O‘ZBEKISTON MILLIY UNIVERSITETI JIZZAX FILIALI

ISSN 2181-3884

ZAMONAVIY ILMIY-TEXNIK TADQIQOTLAR

xalqaro jurnali

Vol. 2 No. 4

2026

International Journal of Contemporary Scientific and Technical Research

Volume 2 Issue 4
2026

Zamonaviy ilmiy-texnik tadqiqotlar xalqaro jurnali

4-son
2026

Международный журнал современных научно-технических исследований

Том 2 Выпуск 4
2026

MIRZO ULUG‘BEK NOMIDAGI O‘ZBEKISTON MILLIY UNIVERSITETINING JIZZAX FILIALI

Xalqaro ilmiy-elektron jurnali

Tahrir kengashi raisi:

P.f.d., prof., O.X. Turakulov

Bosh muharrir:

A.A. Abdumalikov

Tahrir kengashi mas’ul kotibi:

D.M. Jomurodov

Tahririyat hay’ati:

1. B.f.d., prof., S. Murodova (O’zMU Jizzax filiali, UZB)
2. P.f.n., prof., S. Usmanov (O’zMU Jizzax filiali, UZB)
3. T.f.n., dos., S. Tavboev (O’zMU Jizzax filiali, UZB)
4. F.f.d., prof., N. Djusupov (O’zDJTU, UZB)
5. T.f.d., dos., S. Ruzimov (TTPU, UZB)
6. P.f.d., prof., G. Lekerova (SKU, KZH)
7. DSc., Ali Coşkan (ASUI, TR)
8. T.f.d., prof., D. Irgasheva (TATU, UZB)
9. T.f.d., prof., I. Siddikov (NRU TIIAME, UZB)
10. T.f.n., dos., S. Eshkabilov (NDU, US)

Zamonaviy ilmiy-texnik tadqiqotlar xalqaro elektron jurnali (International Journal of Contemporary Scientific and Technical Research) O’zMU Jizzax filiali Kengashining qaroriga asosan tashkil etilib, 2022-yil 9-sentabrda O‘zbekiston Respublikasi Prezidenti Administratsiyasi huzuridagi Axborot va ommaviy kommunikatsiyalar agentligi tomonidan №038200 raqami bilan ro‘yxatidan o‘tkazilgan, shuningdek davlatlararo standartlar talabi asosida O‘zbekiston Milliy kutubxonasidan jurnal uchun 2181-3884 ISSN raqami olingan.

© “O‘zMU Jizzax filiali” – 2023

Tahririyat manzili:

130100, Jizzax viloyati, Jizzax shahri,
Sh.Rashidov Shox ko‘chasi, 259-uy.

DASTURLASHDA OPERATORLARDAN FOYDALANISH KO‘NIKMALARINI TAHLIL QILISH

Xodjaeva Nodira Sharifovna

Toshkent davlat transport universiteti, o‘qituvchi.

Annotatsiya.

Mazkur maqolada oliy ta’lim muassasalarida dasturlash fanini o‘qitish jarayonida talabalarning operatorlardan foydalanish ko‘nikmalarini shakllantirish va rivojlantirishning pedagogik hamda metodik asoslari tahlil qilingan. Tadqiqotning maqsadi dasturlash tillaridagi arifmetik, mantiqiy, taqqoslash va o‘zlashtirish operatorlari, shuningdek, shartli va takrorlanuvchi boshqaruv konstruksiyalarini o‘zlashtirish jarayonida yuzaga keladigan didaktik muammolarni aniqlash hamda ularni bartaraf etishga qaratilgan samarali o‘qitish usullarini asoslashdan iborat. Tadqiqot doirasida kompetensiyaviy, faoliyatga yo‘naltirilgan va muammoli ta’lim yondashuvlarining dasturlash ko‘nikmalarini rivojlantirishdagi imkoniyatlari ko‘rib chiqiladi. Python va C++ dasturlash tillari misolida operatorlardan foydalanishga oid amaliy topshiriqlarni tashkil etish, algoritmik masalalarni bosqichma-bosqich murakkablashtirish, dasturiy koddagi xatolarni aniqlash va tuzatish orqali talabalarning mustaqil fikrlash faoliyatini rivojlantirish yo‘llari yoritiladi. Talabalarning operatorlardan foydalanish kompetensiyasini baholash uchun nazariy bilimlarni amaliyotda qo‘llash, algoritmning mantiqiy to‘g‘riligini ta’minlash, dastur natijasini tahlil qilish va muammoli vaziyatlarda mustaqil yechim ishlab chiqish kabi mezonlar taklif etiladi. Maqolada bayon etilgan metodik tavsiyalar dasturlash fanini o‘qitish mazmunini takomillashtirish, talabalarning algoritmik tafakkuri va amaliy dasturlash kompetensiyalarini rivojlantirishga xizmat qilishi mumkin.

Kalit so‘zlar: dasturlashni o‘qitish metodikasi, pedagogik texnologiyalar, dasturlash operatorlari, algoritmik tafakkur, kompetensiyaviy yondashuv, muammoli ta’lim, amaliy ko‘nikmalar, interfaol ta’lim, mantiqiy fikrlash, dasturlash kompetensiyasi, mustaqil ta’lim, baholash mezonlari.

АНАЛИЗ НАВЫКОВ ИСПОЛЬЗОВАНИЯ ОПЕРАТОРОВ В ПРОГРАММИРОВАНИИ

Ходжаева Нодира Шарифовна

Ташкентский государственный транспортный университет, преподаватель.

Аннотация.

В данной статье анализируются педагогические и методические основы формирования и развития у студентов навыков использования операторов в процессе преподавания программирования в высших учебных заведениях. Цель исследования заключается в выявлении дидактических проблем, возникающих при освоении арифметических, логических операторов, операторов сравнения и присваивания, а также условных конструкций и конструкций управления циклами в языках программирования, и обосновании эффективных методов обучения, направленных на их преодоление. В рамках исследования рассматриваются возможности компетентностного, деятельностного и проблемно-ориентированного подходов к развитию навыков программирования. На примере языков программирования Python и C++ освещаются способы развития самостоятельного мышления студентов посредством организации практических заданий по использованию операторов, поэтапного усложнения алгоритмических задач, выявления и исправления ошибок в программном коде. Для оценки компетентности студентов в использовании операторов предлагаются такие критерии, как применение теоретических знаний на практике, обеспечение логической корректности алгоритма, анализ результатов выполнения программы и самостоятельная разработка решений в проблемных ситуациях. Представленные в статье методические рекомендации могут способствовать совершенствованию содержания обучения программированию, развитию алгоритмического мышления студентов и их практических компетенций в области

программирования.

Ключевые слова: методика обучения программированию, педагогические технологии, операторы программирования, алгоритмическое мышление, компетентностный подход, проблемное обучение, практические навыки, интерактивное обучение, логическое мышление, компетенция в программировании, самостоятельное обучение, критерии оценивания.

ANALYSIS OF SKILLS IN USING OPERATORS IN PROGRAMMING

Xodjaeva Nodira Sharifovna

Tashkent State Transport University, Lecturer.

Abstract.

This article analyzes the pedagogical and methodological foundations for developing students' skills in using operators in the process of teaching programming at higher education institutions. The aim of the study is to identify didactic challenges arising in the acquisition of arithmetic, logical, comparison, and assignment operators, as well as conditional and iterative control structures in programming languages, and to substantiate effective teaching methods aimed at overcoming these challenges. The study examines the potential of competency-based, activity-oriented, and problem-based learning approaches for developing programming skills. Using Python and C++ as examples, the article explores ways to develop students' independent thinking through the organization of practical tasks involving operators, the gradual increase in the complexity of algorithmic problems, and the identification and correction of errors in program code. To assess students' competence in using operators, the study proposes criteria such as applying theoretical knowledge in practice, ensuring the logical correctness of algorithms, analyzing program execution results, and independently developing solutions in problem situations. The methodological recommendations presented in the article may contribute to improving the content of programming instruction and developing students' algorithmic thinking and practical programming competencies.

Keywords: programming teaching methodology, pedagogical technologies, programming operators, algorithmic thinking, competency-based approach, problem-based learning, practical skills, interactive learning, logical thinking, programming competence, independent learning, assessment criteria.

Kirish. Bugungi kunda jahon miqyosida raqamli texnologiyalarning jadal rivojlanishi, sun'iy intellekt, katta hajmdagi ma'lumotlarni qayta ishlash, avtomatlashtirilgan boshqaruv tizimlari va zamonaviy dasturiy ta'minot vositalarining turli sohalarga keng joriy etilishi ta'lim tizimi oldiga yangi pedagogik vazifalarni qo'ymoqda. Xususan, oliy ta'lim muassasalarida raqamli kompetensiyalarga ega, algoritmik va mantiqiy fikrlay oladigan, muammoli vaziyatlarda mustaqil qaror qabul qiladigan hamda zamonaviy dasturlash texnologiyalaridan samarali foydalana oladigan mutaxassislarni tayyorlash dolzarb masalalardan biriga aylanmoqda. Mazkur jarayonda dasturlash fanlarini o'qitish mazmunini takomillashtirish, nazariy bilimlarni amaliy faoliyat bilan integratsiyalash hamda talabalarning dasturiy masalalarni mustaqil hal qilish ko'nikmalarini rivojlantirish alohida ilmiy-pedagogik ahamiyat kasb etadi.

Dasturlash zamonaviy axborot texnologiyalarining fundamental tarkibiy qismlaridan biri bo'lib, u nafaqat dasturiy mahsulotlar yaratish, balki algoritmik tafakkur, tizimli tahlil, mantiqiy xulosa chiqarish va muammolarni bosqichma-bosqich hal etish kabi intellektual faoliyat turlarini rivojlantirishga xizmat qiladi. Dasturlashni o'zlashtirish jarayonida talaba muayyan masalaning shartini tahlil qiladi, kirish va chiqish ma'lumotlari o'rtasidagi bog'liqlikni aniqlaydi, algoritmik yechim ishlab chiqadi hamda ushbu yechimni dasturlash tili vositalari yordamida amalga oshiradi. Bunday faoliyat talabalardan dasturlash tilining sintaktik qoidalarini bilish bilan bir qatorda, uning semantik xususiyatlarini tushunish, dastur bajarilishining mantiqiy ketma-ketligini anglash va olingan natijalarni baholashni ham talab etadi.

Dasturlash tillarining asosiy tarkibiy elementlari hisoblangan operatorlar dasturiy algoritmlarni ifodalash va amalga oshirishda muhim o'rin egallaydi. Arifmetik, mantiqiy, taqqoslash va o'zlashtirish operatorlari ma'lumotlar ustida turli amallarni bajarish, ifodalarni hisoblash hamda mantiqiy shartlarni shakllantirish imkonini beradi. Shartli va takrorlanuvchi boshqaruv konstruksiyalari esa algoritmning bajarilish yo'nalishini belgilash, muayyan shartlarga asoslangan qarorlarni amalga oshirish hamda takroriy hisoblash jarayonlarini tashkil etishda qo'llaniladi. Shu jihatdan operatorlardan to'g'ri va maqsadga muvofiq foydalanish ko'nikmalarining shakllanganligi talabalarning dasturlash bo'yicha amaliy kompetensiyalarini tavsiflovchi muhim ko'rsatkichlardan biri hisoblanadi.

Dasturlash operatorlarini o'rganish jarayoni faqat ularning yozilish shakli va bajaradigan vazifalarini eslab qolish

bilan cheklanmaydi. Mazkur jarayon operatorlarning o‘zaro bog‘liqligini tushunish, ularning ustuvorlik darajasini hisobga olish, murakkab mantiqiy ifodalarni tuzish, shartli konstruksiyalar yordamida algoritmik tarmoqlanishlarni tashkil etish va takrorlash jarayonlarini boshqarish kabi amaliy faoliyatlarni ham qamrab oladi. Ayniqsa, bir nechta operator ishtirok etgan ifodalarning bajarilish natijasini oldindan aniqlash, turli algoritmik yechimlarni taqqoslash hamda dasturiy koddagi mantiqiy xatolarni mustaqil aniqlash talabalardan yuqori darajadagi tahliliy va algoritmik fikrlashni talab qiladi.

Operatorlardan foydalanishdagi xatolarni pedagogik jihatdan tahlil qilish ularning yuzaga kelish sabablarini aniqlash va tegishli metodik yechimlarni ishlab chiqish uchun muhimdir. Masalan, C++ dasturlash tilida o‘zlashtirish (=) va tenglikni tekshirish (==) operatorlarini chalkashtirish, mantiqiy && va || operatorlarining qo‘llanish shartlarini noto‘g‘ri talqin qilish, arifmetik ifodalarda operatorlar ustuvorligini hisobga olmaslik hamda takrorlash konstruksiyalarida to‘xtash shartini noto‘g‘ri belgilash dastur bajarilishida kutilmagan natijalarni yuzaga keltirishi mumkin. Python dasturlash tilida esa and, or, not mantiqiy operatorlari, if, elif, else shartli konstruksiyalari hamda for va while sikllarining o‘zaro funksional bog‘liqligini tushunish alohida ahamiyatga ega. Bunday holatlarni o‘quv jarayonida tahlil qilish talabalarning xatolarni aniqlash, ularning sabablarini tushuntirish va to‘g‘ri algoritmik yechimlarni ishlab chiqish ko‘nikmalarini rivojlantirishga xizmat qiladi.

Pedagogik nuqtayi nazardan dasturlash operatorlarini o‘rgatishda kompetensiyaviy yondashuvdan foydalanish muhim ahamiyat kasb etadi. Mazkur yondashuv ta’lim natijalarini faqat o‘zlashtirilgan nazariy bilimlar hajmi bilan emas, balki ushbu bilimlardan amaliy vaziyatlarda foydalanish, muammolarni mustaqil hal qilish va faoliyat natijalarini baholash qobiliyati orqali aniqlashni nazarda tutadi. Dasturlash ta’limida kompetensiyaviy yondashuv operatorlarning vazifalarini tushuntirish, ularga oid mashqlarni bajarish, algoritmik masalalarni mustaqil yechish, koddagi xatolarni aniqlash va dastur natijalarini tahlil qilish kabi o‘zaro bog‘liq faoliyatlar orqali amalga oshirilishi mumkin. Bunda talabalar dasturlash tilining alohida elementlarini o‘zlashtirishdan ularni yaxlit algoritmik yechim tarkibida qo‘llash bosqichiga o‘tadilar.

Shuningdek, dasturlash operatorlarini o‘qitishda muammoli ta’lim, faoliyatga yo‘naltirilgan ta’lim, differensial yondashuv va interfaol metodlardan foydalanish muhim metodik imkoniyatlarga ega. Muammoli ta’lim jarayonida talabaga tayyor yechimni taqdim etish o‘rniga, muayyan dasturiy vaziyatni tahlil qilish va maqbul operatorlarni mustaqil tanlash vazifasi yuklatiladi. Faoliyatga yo‘naltirilgan yondashuvda nazariy tushunchalar amaliy dasturlash topshiriqlari bilan mustahkamlanadi. Differensial yondashuv esa talabalarning dastlabki bilim darajasi va individual o‘zlashtirish xususiyatlarini hisobga olgan holda turli murakkablikdagi topshiriqlarni tashkil etish imkonini beradi. Interfaol metodlar orqali kodni birgalikda tahlil qilish, dastur natijalarini muhokama qilish va muqobil algoritmik yechimlarni solishtirish talabalarning o‘quv-bilish faolligini oshirishga yo‘naltiriladi.

O‘zbekiston Respublikasida ta’lim tizimini raqamlashtirish, axborot-kommunikatsiya texnologiyalarini keng joriy etish va zamonaviy kasbiy kompetensiyalarga ega mutaxassislarni tayyorlash davlat siyosatining ustuvor yo‘nalishlari qatoriga kiradi. Jumladan, O‘zbekiston Respublikasi Prezidentining 2020-yil 5-oktabrdagi PF-6079-son Farmoni bilan tasdiqlangan “Raqamli O‘zbekiston – 2030” strategiyasida raqamli texnologiyalarni rivojlantirish, raqamli ko‘nikmalarni kengaytirish va sohada malakali kadrlar tayyorlash bilan bog‘liq vazifalar belgilangan. Shuningdek, O‘zbekiston Respublikasining 2020-yil 23-sentabrdagi O‘RQ-637-son “Ta’lim to‘g‘risida”gi Qonuni ta’lim jarayonini tashkil etish va uning sifatini ta’minlashning huquqiy asoslarini belgilaydi. Mazkur me’yoriy-huquqiy asoslar oliy ta’limda dasturlash fanlarini o‘qitish metodikasini takomillashtirish, ta’lim mazmunini amaliy faoliyat bilan uyg‘unlashtirish va talabalarning raqamli kompetensiyalarini rivojlantirishga qaratilgan tadqiqotlarning dolzarbligini belgilashga xizmat qiladi.

MUHOKAMA VA NATIJALAR

Dasturlash operatorlaridan foydalanish ko‘nikmalarini baholash uchun o‘quv faoliyatini beshta funksional yo‘nalishga ajratish taklif etiladi. Bunda har bir yo‘nalish muayyan kuzatiladigan amaliy harakat orqali baholanadi.

1-jadval.

Yo‘nalish	**Talabadan kutiladigan amaliy natija**
Arifmetik amallar	Ifodalarni to‘g‘ri hisoblash va ustuvorlikni aniqlash
Taqqoslash amallari	Shartlarning rost yoki yolg‘onligini aniqlash
Mantiqiy amallar	Murakkab shartlarni and, or, not orqali tuzish
O‘zlashtirish amallari	O‘zgaruvchilar qiymati qanday yangilanishini tushuntirish
Boshqaruv konstruksiyalari	Shartli tarmoqlanish va takrorlash algoritmlarini yaratish

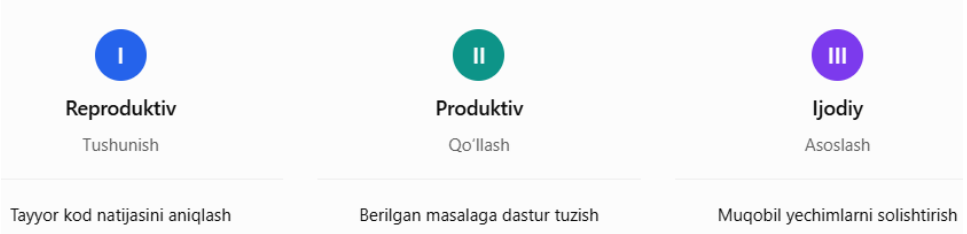
Bu tasnifda shartli va takrorlash konstruksiyalari arifmetik operatorlar bilan aynan bir sintaktik turga kiritilmaydi. Ular algoritm bajarilishini boshqaruvchi konstruksiyalar sifatida alohida o‘rganiladi. Mazkur farqlash Python va C++ tillarining sintaktik xususiyatlarini hisobga olish uchun muhimdir. Operatorlarni o‘rgatishda bilimni bevosita tayyor dasturga aylantirishdan ko‘ra, uni ketma-ket faoliyat bosqichlari orqali shakllantirish taklif etiladi.



1-rasm. Operatorlardan foydalanish ko'nikmalarini shakllantirishning bosqichli modeli

Ushbu modelda diagnostika bosqichi talabaning mavjud bilim darajasini aniqlashga, semantik tushunish bosqichi operatorlarning bajarilish mexanizmini anglashga, amaliy qo'llash bosqichi esa algoritmik topshiriqlarni dasturlash tilida ifodalashga xizmat qiladi. Mustaqil yechim bosqichida talaba masalani hal qilishning bir nechta variantini ishlab chiqadi. Refleksiv baholash bosqichida esa dastur natijasi, tanlangan operatorlarning maqsadga muvofiqligi va aniqlangan xatolar tahlil qilinadi. Modelning muhim xususiyati baholash natijasiga ko'ra oldingi bosqichlarga qaytish imkoniyatining mavjudligidir. Masalan, talaba mantiqiy ifodani tuzishda xatoga yo'l qo'ysa, o'qituvchi uni bevosita murakkab dasturlash topshirig'iga yo'naltirmasdan, avval rostlik jadvallari va sodda shartlarni hisoblash mashqlari orqali tegishli bilimni mustahkamlaydi.

Dasturlash operatorlarini o'qitishda bir xil tipdagi mashqlarni takrorlash o'rniga, turli kognitiv faoliyatni talab qiladigan topshiriqlarni qo'llash maqsadga muvofiq. Buning uchun uch darajali topshiriqlar tizimi taklif etiladi.



2-rasm. Amaliy topshiriqlarning uch darajali tuzilishi

Birinchi daraja uchun quyidagi Python kodining natijasini aniqlash topshirig'i berilishi mumkin:

```python
a = 5
b = 3
c = a + b * 2
print(c)
```

Dastur natijasi 11 ga teng. Bu topshiriq orqali talabaning arifmetik operatorlar ustuvorligi haqidagi bilimini aniqlash mumkin. Agar talaba natijani 16 deb hisoblasa, ko'paytirish amalining qo'shishdan oldin bajarilishini tushunmagan bo'lishi mumkin.

Ikkinchi daraja uchun 1 dan n gacha bo'lgan juft sonlar yig'indisini hisoblash vazifasi beriladi.

```python
n = int(input())
s = 0

for i in range(1, n + 1):
    if i % 2 == 0:
        s += i

print(s)
```

Mazkur topshiriqda % arifmetik operatori, == taqqoslash operatori, += kengaytirilgan o‘zlashtirish operatori, if shartli konstruksiyasi va for takrorlash konstruksiyasi o‘zaro bog‘langan holda qo‘llanadi.

Uchinchi daraja uchun talabaga ushbu masalani kamida ikki xil algoritmik usulda yechish, yechimlarni solishtirish va tanlangan usulni asoslash vazifasi beriladi. Masalan, juft sonlarni siklda ketma-ket qo‘shish yoki k = n // 2 orqali juft sonlar sonini aniqlab, k * (k + 1) formulasidan foydalanish mumkin. Bunda talaba faqat dastur yozmaydi, balki algoritmning to‘g‘riligini va samaradorligini ham tahlil qiladi.

Dasturiy xatolarni faqat texnik nosozlik sifatida emas, balki talabaning bilimida mavjud bo‘shliqlarni aniqlash imkonini beruvchi diagnostik axborot sifatida o‘rganish taklif etiladi.

2-jadval.

Xato turi	**Dasturiy namuna**	**Pedagogik diagnostika**
Ustuvorlik xatosi	2 + 3 * 4 natijasini 20 deb hisoblash	Amallar bajarilish tartibini noto‘g‘ri tushunish
Mantiqiy xato	x > 0 or x < 10 ni 0 dan 10 gacha oraliqni tekshirish uchun ishlatish	Mantiqiy bog‘lovchilarni farqlashdagi qiyinchilik
Chegara xatosi	range(1, 10) orqali 10 sonini ham qamrab olishni kutish	Interval chegaralarini tushunmaslik
O‘zlashtirish xatosi	x = x + 1 ifodasini matematik tenglama sifatida talqin qilish	O‘zgaruvchi holatining yangilanishini anglamaslik
Sikl xatosi	while ichida shartga ta’sir etuvchi qiymatni o‘zgartirmaslik	Takrorlashni boshqarish mexanizmini tushunmaslik

Xatolarni tahlil qilishda uch bosqichli yondashuv tavsiya etiladi: xatoni aniqlash, uning sababini izohlash va tuzatilgan kodni sinovdan o‘tkazish. Bu usul o‘qituvchiga faqat noto‘g‘ri javobni belgilash emas, balki talabaning qaysi tushunchani qayta o‘zlashtirishi zarurligini aniqlash imkonini beradi.

Masalan, x > 0 or x < 10 ifodasi haqiqiy sonlar uchun doimo rost bo‘ladi. Chunki har qanday haqiqiy son kamida shu ikki shartdan birini qanoatlantiradi. Agar maqsad sonning 0 dan katta va 10 dan kichikligini tekshirish bo‘lsa, x > 0 and x < 10 ifodasidan foydalanish kerak. Mazkur misolni rostlik jadvali asosida tahlil qilish mantiqiy operatorlarning semantikasini tushuntirish uchun qulay metodik vosita hisoblanadi.

Talabalarning operatorlardan foydalanish kompetensiyasini baholash uchun 100 ballik analitik rubrika taklif etiladi. Mazkur rubrika amaliy mashg‘ulot yoki diagnostik nazorat uchun mualliflik baholash vositasi bo‘lib, OAK tomonidan belgilangan majburiy ball taqsimoti sifatida talqin qilinmaydi.

Baholash mezoni	**Tekshiriladigan ko‘nikma**	**Maksimal ball**
Sintaktik savodxonlik	Operatorlarni dasturlash tili qoidalariga muvofiq yozish	15
Semantik tushunish	Operatorlarning bajarilish natijasini izohlash	20
Amaliy qo‘llash	Masalaga mos operator va konstruksiyalarni tanlash	25
Xatolarni aniqlash	Koddagi xatolarni topish va tuzatish	20
Algoritmik asoslash	Yechimni tushuntirish va muqobil variantlarni solishtirish	20
Jami		100

Baholash mezonlari uchta yirik komponentga birlashtirilishi mumkin: kognitiv komponent (35 ball), amaliy-faoliyatli komponent (45 ball) va refleksiv-baholash komponenti (20 ball). Bunda sintaktik savodxonlik hamda semantik tushunish kognitiv komponentni, amaliy qo‘llash va xatolarni aniqlash amaliy-faoliyatli komponentni, algoritmik asoslash esa refleksiv-baholash komponentini ifodalaydi.

XULOSA

Dasturlashda operatorlardan foydalanish ko‘nikmalarini shakllantirish va rivojlantirish talabalarning algoritmik tafakkuri, mantiqiy fikrlashi, mustaqil qaror qabul qilish qobiliyati hamda amaliy dasturlash kompetensiyalarini takomillashtirishning muhim pedagogik omillaridan biri hisoblanadi. Operatorlardan foydalanish ko‘nikmalarini kognitiv, amaliy-faoliyatli va refleksiv-baholash komponentlari asosida tizimli baholash, amaliy topshiriqlarni reproduktiv, produktiv va ijodiy darajalarda tashkil etish hamda dasturiy xatolardan pedagogik diagnostika vositasi sifatida foydalanish o‘quv jarayonini takomillashtirishning istiqbolli yo‘nalishlari sifatida belgilandi. Shuningdek, dasturiy kodning bajarilish jarayonini vizuallashtirish, differensial topshiriqlardan foydalanish, algoritmik yechimlarni qiyosiy tahlil qilish va refleksiv faoliyatni tashkil etish talabalarning nazariy bilimlarini amaliy vaziyatlarda qo‘llash imkoniyatlarini kengaytirishi mumkinligi aniqlandi.

FOYDALANILGAN ADABIYOTLAR

1. Robins A., Rountree J., Rountree N. Learning and Teaching Programming: A Review and Discussion // Computer Science Education. – 2003. – Vol. 13, № 2. – P. 137–172. – DOI: https://doi.org/10.1076/csed.13.2.137.14200.

2. Lahtinen E., Ala-Mutka K., Järvinen H.-M. A Study of the Difficulties of Novice Programmers // Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education. – 2005. – P. 14–18. – DOI:

https://doi.org/10.1145/1067445.1067453.

3. Lister R., Adams E. S., Fitzgerald S. et al. A Multi-National Study of Reading and Tracing Skills in Novice Programmers // ACM SIGCSE Bulletin. – 2004. – Vol. 36, № 4. – P. 119–150. – DOI: https://doi.org/10.1145/1044550.1041673.

4. Sorva J. Notional Machines and Introductory Programming Education // ACM Transactions on Computing Education. – 2013. – Vol. 13, № 2. – Article 8. – DOI: https://doi.org/10.1145/2483710.2483713.

5. Fincher S., Baker B., Box I. et al. Programmed to Succeed? A Multi-National, Multi-Institutional Study of Introductory Programming Courses. – Canterbury: University of Kent, Computing Laboratory, 2005. – Technical Report № 1-05.