровер мен Р. Пианың мектеп деңгейіндегі зерттеулері және Е.К. Хеннердің білім беру саласындағы пайымдаулары жүйеленді.

Екінші бағыт аясында пәнаралық байланыстарға дәлелді құрылым беру үшін салыстырмалы талдау мен логикалық құрылымдық модельдеу әдістері іске қосылды. Салыстырмалы талдау тәсілі математикалық ойлау операциялары мен бағдарламалаудағы практикалық әрекеттердің сәйкестігін анықтауға мүмкіндік берді. Мәселен, жалпылау мен формальдау үдерістері алгоритм құру және кодты жүйелеу ісімен ұштастырылды. Нәтижесінде «математикалық модель — алгоритм — бағдарламалық іске асыру — тексеру» тізбегінен тұратын арнайы байланыс схемасы негізделді.

Сонымен қатар, логикалық құрылымдық модельдеу әдісі алынған тұжырымдарды біртұтас жүйеге келтіруге жағдай жасады. Бұл әдіс бағдарламалау тапсырмаларының математикалық таным компоненттерін қалай жандандыратынын көрсететін интерпретациялық негіз қалыптастыруға көмектесті. Таңдалған әдістер жиынтығы зерттеу нысанын жан жақты зерделеуге және алынған нәтижелердің ғылыми негізділігін қамтамасыз етуге бағытталды.

Зерттеудің нәтижелері мен оларды талқылау бөлімі академиялық бай тілмен, дефиссіз және «бұл деген сөз» тіркесін қолданбай, мазмұнын толық сақтай отырып қайта өңделді:

Зерттеу нәтижелері мен талқылау. Ғылыми ізденіс барысында ұғымдық терминологиялық сараптама, әдеби шолу, мазмұндық талдау және логикалық құрылымдық модельдеу әдістерін ұштастыру нәтижесінде маңызды заңдылық айқындалды. Бағдарламалау мен математика жай ғана сабақтас салалар емес, олар бірін бірі толықтырып, интеллектуалдық әлеуетті арттыратын бірегей ойлау тренажеры іспетті. Теориялық дереккөздерді зерделеу барысында есепті құрамдас бөліктерге жіктеу (декомпозиция), басты мәселені іріктеп алу (абстракция), шешім жолдарын жүйелеу (алгоритмдеу) және нәтиженің дұрыстығын айғақтау (верификация) секілді операциялардың негізгі өзек екені көрінді. Егер аталған үдерістер математика ғылымында ежелден қалыптасқан мәдениет болса, бағдарламалау оны күнделікті практикалық әрекетке айналдырады. Код жазу барысында логикалық ауытқуларға жол берілмейді, өйткені жүйе кез келген қателікті дереу анықтап, түзетуді талап етеді. Осы орайда Дж. Вингтің есептеуіш ойлауды әмбебап тәсіл ретінде қарастыруы.

Ғылыми еңбектерге жасалған мазмұндық талдау ортақ түйіндерді бір жүйеге топтастыруға мүмкіндік берді. С. Гровер мен Р. Пиа есептеуіш ойлау компоненттерінің пәнаралық деңгейде еркін ауысатынын негіздесе, К. Бреннан мен М. Резник білім алушының жетістігін бағалауда тек нәтижеге емес, құру, сынау және түзету үдерістеріне басымдық беруді ұсынады. Е.К. Хеннердің есептеуіш ойлауды формальдау және алгоритмдеумен байланыстыруы. Математикалық дәлелдеу дәстүріне өте жақын. Л.М. Фридман жүйелеген математикалық ұғымдарды қалыптастыру мен жалпылау логикасы. бағдарламалау әрекетімен үндестік тапты. Мұндағы басты айырмашылық тек қолдану формасында: математикалық есептеулер көбіне қағаз бетіндегі теория күйінде қалса, бағдарламалау сол идеяларды қозғалысқа түсетін механизмге айналдырады.

Салыстырмалы талдау арқылы математикалық ойлау операциялары бағдарламалаудағы нақты іс әрекеттермен сәйкестендірілді. Зерттеудің ең айқын тұсы бір логикалық амалдың екі салада түрліше аталғанымен, атқаратын қызметінің бірдей болуында. Мысалы, математикадағы «дәлелдеу» үдерісі бағдарламалауда «тестілеу, трассировка және дебаг» түрінде көрініс табады. Ал математикалық «модельдеу» әрекеті бағдарламалауда «деректер құрылымы мен симуляция» арқылы іске асады. Мұндай сәйкестік жай ғана ұқсастық емес, ол білім алушының бір салада жинақтаған тәжірибесін екінші салада автоматты түрде қолдануына жағдай жасайтын ішкі механизм.

Кесте 1 – Математикалық ойлау операциялары мен бағдарламалаудағы көрінісі

Математикалық ойлау операциясы	Математикадағы типтік әрекет	Бағдарламалаудағы балама әрекет	Күтілетін нәтиже
Абстракция	Маңызды шарттарды іріктеу, артық деректерді жою	Айнымалыларды анықтау, функция мен модульдерге жіктеу	Ойдың айқындығы, техникалық қателердің азаюы
Декомпозиция	Күрделі есепті шағын бөліктерге ыдырату	Процедуралар мен ішкі функцияларға бөлу	Ауқымды мәселелерді басқару қабілетінің артуы
Алгоритмдеу	Шешімге апаратын ретті қадамдарды құру	Псевдокод, блок схема немесе код құрылымын түзу	Әрекеттердің жүйелілігі мен қисындылығы
Жалпылау	Формулалар мен теоремалар арқылы заңдылық шығару	Қайта қолданылатын функциялар мен шаблондар жасау	Бірыңғай білімді түрлі жағдайларда қолдану
Тексеру (Дәлелдеу)	Дәлелдемелер келтіру, қарсы мысалдар іздеу	Тест кейстерін әзірлеу, дебаг, трассировка жасау	Нәтиженің дұрыстығына көз жеткізу мәдениеті
Модельдеу	Математикалық модельдер құрастыру	Деректер моделін, симуляциялық модельдерді түзу	Нақты құбылыстарды жүйелеу дағдысы

Кесте 1 мәліметтері бағдарламалаудың математиканы алмастырмайтынын, керісінше оның операцияларын бекітіп, жұмыс істейтін қалыпқа келтіретінін көрсетеді. Алгоритмнің базалық анықтамасы қадамдық жүйе ретінде берілетіндіктен, бағдарламалау аталған ұғымды өмірге әкелетін басты орта болып табылады. Мұнда С. Паперттің конструктивті тәжірибе арқылы мағынаны тереңдету идеясы айқын көрінеді: оқушы дайын формуланы жаттап қана қоймай, оны өз әрекетімен құрастырып шығады. Осылайша, математикадағы көрінбейтін ойлау қадамдары бағдарламалауда нақты қозғалысқа ие болады.

Кестеден шығатын негізгі тұжырым бағдарламалаудың «айна» эффектісіне ие болуында. Ол білім алушының ойлау жүйесіндегі осал тұстарды дереу айқындап береді. Егер абстракциялау қабілеті төмен болса, код шашыраңқы болады; декомпозиция дағдысы жетіспесе, адам үлкен тапсырма алдында қауқарсыз қалады; тексеру мәдениеті қалыптаспаса, қателікті табу мүмкін болмайды. Бағдарламалаудың адамды ұялту үшін емес, оның ойлау жүйесін түзету және жетілдіру үшін қажетті құрал екені осылайша дәлелденеді.

Логикалық құрылымдық модельдеу үдерісінің қорытындысы бойынша пәнаралық сабақтастықты айқындайтын біртұтас схема әзірленді. Аталған құрылым жай ғана көрнекілік емес, ол оқытудың ішкі механикасын ашатын кешенді циклді сипаттайды. Оның құрамына келесідей маңызды кезеңдер кіреді:

- Мәселені терең ұғыну және оны формальдау (математикалық модель түзу);
- Шешімге бағытталған ретті қадамдарды айқындау (алгоритм жасау);
- Тұжырымдалған ойды практикалық тұрғыда жүзеге асыру (бағдарламалау);
- Нәтиженің дұрыстығын айғақтау және олқылықтарды жою (верификация);
- Құрылымды қайта қарау және оны оңтайландыру (итерация).

Бұл цикл К. Бреннан мен М. Резник ұсынған «процесті бағалау» логикасымен толық үйлеседі. Олардың пайымдауынша, білім алушының зияткерлік деңгейі дайын жауаппен емес, шешімді қайта құру және түзету қабілетімен өлшенеді. Сонымен қатар Е.К. Хеннер атап көрсеткендей, есептеуіш ойлаудың негізгі өзегі формальдау мен тексеру мәдениетінде жатыр. Аталған қасиет математика ғылымындағы дәлелдемелер жүйесінің практикалық баламасы болып табылады. Дж. Вингтің бұл дағдыларды әмбебап құрал ретінде қарастыруы да осы цикл арқылы өз дәлелін тапты: мұнда тек код жазу емес, жалпы мәселелерді шешудің жоғары мәдениеті қалыптасады.

Кесте 2 – Байланыстың циклдік моделі және танымдық нәтижелер

Кезең	Негізгі бағдар	Ойлау әрекеті	Қалыптасатын қасиет	Ықтимал кедергілер
-------	----------------	---------------	---------------------	--------------------

1. Формальдау	Есептің талабы қандай?	Шарттарды жүйелеу, параметрлерді іріктеу	Дәлдік, ұғымдармен жұмыс істеу	Тапсырма мәнін толық түсінбеу
2. Алгоритм құру	Шешім жолы қандай?	Жоспарлау, кезеңдерге жіктеу	Жүйелілік пен реттілік	Қадамдардың логикалық ауысуы
3. Бағдарламалау	Оны қалай іске асырамын?	Кодтау және құрылымдау	Тәртіп пен ұйымдасқандық	Синтаксистік шектеулерге тірелу
4. Тексеру	Нәтиже дұрыс па?	Сынақтан өткізу, қатемен жұмыс	Сын тұрғысынан пайымдау	Сәтсіздіктен немесе қатеден жасқану
5. Итерация	Жетілдіру мүмкін бе?	Оңтайландыру, жалпылау	Икемділік пен төзімділік	Сабырлылықтың жетіспеуі

Зерттеу барысында анықталған ең маңызды жайт математикалық ойлаудың көбіне оқушы үшін көрінбейтін, ішкі процесс ретінде қабылдануында. Ал бағдарламалау ортасы бұл үдерістерді кезең кезеңімен сыртқа шығаруға мәжбүрлейді. Мысалы, егер формальдау кезеңі тиісті деңгейде орындалмаса, жазылған код өз қызметін түсіндіре алмайды. Тексеру мәдениеті қалыптаспаған жағдайда білім алушы кез келген кедергіге тап болғанда математикада да, бағдарламалауда да тығырыққа тіреледі.

Л.М. Фридманның математикалық ұғымдарды қалыптастыру және оқу әрекетін ұйымдастыру туралы тұжырымдары осы циклдің іргетасы болып табылады: әрекет дұрыс ұйымдастырылса, танымдық нәтиже де сапалы болады. Ал классикалық алгоритмдер жүйесі бағдарламалау мен математиканың тоғысқан инженерлік қырын аша түседі. Осылайша оқушы біртіндеп жай ғана жауап іздеуден күрделі шешім құрастыру деңгейіне көтеріледі, бұл оның интеллектуалдық дамуының жаңа сатысы болып саналады.

Бағдарламалау мен математиканың сабақтастығы тек оқыту үдерісі танымдық рефлексия мен нақты мәселелерді шешуге негізделгенде ғана жүзеге асып, оқушының логикалық қателермен жұмыс істеу мәдениетін қалыптастырады. Бұл жүйелі механизм білім алушының қиындықтарға төзімділігін арттырып, қателікті кедергі емес, ойлау жүйесін жетілдіретін құнды ақпарат ретінде қабылдауға дағдыландырады.

Қорытынды. Жүргізілген теориялық талдау бағдарламалау мен математикалық пайымдау арасындағы байланыстың жай ғана ұқсастық емес, ортақ танымдық механизмдерге негізделген жүйелі құбылыс екенін айқындады. Ғылыми еңбектерді ұғымдық және салыстырмалы тұрғыдан зерделеу нәтижесінде абстракция, декомпозиция, алгоритмдеу, жалпылау және тексеру секілді операциялардың екі салада да негізгі қозғаушы күш екені нақтыланды. Егер математика аталған амалдарды теориялық деңгейде тереңдетсе, бағдарламалау оларды нақты іс әрекет арқылы бекітеді. Ол ойлау үдерісін процедуралық қалыпқа түсіріп, нәтижені тестілеу мен

қателерді түзету (дебаг) арқылы жүйелі тексеруге мәжбүрлейді. Осылайша, бағдарламалау математикалық ойлауды өздігінен емес, тек мақсатты түрде ұйымдастырылғанда, яғни мазмұнды тапсырмалар мен рефлексиялық талдау негізінде ғана күшейтетін тиімді орта ретінде танылды.

Логикалық құрылымдық модельдеу әдісі арқылы негізделген «математикалық модель — алгоритм — бағдарлама — тексеру — итерация» циклі пәнаралық сабақтастықтың әмбебап схемасы ретінде ұсынылды. Аталған цикл білім алушының бойында жүйелілік, дәлдік, жауапкершілік және өз әрекетін бақылау дағдыларын қалыптастырады. Сонымен қатар қателікке деген көзқарасты өзгертіп, оны сәтсіздік емес, керісінше түзетуге болатын құнды ақпарат ретінде қабылдауға үйретеді.

Нәтижесінде зерттеу нысаны толық зерделеніп, бағдарламалау мен математиканың бірін бірі алмастырмайтыны, керісінше өзара толықтыратыны дәлелденді. Математикалық ойлау сапасын арттырудың басты жолы бағдарламалауды алгоритмдік және рефлексивті форматта қолдану болса, бағдарламалаудың сапалық деңгейін көтерудің негізгі тірегі математикалық мәдениет, яғни шешімнің дәлдігі мен модельдеу қабілеті болып табылады.

Әдебиеттер:

1. Wing J.M. *Computational thinking // Communications of the ACM.* – 2006. – Vol. 49, No. 3. – P. 33-35. – DOI: 10.1145/1118178.1118215.
2. Grover S., Pea R. *Computational Thinking in K-12: A Review of the State of the Field // Educational Researcher.* – 2013. – Vol. 42, No. 1. – P. 38-43. – DOI: 10.3102/0013189X12463051.
3. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. *Introduction to Algorithms.* – 3rd ed. – Cambridge, MA: The MIT Press, 2009. – 1312 p. – ISBN 978-0-262-53305-8.
4. Brennan K., Resnick M. *New frameworks for studying and assessing the development of computational thinking: Paper presented at the Annual Meeting of the American Educational Research Association (AERA 2012).* – Vancouver, BC, Canada, 13-17 Apr. 2012. – 25 p.
5. Хеннер Е.К. *Вычислительное мышление // Образование и наука.* – 2016. – № 2 (131). – С. 18-33. – DOI: 10.17853/1994-5639-2016-2-18-33.
6. Фридман Л.М. *Теоретические основы методики обучения математике: учебное пособие.* – 2-е изд., испр. и доп. – М.: Изд-во УРСС, 2005. – 244 с. – ISBN 5-354-00883-2.
7. Knuth D.E. *The Art of Computer Programming. Vol. 1: Fundamental Algorithms.* – 3rd ed. – Boston: Addison-Wesley, 1997. – 652 p. – ISBN 978-0-201-89683-1.
8. Камалова Г.Б., Акжолова А.А. *Цифрлық білім берудегі есептеуіш ойлау ұғымы және оның дағдылары // Вестник КазНПУ им. Абая, серия «Физико-*

математические науки». – 2021. – № 4(76). – С. 168–173. – DOI:
10.51889/2021-4.1728-7901.23.

9. Каган Э.М. Обучение программированию как подход к развитию логического, абстрактного и вычислительного мышления у школьников // Вестник Российского университета дружбы народов. Серия: Информатизация образования. – 2017. – Т. 14, № 4. – С. 442–451. – DOI:
10.22363/2312-8631-2017-14-4-442-451.
10. Shute V.J., Sun C., Asbell-Clarke J. Demystifying computational thinking // Educational Research Review. – 2017. – Vol. 22. – P. 142–158. – DOI:
10.1016/j.edurev.2017.09.003.

ҚМ АА Куәлік нөмірі: **KZ45VPY00102718** — ҚР Мәдениет және Ақпарат министрлігі

© 2026 **Bilimger.kz** Ақпараттық-танымдық білім порталы. Барлық мазмұн авторлық құқықпен қорғалған.