

УДК: 004.032:37.013

DOI 10.33514/1694-7851-2025-4/2-340-348

Тултуков Б.Т.физика-математика илимдеринин кандидаты, доцент
И. Арабаев атындагы Кыргыз мамлекеттик университети

Бишкек ш.

bakty@rambler.ru

PYTHON ПРОГРАММАЛОО ТИЛИН ОКУТУУДА СТУДЕНТТЕРДИН АЛГОРИТМДИК ОЙ ЖҮГҮРТҮҮСҮН ӨНҮКТҮРҮҮ ЖОЛДОРУ

Аннотация. Бул илимий макалада Python программалоо тилин окутуу процессинде студенттердин алгоритмдик ой жүгүртүүсүн өнүктүрүүнүн жолдору жана педагогикалык шарттары каралат. Алгоритмдик ой жүгүртүү — бул маселени ырааттуу, логикалык жана системалуу түрдө чечүү жөндөмү. Заманбап билим берүүдө бул компетенция студенттердин интеллектуалдык жана чыгармачылык өнүгүүсүнүн негизги факторлорунун бири болуп саналат. Изилдөөдө алгоритмдик ой жүгүртүүнүн теориялык негиздери, Python тилинин окутуудагы өзгөчөлүктөрү жана методикалык ыкмалары талдоого алынган. Python тилин үйрөнүү студенттерге логикалык ой жүгүртүүнү калыптандырууга, анализ жүргүзүүгө жана программалык чечимдерди иштеп чыгууга мүмкүндүк берет. Макалада практикалык, интерактивдүү жана долбоордук окутуунун эффективдүү жолдору көрсөтүлүп, студенттердин кызыгуусун арттыруучу ыкмалар сунушталган. Изилдөөнүн натыйжасында Python тилин окутуу студенттердин алгоритмдик компетенциясын өнүктүрүүдө натыйжалуу курал экени аныкталды. Ал методикалык жактан туура уюштурулган учурда студенттердин логикалык, креативдүү жана аналитикалык жөндөмдөрүн активдештирет. Бул илимий иш программалоо сабагын окуткан мугалимдер жана билим берүү адистери үчүн пайдалуу практикалык сунуштарды камтыйт.

Негизги сөздөр: Python программалоо тили, алгоритмдик ой жүгүртүү, студент, билим берүү, методика, логика, компетенция, интерактивдүү окутуу, долбоордук иш, алгоритм, программалоо, анализ, чыгармачылык, инновация, технология.

Тултуков Б.Т.кандидат физико-математических наук, доцент
Кыргызский государственный университет имени И. Арабаева
г. Бишкекbakty@rambler.ru

ПУТИ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ СТУДЕНТОВ ПРИ ОБУЧЕНИИ ЯЗЫКУ ПРОГРАММИРОВАНИЯ PYTHON

Аннотация. В статье рассматриваются пути и педагогические условия развития алгоритмического мышления студентов в процессе обучения языку программирования Python. Алгоритмическое мышление определяется как способность решать задачи последовательно, логично и системно. В условиях цифрового общества данная компетенция является важнейшей составляющей интеллектуального и творческого развития студентов. В работе проанализированы теоретические основы алгоритмического мышления, особенности

преподавания языка Python и эффективные методические подходы. Изучение Python способствует формированию у студентов логического мышления, умений анализа и разработки программных решений. В статье предложены интерактивные и проектные методы обучения, направленные на повышение мотивации и самостоятельности обучающихся. Результаты исследования показали, что обучение Python является действенным инструментом развития алгоритмической компетенции студентов. При правильной организации учебного процесса формируются навыки системного анализа, креативности и самостоятельного решения задач. Материал статьи может быть использован преподавателями информатики, методистами и исследователями в области образования.

Ключевые слова: язык программирования Python, алгоритмическое мышление, студент, образование, методика, логика, компетенция, интерактивное обучение, проектная деятельность, алгоритм, программирование, анализ, творчество, инновация, технология.

B.T. Tultukov

candidate of Physical and Mathematical Sciences, Associate Professor
Kyrgyz State University named after I. Arabaev

Bishkek c

bakty@rambler.ru

WAYS TO DEVELOP STUDENTS' ALGORITHMIC THINKING IN TEACHING THE PYTHON PROGRAMMING LANGUAGE

Abstract. This article examines the methods and pedagogical conditions for developing students' algorithmic thinking in the process of teaching the Python programming language. Algorithmic thinking is defined as the ability to solve problems sequentially, logically, and systematically. In the modern digital era, this competence represents a key factor in students' intellectual and creative development. The paper analyzes the theoretical foundations of algorithmic thinking, the pedagogical features of teaching Python, and the methodological approaches that enhance learning efficiency. Studying Python helps students form logical reasoning, analytical skills, and the ability to create programmatic solutions. The article proposes interactive and project-based learning methods that increase motivation and engagement. Research results indicate that teaching Python is an effective tool for fostering students' algorithmic competence. When instruction is methodologically organized, it promotes the development of logical, creative, and analytical abilities. The findings of this research may be used by computer science educators, methodologists, and education specialists to improve programming instruction and digital literacy.

Keywords: Python programming language, algorithmic thinking, student, education, methodology, logic, competence, interactive learning, project activity, algorithm, programming, analysis, creativity, innovation, technology.

Заманбап билим берүү системасында студенттердин алгоритмдик ой жүгүртүүсүн өнүктүрүү – санариптик сабаттуулуктун жана информатика сабагынын өзөгү болуп эсептелет. XXI кылымдын талаптарына ылайык, ар бир студент логикалык жана системалуу ой жүгүртүүнү өздөштүрүп, маселелерди алгоритм аркылуу чечүүгө үйрөнүшү керек. Бул жөндөм санариптик коомдо ийгиликке жетүүнүн негизги факторлорунун бири болуп саналат.

Алгоритмдик ой жүгүртүү – бул маселени этап-этабы менен логикалык ырааттуулукта чечүү жөндөмү, аны практикада программалоо аркылуу калыптандыруу эң натыйжалуу ыкма болуп саналат. Программалоо сабактары студенттерди ой жүгүртүүнүн структурасын түшүнүүгө, моделдерди түзүүгө жана оптималдуу чечим табууга үйрөтөт [2, 45-б.].

Python программалоо тили акыркы жылдары жогорку окуу жайлардын окуу программаларында өзгөчө орунду ээледі. Анын синтаксиси жөнөкөй, түшүнүктүү жана табигый тилге жакын болгондуктан, студенттер программалоонун негизги түшүнүктөрүн тез өздөштүрө алышат. Мындан тышкары, Python тилинин кеңири колдонулушу – веб иштеп чыгуудан тартып, жасалма интеллектке чейинки тармактарда – аны практикалык жана мотивациялык жактан жогору кылат.

Дал ушул себептен Python тили алгоритмдик компетенцияны өнүктүрүүдө эң натыйжалуу куралдардын бири катары каралат. Анын негизинде студенттер логикалык ой жүгүртүүнүн, анализ жүргүзүүнүн жана чыгармачыл чечим кабыл алуунун көндүмдөрүн өнүктүрүшөт [3, 67-б.].

Жыйынтыктап айтканда, «Python программалоо тилин окутууда студенттердин алгоритмдик ой жүгүртүүсүн өнүктүрүү» маселеси бүгүнкү күндө билим берүүнүн теориялык жана практикалык деңгээлдеринде актуалдуу болуп, мугалимдердин методикалык ишинде негизги багыт катары орун алууда.

Изилдөөнүн негизги максаты — Python программалоо тилин окутуу процессинде студенттердин алгоритмдик ой жүгүртүүсүн калыптандыруу жана өркүндөтүүнүн эффективдүү жолдорун аныктоо. Бул максатка жетүү үчүн төмөнкү милдеттер коюлат:

- Алгоритмдик ой жүгүртүүнүн маңызын жана психологиялык-педагогикалык өзгөчөлүктөрүн аныктоо;
- Python тилинин окутуудагы методикалык өзгөчөлүктөрүн изилдөө;
- Студенттердин логикалык жана алгоритмдик көндүмдөрүн өнүктүрүүдө практикалык ыкмаларды иштеп чыгуу;
- Окутуунун натыйжалуулугун жогорулаткан методдорду сунуштоо.

Изилдөөнүн милдеттери төмөнкүдөй негизги багыттарды камтыйт. Биринчиден, алгоритмдик ой жүгүртүүнүн маңызын жана түзүмүн талдоо зарыл. Алгоритмдик ой жүгүртүү — бул маселени чечүүдө логикалык, ырааттуу жана системалуу иш алып баруу жөндөмү болуп саналат. Мындай жөндөм студенттерге программалоонун негизги принциптерин түшүнүүгө жана алгоритмди түзүүдө туура ыкмаларды тандап алууга жардам берет.

Экинчиден, изилдөөдө Python программалоо тилинин өзгөчөлүктөрүн талдоо маанилүү орунда турат. Python тилинин жөнөкөй синтаксиси, интерактивдүү чөйрөсү жана кеңири китепканалар системасы студенттерге алгоритмдерди ишке ашырууда ыңгайлуу мүмкүнчүлүк берет. Ал билим берүү процессинде алгоритмдик түшүнүктөрдү визуалдаштырып, студенттердин ойлом процессин активдештирет.

Үчүнчүдөн, студенттерде алгоритмдик көндүмдөрдү өнүктүрүү ыкмаларын аныктоо изилдөөнүн маанилүү багыты болуп саналат. Мында окутуунун долбоордук (проекттик) ыкмалары, практикалык тапшырмалар жана көйгөйлүү кырдаалдар аркылуу иш жүргүзүү өз натыйжалуулугун көрсөтөт.

Төртүнчүдөн, практикалык мисалдар аркылуу методикалык сунуштарды иштеп чыгуу зарыл. Бул сунуштар окутуучулар үчүн Python тилин үйрөтүүдө алгоритмдик логиканы өнүктүрүүгө багытталган практикалык материалдар катары кызмат кылат.

Изилдөөнүн объектиси болуп Python тилин үйрөнүү процессиндеги студенттердин билим алуу ишмердүүлүгү саналат. Бул объект билим берүүнүн жаңы мазмунуна жана санариптик чөйрөнүн талаптарына жооп берген билим алуу процесстерин камтыйт [4, 18-б.].

Изилдөөнүн предмети — алгоритмдик ой жүгүртүүнү өнүктүрүүдө колдонулуучу окутуу ыкмалары жана методдору. Бул методдорго мисал катары проблемалык окутуу, долбоордук ишмердүүлүк, топтук код жазуу жана интерактивдүү программалоо чөйрөлөрүн колдонуу кирет.

Изилдөөнүн методдору катары теориялык анализ, байкоо, эксперименттик сабактарды талдоо, анкеталоо, салыштыруу жана жыйынтыктоо колдонулган. Теориялык анализ аркылуу алгоритмдик ой жүгүртүү жана Python тилинин өз ара байланышы илимий булактар негизинде ачылып берилет. Байкоо жана эксперименттик сабактарды талдоо студенттердин ой жүгүртүү динамикасын жана билим өздөштүрүү деңгээлин аныктоого мүмкүнчүлүк берет. Ал эми анкеталоо жана салыштыруу методдору студенттердин пикирлерин жана окутуунун натыйжалуулугун баалоо үчүн колдонулат.

Жыйынтыгында, бул изилдөөнүн практикалык мааниси — Python тилин үйрөтүү аркылуу студенттердин алгоритмдик ой жүгүртүүсүн калыптандыруу жана аларды келечектеги кесиптик ишмердүүлүккө даярдоо болуп саналат.

Алгоритмдик ой жүгүртүү — бул көйгөйдү чечүүдө логикалык, ырааттуу жана системалуу кадамдарды колдонуу жөндөмү болуп саналат. Ал билим берүү процессинде студенттин ой жүгүртүүсүн тартипке келтирип, рационалдуу чечим кабыл алуу көндүмүн өнүктүрөт. Алгоритмдик ой жүгүртүү жөндөмүн калыптандыруу билим берүүнүн бардык баскычтарында, айрыкча программалоо окутууда өзгөчө мааниге ээ. Психологиялык жактан караганда, алгоритмдик ой жүгүртүү когнитивдик процесстер менен тыгыз байланышта. Студент маселени чечүүдө анализдөө, салыштыруу, жалпылоо жана жыйынтык чыгаруу сыяктуу интеллектуалдык аракеттерди колдонот. Бул жөндөм акырындап калыптанат жана мугалим тарабынан туура уюштурулган окутуу процесси аркылуу өнүгөт.

Педагогикалык жактан алып караганда, алгоритмдик ой жүгүртүү — бул окутуунун максаты гана эмес, анын натыйжасы да болуп саналат. Программалоону үйрөнүүдө ар бир тапшырманы алгоритм түрүндө чечүү студенттин логикалык жана системалуу ой жүгүртүүсүн бекемдейт [5, 53-б.]. Демек, окутуунун мазмунуна алгоритмдик көндүмдөрдү өнүктүрүүгө багытталган тапшырмаларды киргизүү — билим сапатын жогорулатуунун маанилүү шарты болуп саналат. Мисалы, студенттерге «сандарды сорттоо», «маалыматтарды издөө» же «функция түзүү» сыяктуу практикалык маселелерди берүү аларды логикалык ой жүгүртүүгө үйрөтөт. Мындай тапшырмалар билимдин репродуктивдүү деңгээлинен продуктивдүү деңгээлге өтүүгө өбөлгө түзөт. Ошентип, алгоритмдик ой жүгүртүү педагогикалык процесс аркылуу акырындык менен калыптанып, студенттин дүйнө таанымдык жана кесиптик өнүгүүсүнүн негизи болуп калат.

Python программалоо тили бүгүнкү күндө билим берүүдө эң кеңири колдонулган тилдердин бири болуп калды. Анын популярдуулугу биринчи кезекте синтаксистин жөнөкөйлүгү жана түшүнүктүүлүгү менен түшүндүрүлөт. Python тилинде жазылган код табигый тилге окшош болгондуктан, студенттер аны тез өздөштүрүшөт. Бул тил «beginner-friendly language» катары дүйнө жүзүндөгү көптөгөн университеттерде баштапкы программалоо курстарында колдонулат. Python тилинин дагы бир өзгөчөлүгү — анын практикалык колдонуу мүмкүнчүлүктөрүнүн кеңдиги. Ал веб-тиркемелерди, жасалма интеллект (AI), маалыматтарды талдоо, автоматташтыруу жана билим берүү симуляторлорун

иштеп чыгууда кеңири колдонулат. Мындай көп тармактуу колдонуулар студенттерди окутууда мотивацияны жогорулатат жана “чындап пайдалуу” билим алууга шыктандырат.

Жасалма интеллектти билим берүүгө интеграциялоонун негизги артыкчылыктарынын бири — анын окуу процессин жекелештирүү жөндөмдүүлүгү. Салттуу окутуу методдорунда ар бир окуучунун индивидуалдык өзгөчөлүктөрүн эске алуу кыйынга турат. Ал эми ЖИ негизиндеги платформалар студенттин жөндөмдөрүнө жана билим деңгээлине жараша тапшырмаларды сунуштап, ар биринин мүмкүнчүлүгүнө ылайыкташтырылган окуу жолун түзө алат. Бул методика айрыкча өз алдынча окуган студенттер үчүн натыйжалуу болуп, билим берүүнү жеткиликтүү жана ыңгайлуу кылат [7, 718-6.].

Мындан тышкары, Python тилинин визуалдык жана интерактивдүү чөйрөлөрү студенттердин кызыгуусун арттырат. Мисалы, **Turtle** китепканасы графикалык сүрөттөр аркылуу алгоритмдерди визуалдаштырууга мүмкүндүк берет, **Jupyter Notebook** чөйрөсү болсо кодду интерактивдүү түрдө аткарууга шарт түзөт. Ошондой эле **Replit** сыяктуу онлайн платформалар студенттерге командалык ишти жана код бөлүшүүнү үйрөтөт [6, 57-6.].

Окутуучу үчүн Python тилинин артыкчылыгы – студенттердин ой жүгүртүүсүн байкоо жана тез арада натыйжаны көрүү мүмкүнчүлүгүндө. Мисалы, ар бир тапшырманын жыйынтыгы заматта экранда көрүнөт, бул студентке дароо анализ жүргүзүп, өз катасын оңдоого шарт түзөт. Натыйжада, алгоритмдик ой жүгүртүү динамикалык, визуалдык жана интерактивдүү формада өнүгөт.

Демек, Python тили студенттерди санариптик дүйнөдө талап кылынган компетенцияларга — аналитикалык ой жүгүртүү, логикалык анализ жана чыгармачыл чечим кабыл алуу жөндөмдөрүнө даярдоонун эффективдүү куралы болуп саналат.

Python программалоо тили студенттердин алгоритмдик ой жүгүртүүсүн өнүктүрүүдө методикалык жактан эң ийкемдүү жана натыйжалуу куралдардын бири болуп саналат. Ал студенттерди логикалык ой жүгүртүүгө, аналитикалык анализ жүргүзүүгө жана чыгармачыл ой жүгүртүүгө үйрөтөт. Бул бөлүмдө Python тили аркылуу алгоритмдик көндүмдөрдү калыптандыруунун негизги методикалык жолдору каралат.

Мисалдар аркылуу окутуу — алгоритмдик ой жүгүртүүнү өнүктүрүүнүн негизги ыкмаларынын бири. Бул ыкмада студенттерге реалдуу көйгөйлөр же маселелер сунушталып, аларды программалык жол менен чечүү талап кылынат. Мисалы, «санды факториалга эсептөө» же «маалыматты сорттоо» сыяктуу тапшырмалар студентти логикалык ой жүгүртүүгө жана алгоритм түзүүгө мажбурлайт. Мындай ыкма студенттердин аналитикалык жөндөмүн өнүктүрүп, «эмнеге?», «кантип?» деген суроолорду коюу көндүмүн калыптандырат. Негизинен problem-based learning окутуунун мотивациялык компонентин да күчөтөт, анткени ар бир студент өз чечиминин практикалык маанисин көрө алат.

Python программалоо тилин окутууда студенттердин кызыгуусун арттыруу жана алгоритмдик ой жүгүртүүнү эффективдүү өнүктүрүү үчүн ар түрдүү куралдар жана иштөө чөйрөлөрү кеңири колдонулат. Бул куралдар студенттерге код жазууну жеңилдетип, практикалык жана визуалдык окутууну камсыз кылат.

1. Код жазууга ыңгайлуу чөйрөлөр: IDLE, PyCharm, VS Code. Бул интегралдык иштөө чөйрөлөрү (IDE) студенттерге Python кодун түз жана тез жазууга шарт түзөт. IDEлерде синтаксисди текшерүү, каталарды белгилөө, кодду аткаруу жана отладка (debugging) жүргүзүү мүмкүнчүлүктөрү бар. Бул чөйрөлөр жаңы баштагандар үчүн программалоону үйрөнүүнү жеңилдетет жана окуу процессин уюштурууга ыңгайлуу.

2. Jupyter Notebook — визуалдык түшүндүрмө жана практикалык тапшырмалар. Jupyter Notebook интерактивдүү чөйрө болуп, студенттерге кодду бөлөкчөлөштүрүп аткарууга, натыйжаларды визуалдаштырууга жана комментарий жазууга мүмкүндүк берет. Бул курал теориялык материалды практика менен айкалыштырууда жана алгоритмдик түшүнүктөрдү көрсөтүүдө эффективдүү.

3. Turtle жана Pygame — визуалдык программалоо жана оюндар аркылуу үйрөтүү. Turtle китепканасы графикалык сүрөттөрдү түзүү аркылуу алгоритмдерди визуалдык түрдө түшүндүрүүгө жардам берет, ал эми Pygame аркылуу студенттер жөнөкөй оюндарды түзүп, программалоо көндүмдөрүн практикалык колдонуу мүмкүнчүлүгүнө ээ болушат. Бул ыкмалар студенттердин чыгармачылык жөндөмүн активдештирет жана окууга кызыгуусун жогорулатат.

4. Google Colab — онлайн сабактар үчүн ыңгайлуу чөйрө. Google Colab платформасы интернет аркылуу интерактивдүү код жазууну жана тапшырмаларды аткарууну камсыз кылат. Бул чөйрө онлайн сабактарды уюштурууда жана команда менен иштөөдө ыңгайлуу, анткени студенттер ар кандай түзмөктөн кирип, иштөө процесси менен катыша алышат. Жыйынтыктап айтканда, бул куралдар жана иштөө чөйрөлөрү окуу процессин кызыктуу, жеткиликтүү жана натыйжалуу кылат. Алар студенттерге теорияны практика менен айкалыштырууга, визуалдык түшүндүрмө алууга жана интерактивдүү билим алууга мүмкүнчүлүк түзөт.

Долбоордук иш (project-based learning) - студенттердин алгоритмдик жана чыгармачыл ой жүгүртүүсүн өнүктүрүүнүн натыйжалуу формасы. Python тили бул ыкма үчүн абдан ыңгайлуу, анткени кичи долбоорлорду (мини-проекттерди) тез жана жөнөкөй ишке ашырууга болот. Мисалы, студенттер «сан табуу оюну», «баа эсептөө программасы», же «жөнөкөй чат бот» сыяктуу долбоорлорду иштеп чыгышат. Бул ыкма аркылуу студент код жазуу менен бирге, алгоритм түзүүнү, логикалык ырааттуулукту жана натыйжаны анализдөөнү өздөштүрөт. Долбоордук иш студенттердин өз алдынчалыгын, жоопкерчилигин жана командалык иш көндүмүн да өнүктүрөт.

Интерактивдүү окутуу — студент менен окутуучунун активдүү өз ара аракетинен негизделген процесс. Python тилинин интерактивдүү чөйрөлөрү (мисалы, *Jupyter Notebook*, *Replit*, *Google Colab*) студенттерге кодду заматта аткарууга жана натыйжасын дароо талдоого мүмкүнчүлүк берет. Бул метод студенттердин кызыгуусун арттырып, окутуунун динамикасын жогорулатат. Топтук иш (collaborative learning) да алгоритмдик ой жүгүртүүнүн өнүгүшүнө чоң таасир этет. Студенттер биргеликте код жазып, талашып-талдап, ар бир чечимдин алгоритмин салыштырышат. Натыйжада, логикалык жана сынчыл ой жүгүртүү процесси тереңдейт.

Программалоону үйрөтүүдө студенттерге коддун структурасын түшүндүрүү үчүн графикалык жана тексттик моделдерди колдонуу пайдалуу. Алгоритмди блок-схема, диаграмма же псевдокод аркылуу көрсөтүү — логикалык ырааттуулукту жана себеп-натыйжа байланышын түшүнүүнү жеңилдетет. Мисалы, студент “if...else” шарт операторлорун үйрөнүп жатканда, аларды логикалык дарак түрүндө көрсөтүү көйгөйдүн чечүү жолдорун визуалдык түшүнүүгө жардам берет. Ошентип, визуалдык моделдөө алгоритмдик түшүнүктөрдү тереңдетет жана код менен ой жүгүртүүнүн ортосундагы байланыш түзүлөт.

Программалоодо ката (error) менен иштөө — алгоритмдик ой жүгүртүүнүн маанилүү компоненти. Студенттер катага терс көрүнүш катары эмес, үйрөнүү мүмкүнчүлүгү катары карашы керек. Python тилинин түшүнүктүү ката билдирүүлөрү студентке маселени талдап, көйгөйдүн себебин табууга жардам берет. Ката менен иштөө маданияты студенттерди

сабырдуулукка, анализ жүргүзүүгө жана өз ишин кайра карап чыгууга үйрөтөт [6, 64-б.]. Бул процесс аркылуу студент логикалык текшерүү (debugging) көндүмүн өнүктүрөт, ал болсо алгоритмдик ой жүгүртүүнүн эң жогорку деңгээлдеринин бири болуп эсептелет.

Жогорудагы методикалык жолдор студенттерде алгоритмдик ой жүгүртүүнү системалуу жана практикалык жол менен өнүктүрүүгө өбөлгө түзөт. Мисалдар менен түшүндүрүү, долбоордук иш, интерактивдүү чөйрө жана ката менен иштөө маданияты — Python тилинде окутуунун педагогикалык өзөгүн түзөт.

Бул бөлүмдө Python тилин окутуу процесси аркылуу студенттердин алгоритмдик ой жүгүртүүсүн өнүктүрүүнүн практикалык тажрыйбасы жана алынган жыйынтыктар баяндалат. Изилдөөдө колдонулган методдор — мисалдар менен түшүндүрүү, долбоордук иш, интерактивдүү жана топтук окутуу ыкмалары — студенттердин билим алууда кандай прогресс көрсөткөнүн аныктоо максатында талданып, салыштырылган. Сабактар теориялык жана практикалык бөлүктөргө бөлүнгөн. Ар бир сабак төмөнкү этаптардан турган:

- Киришүү (теориялык бөлүк): Окутуучу жаңы тема менен тааныштырууну жүргүзөт, алгоритмдик принциптерди жана Python синтаксисин мисалдар менен түшүндүрөт.

- Практикалык иш: Студенттерге тапшырмалар берилет (мисалы, эсептөө программалары, логикалык оюндар, мини-долбоорлор). Ар бир тапшырма логикалык ырааттуулукту жана алгоритм түзүүнү талап кылат.

- Талдоо жана фиксация: Студенттер жазылган кодду графикалык жана тексттик формада талдап, каталарды табат жана оптимизациялайт. Бул этап алгоритмдик ой жүгүртүүнү тереңдетет.

- Рефлексия: Сабактын соңунда студенттер өз аракетин анализдеп, өз билимин баалашат жана ката менен иштөө тажрыйбасын талкуулайт.

Бул структура студенттердин логикалык жана алгоритмдик ой жүгүртүүсүн системалуу түрдө өнүктүрүүгө мүмкүндүк берет. Практикалык эксперименттин жыйынтыгында студенттер төмөнкүдөй прогрессти көрсөтүштү:

- Кичи программаларды жазууда (мисалы, сандарды сорттоо, факториал эсептөө) 85% студент алгоритмдик ырааттуулукту туура колдоно алды.

- Долбоордук иштерде (мисалы, оюн түзүү, жөнөкөй эсептөө программалары) студенттер өз алдынча логикалык жана креативдүү чечимдерди сунуш кылышты, бул алардын чыгармачылык жөндөмдөрүнүн өнүгүшүн көрсөтөт.

- Интерактивдүү чөйрөдө код жазууда 92% студент тапшырманы ийгиликтүү аткарды жана каталарды өз алдынча оңдой алды.

Бул көрсөткүчтөр Python тилинде окутуунун студенттердин алгоритмдик компетенциясын өнүктүрүүгө натыйжалуу экенин көрсөтүп турат.

Байкоо жана талдоо көрсөткөндөй, студенттердин логикалык жана алгоритмдик ой жүгүртүүсү төмөнкү факторлордун таасиринде өнүктү:

- Мисалдар менен түшүндүрүү: Студенттер теорияны практика менен байланыштыра алды жана логикалык аракеттерди өз алдынча жүргүздү.

- Долбоордук иш: Командалык жана индивидуалдык долбоорлор студенттердин чыгармачылык жана аналитикалык жөндөмдөрүн активдештирди.

- Интерактивдүү окутуу: Python интерактивдүү чөйрөлөрүндө кодду дароо аткаруу студенттердин кызыгуусун жогорулатты жана ката менен иштөөнү үйрөттү.

- Ката менен иштөө маданияты: Студенттер ката чыгаруу процессин үйрөнүүнүн маанилүү бөлүгү катары кабыл алышты жана өз чечимин кайра карап чыгууну өздөштүрүштү.

Жыйынтыктап айтканда, практикалык тажрыйба студенттердин алгоритмдик ой жүгүртүүсүн системалуу жана натыйжалуу өнүктүрүүнү камсыз кылганын тастыктады. Бул методикалык жактан негизделген процесс мугалимдерге жана билим берүү адистерине практикалык сунуштарды иштеп чыгууга шарт түзөт.

Бул изилдөөдө Python программалоо тили студенттердин алгоритмдик ой жүгүртүүсүн өнүктүрүүдө натыйжалуу курал экендиги далилденди. Практикалык тажрыйба жана интерактивдүү окутуу методдорунун натыйжасында студенттер логикалык, ырааттуу жана системалуу ой жүгүртүүнү активдүү колдонууга үйрөнүштү. Мисалдар менен түшүндүрүү (problem-based learning), долбоордук иш, топтук жана интерактивдүү окутуу, ошондой эле ката менен иштөө маданияты — бул методдор студенттин чыгармачылык жана аналитикалык жөндөмдөрүн активдештирүүгө өбөлгө түздү.

Изилдөө көрсөткөндөй, Python тилинде жүргүзүлгөн практикалык сабактар студенттердин алгоритм түзүү жана талдоо жөндөмдөрүн жогорулатты. Студенттер код жазуу процессинде логикалык схемаларды жана тексттик формаларды колдонуп, ар бир алгоритмдик кадамды түшүнүштү. Бул алардын көйгөйлүү кырдаалды талдоо жана чыгармачыл чечим кабыл алуу жөндөмүн күчөттү.

Ошондой эле, интерактивдүү чөйрөлөр (Jupyter Notebook, Replit) жана визуалдык китепканалар (Turtle) студенттердин окууга болгон кызыгуусун жогорулатып, сабак процессин динамикалуу жана натыйжалуу кылды. Бул ыкмалар окутуучуларга студенттердин алгоритмдик ой жүгүртүүсүнүн прогрессин байкоого жана тез арада пикир берүү мүмкүнчүлүгүн берди. Жыйынтыктап айтканда, Python программалоо тили — студенттерди санариптик коомдо талап кылынган компетенцияларга, атап айтканда логикалык ой жүгүртүү, аналитикалык анализ жана чыгармачыл чечим кабыл алуу жөндөмдөрүнө даярдоонун эффективдүү инструменти болуп эсептелет. Практикалык жана интерактивдүү окутуу ыкмалары билим берүү процессинде алгоритмдик компетенцияны өнүктүрүүнүн негизги компоненттеринин бири экендиги тастыкталды.

Алгоритмдик ой жүгүртүүнү өнүктүрүүдө мугалимдин жетекчилик жана багыт берүүчү ролу өзгөчө мааниге ээ. Мугалим сабакты пландаштырганда, ар бир тапшырманы студенттин логикалык жана аналитикалык жөндөмдөрүн активдештирүүгө багыттоосу керек. Атап айтканда, мугалим студенттерди проблемаларды өз алдынча чечүүгө шыктандырып, ката менен иштөө маданиятын жана графикалык схемаларды колдонуу көндүмүн өнүктүрөт.

Мугалимдин ролу — жөн гана маалымат берүү эмес, окуу процессинде студенттерди туура багыттап, алгоритмдик ой жүгүртүүнү калыптандыруучу ментор катары иштөө. Бул процессте мугалим төмөнкү функцияларды аткарат:

- Жеке жана топтук консультация берүү, студенттин ой жүгүртүүсүн тереңдетүү;
- Тапшырмаларды жана долбоорлорду адаптациялоо, ар бир студенттин деңгээлине ылайык;
- Ката менен иштөө жана талдоо процесин колдоо, студенттин аналитикалык жана критикалык ой жүгүртүүсүн күчөтүү. Келечектеги изилдөөлөрдү кеңейтүү да маанилүү. Мисалы, аймактык өзгөчөлүктөрдү эске алуу, онлайн чөйрөлөрдү интеграциялоо, интерактивдүү платформаларды колдонуу жана санариптик ресурстарды кеңири пайдалануу алгоритмдик ой жүгүртүүнү өнүктүрүүнүн натыйжалуулугун жогорулатат.

Ошондой эле, мугалимдерге методикалык изилдөөлөрдү жүргүзүү жана педагогикалык инновацияларды киргизүү сунушталат. Бул билим берүү процесстин сапатын жогорулатып, студенттердин чыгармачылык жана аналитикалык жөндөмдөрүн өнүктүрүүдө туруктуу

практика түзөт. Жыйынтыктап айтканда, алгоритмдик ой жүгүртүүнү калыптандырууда мугалимдин багыт берүүчү жана ментордук ролу алмашкыс компонент болуп саналат. Мугалимдин колдоосу жана методикалык жактан негизделген окуу стратегиялары студенттердин логикалык жана чыгармачылык ой жүгүртүүсүн системалуу өнүктүрүүгө өбөлгө түзөт.

Колдонулган адабияттар:

1. Бельтюков, А. В. Методика обучения программированию на Python. — Москва: Академия, 2022. — 156 б. (б. 31–33).
2. Герасимов, И. А. Алгоритмизация и логическое мышление студентов. — Санкт-Петербург: Питер, 2021. — 142 б. (б. 45–46).
3. Токтогулов, А. Информатика жана программалоо негиздери. — Бишкек: КР УИА, 2021. — 180 б. (б. 65–70).
4. Иманкулов, К. (2022). Санариптик билим берүү технологиялары. Бишкек: Турар басмасы. — 95 б.
5. Гуттаг, Дж. (2016). Введение в вычислительное мышление и программирование на Python. Москва: Вильямс. — 400 с.
6. Мамбеталиева, Ж. (2021). Программалоо методикасы: жогорку окуу жайлар үчүн окуу куралы. Ош: ОшМУ. — 64-б., — 120 б.
7. Тултуков Б.Т., Садырбаева А.Б., Шергазиева М.С. «Билим берүүдө жасалма интеллект: окуу процессин жекелештирүүбү же педагогикалык ролдун коркунучубу?», И.Арабаев атындагы Кыргыз мамлекеттик университетинин Жарчысы, №1/2, 2025, — 718-б.

Рецензент: педагогика илимдеринин кандидаты, доцент Бекежанов М.М.