Annaqulov Mahmudjon Sodiqovich
Navoiy davlat universiteti mustaqil izlanuvchisi
mannaqulov85@gmail.com
0009-0002-4068-7380

ICHKI ISHLAR AKADEMIK LITSEYI O‘QUVCHILARINING DASTURLASH KOMPETENSIYASINI SHAKLLANTIRISHNING BOSQICHLI METODIKASI

Annotatsiya. *Mazkur maqolada ichki ishlar akademik litseyi o‘quvchilarining dasturlash kompetensiyasini shakllantirishga qaratilgan bosqichli metodikaning nazariy, didaktik va tashkiliy asoslari yoritilgan. Dasturlash kompetensiyasi masalani formallashtirish, ma’lumotlar modelini tanlash, algoritm tuzish, Python tilida kod yaratish, testlash va yechimni asoslashni birlashtiradigan integrativ faoliyat sifatida talqin etilgan. PRIMM yondashuvi asosida kodni tushunishdan mustaqil dastur ishlab chiqishgacha bo‘lgan o‘quv harakatlari izchil loyihalashtirilgan. Pedagogik yordamni diagnostika natijalariga muvofiq kamaytirish, sun’iy yaratilgan kasbiy vaziyatlar yordamida axborot xavfsizligi talablariga rioya qilish va individual o‘quv natijalarini qayd etish mexanizmlari taklif qilingan. Olti mezonli, 18 ballik rubrika, diagnostik topshiriqlar, test protokoli hamda refleksiv hisobotning o‘zaro bog‘liqligi ochib berilgan. Taklif etilayotgan tizim nazariy-metodik ishlanma bo‘lib, uning samaradorligini baholash uchun keyingi empirik tadqiqot mezonlari belgilangan.*

Kalit so‘zlar: *dasturlash kompetensiyasi, ichki ishlar akademik litseyi, Python, algoritmik fikrlash, PRIMM, pedagogik yordam, formallashtirish, testlash, baholash rubrikasi.*

ПОЭТАПНАЯ МЕТОДИКА ФОРМИРОВАНИЯ КОМПЕТЕНЦИИ ПРОГРАММИРОВАНИЯ УЧАЩИХСЯ АКАДЕМИЧЕСКОГО ЛИЦЕЯ МВД

Аннотация. *В статье раскрываются теоретические, дидактические и организационные основания поэтапной методики формирования компетенции программирования у учащихся академических лицеев Министерства внутренних дел. Компетенция рассматривается как интегративная способность формализовать задачу, выбирать модель данных, разрабатывать алгоритм, создавать программу на Python, проводить тестирование и аргументировать полученный результат. С опорой на подход PRIMM предложена последовательность учебных действий от осмысленного чтения готового кода до самостоятельного проектирования решения. Обоснованы постепенное сокращение педагогической поддержки с учётом диагностики, использование искусственно сформированных профессионально ориентированных ситуаций без персональных и служебных данных, а также фиксация индивидуального вклада учащегося. Представлены шесть критериев оценивания с общей шкалой в 18 баллов, диагностические задания, протокол проверки и элементы рефлексивного отчёта. Методическая модель не отождествляется с доказанным экспериментальным эффектом: для последующей проверки определены требования к сопоставимости заданий, надёжности оценивания и анализу первичных данных.*

Ключевые слова: *компетенция программирования, академический лицей МВД, Python, алгоритмическое мышление, поэтапное обучение, педагогическая поддержка, тестирование.*

A STAGED METHODOLOGY FOR DEVELOPING PROGRAMMING COMPETENCE AMONG MIA ACADEMIC LYCEUM STUDENTS

Annotation. *This article presents the theoretical, didactic, and organizational foundations of a staged methodology for developing programming competence among students of academic*

lyceums affiliated with the Ministry of Internal Affairs. Programming competence is defined as an integrated ability to formalize problems, select data representations, design algorithms, implement Python programs, test alternative inputs, and justify solutions. Drawing on the PRIMM approach, the proposed sequence moves from purposeful code reading and modification to independent solution design. The article explains how instructional scaffolding can be gradually withdrawn in response to diagnostic evidence and how profession-related tasks can be constructed using entirely synthetic datasets, thereby protecting personal and service information. It also specifies a six-criterion, eighteen-point assessment rubric, individual diagnostic assignments, testing records, and reflective reporting procedures. Particular attention is paid to distinguishing collaborative learning from individually demonstrated competence. As the contribution is primarily methodological, the paper identifies the conditions required for subsequent empirical validation instead of making unsupported claims about learning gains.

Keywords: *programming competence, MIA academic lyceum, Python, computational thinking, scaffolding, testing, rubric-based assessment.*

Kirish. Raqamli transformatsiya sharoitida informatika ta’limining natijasi tayyor dasturdan foydalanish yoki operatorlarni ta’riflash bilangina cheklanmaydi. O‘quvchi muammoni hisoblash nuqtai nazaridan ifodalashi, ma’lumot va cheklovlarni ajratishi, bajariladigan algoritmni tanlashi, uni kodga aylantirishi hamda natijaning ishonchliligini mustaqil tekshira olishi zarur. Aynan shu harakatlar birligi dasturlash kompetensiyasining amaliy mohiyatini belgilaydi [2; 3].

Ichki ishlar akademik litseylari uchun mazkur masala o‘ziga xos ahamiyat kasb etadi. Bir tomondan, o‘quvchilar umumiy informatika bilimlarini egallaydi, ikkinchi tomondan, kelgusi kasbiy faoliyat bilan mazmunan bog‘liq ma’lumotlarni aniq, izchil va mas’uliyat bilan qayta ishlash madaniyati shakllanadi. Biroq litsey o‘quvchisini amaldagi xizmatchi bilan tenglashtirish yoki uning o‘quv vazifasiga maxfiy xizmat ma’lumotlarini kiritish metodik va etik jihatdan maqbul emas. Shu sababli o‘quv maqsadiga mos, shartli va sintetik ma’lumotlar asosidagi vaziyatlar tanlanadi.

Dissertatsiyada asos qilib olingan 2025 yilgi o‘quv dasturida aniq fanlar yo‘nalishi informatikasiga 360 auditoriya soati, jumladan Python bo‘limiga 80 soat ajratilgan [1]. Bu vaqt dasturlashni boshlang‘ich tushunchalardan tekshiriladigan mustaqil amaliy mahsulotgacha olib borish uchun muayyan didaktik imkoniyat yaratadi. Biroq soatlar hajmi o‘z-o‘zidan kompetensiyani kafolatlamaydi; maqsad, topshiriq, yordam va baholash o‘rtasida mazmuniy muvofiqlik talab etiladi.

Tadqiqot muammosi shundan iboratki, o‘quvchining tayyor kodni takrorlashi bilan yangi shartda mustaqil dastur tuzishi ko‘p hollarda bir xil o‘zlashtirish ko‘rsatkichi sifatida qabul qilinishi mumkin. Demak, metodikani takomillashtirish uchun nafaqat mashqlar ketma-ketligini, balki har bir bosqichda qaysi harakatni o‘quvchi mustaqil bajarganini ko‘rsatadigan dalillarni ham aniqlash zarur.

Muammoning didaktik ildizi dasturlash jarayonining ko‘rinadigan mahsuloti - ishlayotgan kod bilan shu mahsulotni yaratishga olib keladigan tafakkur amallarining doimo mos kelmasligidadir. Masalan, o‘quvchi tayyor namunaning ayrim sonlarini almashtirib, kutilgan natijani olishi mumkin; biroq masala sharti o‘zgarganida algoritmni nima sababdan qayta qurish kerakligini izohlay olmasligi ehtimoli saqlanib qoladi. Shuning uchun o‘quv natijasi kodning bajarilishi, o‘quvchining qarorni tushuntirishi va notanish vazifaga ko‘chira olishi kabi kamida uch turdagi dalil asosida tahlil qilinadi. Ushbu yondashuv o‘rgatish jarayoni bilan uni diagnostika qilishni yagona mantiqiy tizimga birlashtirishga xizmat qiladi.

Tadqiqot doirasida “bosqichli metodika” tushunchasi mashg‘ulotlarni oddiygina ketma-ket joylashtirishni anglatmaydi. U oldingi bosqichda o‘zlashtirilgan harakatning keyingisiga zarur shart bo‘lishini, yordam me’yorining individual tayyorgarlik bilan muvofiqlashishini va o‘tish mezonlarining oldindan belgilanishini nazarda tutadi. Bunday tashkil etishda barcha o‘quvchilar bir xil vaqtda bir xil darajadagi topshiriqqa o‘tishi majburiy emas: ayrimlar uchun qisqa tuzatish mashqi, boshqalar uchun esa yangi kontekstga transfer topshirig‘i maqsadga muvofiq bo‘ladi. Shu

tariqa differensiallashtirish o‘quv mazmunini tartibsizlashtirmasdan, umumiy natijaga eltuvchi turli ta’lim yo‘llarini yaratadi.

Adabiyotlar sharhi. J. Ving hisoblash tafakkurini muammolarni kompyuter fanining tushunchalari orqali ifodalash bilan bog‘lagan [2]. M. Saeli va hammualliflar o‘rta ta’limda dasturlashni o‘qitishning mazmuni, maqsadi hamda pedagogik usulini o‘zaro aloqador holda tahlil qilganlar [3]. S. Sentans, J. Ueyt va M. Kalliya taklif etgan PRIMM yondashuvida prognozlash, ishga tushirish, tadqiq qilish, o‘zgartirish va yaratish jarayonlari o‘quvchini tayyor koddan mustaqil mahsulotga bosqichma-bosqich yetaklaydi [4].

Mazkur qarashlar umumiy metodik asos bo‘lsa-da, ularni ichki ishlar akademik litseyining muayyan o‘quv rejasiga, kasbiy mazmunli sintetik holatlarga va individual natijani baholash talablariga moslashtirish alohida pedagogik yechimni taqozo etadi. Shu nuqtai nazardan maqolaning maqsadi dasturlash kompetensiyasining kuzatiladigan belgilariga asoslangan bosqichli o‘qitish va baholash tizimini ilmiy jihatdan asoslashdan iborat.

Hisoblash tafakkuri haqidagi qarashlarda muammoni qismlarga ajratish, muhim belgilarni tanlash va jarayonni algoritmik ifodalash dasturlashdan kengroq intellektual faoliyat sifatida baholanadi [2]. Shu bilan birga, dasturlashni o‘rganishning boshlang‘ich bosqichida o‘quvchi sintaktik tafsilotlar bilan algoritmik ma’noni aralashtirib yuborishi mumkin. Mazkur qiyinchilikni bartaraf etishda tayyor kodning ishini bashorat qilish, bajarilgan natijani kuzatish hamda dastur fragmentidagi kichik o‘zgarishning oqibatini muhokama qilish samarali didaktik tayanch vazifasini bajaradi [4; 6]. Bu tayanch o‘quvchini kodni mexanik ko‘chirishdan farqli ravishda sabab-oqibat aloqalarini ko‘rishga yo‘naltiradi.

PRIMMning Predict, Run, Investigate, Modify va Make ketma-ketligi dasturga oid kognitiv faoliyatni tartiblashga imkon beradi [4]. Biroq uning har bir bo‘g‘ini alohida maqsad bilan to‘ldirilishi zarur: taxmin qilishda o‘quvchi dastur oqimini izohlaydi; ishga tushirishda taxminni natija bilan qiyoslaydi; tadqiq qilishda o‘zgaruvchi va shartlarning funksiyasini tekshiradi; o‘zgartirishda sababli taxminni amaliy sinaydi; yaratishda esa o‘zi masalani modellashtiradi. Mazkur izchillik kasbiy yo‘naltirilgan vazifalar bilan boyitilganda, o‘quvchi umumiy sintaksisdan predmet mazmunini hisoblash amallariga aylantirish tomon harakat qiladi.

Brennan va Resnickning baholashga oid qarashlarida hisoblash konsepsiyalari, amaliyotlari va nuqtayi nazarlarining bir-biridan farqlanishi muhim o‘rin tutadi [5]. Bu farqlash dasturlash kompetensiyasini “biladi – bilmaydi” juftligi bilan ifodalash yetarli emasligini anglatadi. O‘quvchi siklni nazariy ta’riflab berishi, lekin chegaraviy holatni aniqlay olmasligi yoki dastur ishlashidagi xatoni tasodifiy almashtirishlar bilan tuzatishi mumkin. Shuning uchun maqolada bilimni aniqlashga oid savol bilan mustaqil amaliy ish alohida o‘lchov sifatida qaraladi; ularning natijalari yakuniy pedagogik xulosada o‘zaro solishtiriladi.

Metodologiya. Tadqiqotning metodologik asosini kompetensiyaviy, tizimli-faoliyatli, muammoli va refleksiv yondashuvlar tashkil etadi. Manbalar va o‘quv dasturi qiyosiy tahlil qilindi, kompetensiyaning tarkibi operatsionallashtirildi, dars jarayoni esa maqsad - vazifa – harakat - mahsulot - baholash - qayta aloqa zanjiri bo‘yicha pedagogik modellashtirildi. Ushbu maqolada bayon etilgan natijalar asosan nazariy-metodik xarakterga ega.

Dasturlash kompetensiyasi dastur sintaksisini bilish bilan bir qatorda beshta asosiy harakatda namoyon bo‘ladi: masalaning kirish va chiqish shartlarini formallashtirish; algoritm va ma’lumotlar tuzilmasini tanlash; ishlovchi kod yaratish; odatiy, chegaraviy va noto‘g‘ri kirish holatlarida testlash; qabul qilingan yechimning sababini va amal qilish chegarasini asoslash. Har bir harakatga tegishli mahsulot mavjud bo‘lishi baholashni tekshiriladigan jarayonga aylantiradi.

Bunda motivatsion, kognitiv, amaliy-faoliyatli va refleksiv jihatlar o‘zaro bog‘liq, biroq bir xil emas. Masalan, vazifaga qiziqishning ortishi ishlaydigan kodni mustaqil yozish bilan tenglashtirilmaydi; xuddi shuningdek, tasodifan to‘g‘ri natija bergan kod to‘liq algoritmik tushunish dalili bo‘la olmaydi. Shuning uchun so‘rovnoma, kuzatuv, dastur fayli va test izohi turli turdagi dalil sifatida yig‘iladi [5].

Metodik tizimni loyihalashda to‘rtta talab ustuvor deb olindi. Birinchisi - mazmuniy izchillik: keyingi topshiriq oldingi tushunchaga tayanadi. Ikkinchisi - kuzatuvchanlik: har bir

bosqichda o‘quvchi nima bajargani yozma yoki raqamli dalil orqali ko‘rinadi. Uchinchisi - individuallik: jamoaviy ishlash imkoniyati saqlangani holda mustaqil natija alohida aniqlanadi. To‘rtinchisi - transfer: o‘zlashtirilgan amal tanish misoldan bir oz farq qiladigan yangi vaziyatda sinab ko‘riladi. Ushbu talablarning birgalikda qo‘llanishi mashg‘ulotni nazariy bayon, mashq, tahlil va baholashning uzviy tizimi sifatida tashkil etadi.

Kompetensiyaning tarkibiy komponentlari o‘zaro bog‘langan bo‘lsa ham, ularning dalillari turlicha kuzatiladi. Motivatsion jihat muammoni hal qilishga tayyorlik va faol ishtirokda, kognitiv jihat ma’lumot turi, algoritm va operatorlar o‘rtasidagi aloqani izohlashda namoyon bo‘ladi. Amaliy-faoliyatli jihat to‘liq dastur fayli, ishlash protokoli va kirish-chiqish mosligida, refleksiv jihat esa qabul qilingan yechim cheklovini anglash va xatoga asosli tuzatish kiritishda namoyon bo‘ladi. Shu sababli bir xil test har to‘rtta komponentni birdek o‘lchaydi deb hisoblash metodologik xatodir; ko‘p manbali baholash dalili talab etiladi.

Tavsiya etilayotgan o‘quv siklining minimal hujjatlari vazifa spetsifikatsiyasi, o‘quvchi tuzgan algoritmning qisqa bayoni, bajariladigan kod, kamida uch turdagi test yozuvi va refleksiv izohdan iborat. Vazifa spetsifikatsiyasida muammo, kiritiladigan ma’lumotlar, cheklovlar hamda natijani qabul qilish sharti aniq bo‘lishi kerak. Masalan, ‘murojaatlarni sanash’ kabi umumiy ko‘rsatma o‘rniga qaysi toifa nomlari hisoblanishi, bo‘sh qiymat qanday qayd etilishi va takroriy satrni alohida hodisa deb qabul qilish-qilmaslik bayon etiladi. Bu aniqlik baholovchilar o‘rtasidagi subyektiv talqinni kamaytiradi.

Natijalar. Ishlab chiqilgan metodik yechimning yadrosi o‘quvchi faoliyatiga nisbatan qo‘llaniladigan pedagogik tayanchlarni izchil qisqartirishdan iborat. Birinchi bosqichda vazifa sharti, kirish-chiqish ma’lumotlari va kutilgan natija muhokama qilinadi. Ikkinchi bosqichda qisqa tayyor dastur fragmenti o‘qilib, natija avvaldan prognoz qilinadi. Uchinchi bosqichda kod o‘zgartirilib, gipoteza yangi kirish ma’lumotida tekshiriladi. To‘rtinchi bosqichda mustaqil algoritm va dastur yaratiladi. Beshinchi bosqichda testlar, xatolar tahlili va qisqa refleksiya orqali natija asoslanadi.

Mazkur ketma-ketlik PRIMMning didaktik mantig‘i bilan uyg‘un bo‘lsa-da, uni mexanik takrorlamaydi: litseyga mos ishchi modelda vazifani formallashtirish va individual tekshirish mustaqil nazorat nuqtalari sifatida ajratiladi. Bunda har bir yangi o‘quv bosqichi avvalgisining bajarilganligi haqidagi dalilga tayanadi. Masalan, o‘quvchi siklning vazifasini tushuntirmagan bo‘lsa, murakkab fayllar bilan ishlashni erta kiritish kognitiv yuklamani asossiz oshiradi.

Birinchi - muammoni formallashtirish bosqichida o‘quvchi tabiiy tildagi vazifani “kirish ma’lumotlari - amal - chiqish ma’lumotlari cheklov” tuzilmasiga keltiradi. U ma’lumotlar ro‘yxat, lug‘at yoki fayl ko‘rinishida berilganini, natija bitta sonmi yoki tartiblangan jadvalmi, noma’lum qiymat qanday talqin qilinishini aniqlaydi. O‘qituvchi ushbu bosqichda tayyor kodni bermaydi; muammoga oid savollar bilan aniq shart tuzishga undaydi. Nazorat nuqtasi sifatida o‘quvchining yechimdan oldin kamida bitta oddiy va bitta chegaraviy misol uchun kutilgan natijani yozishi taklif etiladi.

Ikkinchi - kodni bashorat qilish va anglash bosqichida hajmi kichik, tarkibi avval o‘rganilgan operatorlardan tashkil topgan dastur taqdim etiladi. O‘quvchi kodni ishga tushirishdan oldin uning natijasini, o‘zgaruvchi qiymatlarining o‘zgarishini va sikl necha marta takrorlanishini yozadi. Natija prognozdan farq qilganida, qaysi satr noto‘g‘ri tushunilganini aniqlaydi. Bunday faoliyat dastur bajarilishini faqat tashqi natija sifatida ko‘rish emas, balki uning ichki mexanizmini tushunishga imkon beradi. Ayniqsa, shart operatori va takrorlash operatorlarining birgalikdagi ishlashida ushbu bosqichga yetarli vaqt ajratish maqsadga muvofiq.

Uchinchi - tadqiq qilish va modifikatsiyalash bosqichida dasturga bitta mazmunli o‘zgarish kiritiladi. Masalan, dastlab mavjud toifalarni sanaydigan kodga bo‘sh satrni tekshirish yoki natijani kamayish tartibida chiqarish talabi qo‘shiladi. O‘quvchi nima o‘zgarishini oldindan izohlaydi, keyin dastur natijasi bilan qiyoslaydi. Bir vaqtning o‘zida juda ko‘p talab kiritilmasligi lozim: aks holda muammo qayerda paydo bo‘lgani noaniqlashadi va o‘qituvchi yordamining samaradorligini baholash qiyinlashadi. Shu bosqichda kuzatiladigan asosiy dalil - kiritilgan o‘zgarishning sababini tushuntirishdir.

To‘rtinchi - mustaqil yechim yaratish bosqichida o‘quvchi avvalgi dastur matnini takrorlamaydigan, lekin o‘sha algoritmik g‘oyaga tayanuvchi yangi topshiriqni oladi. Masalan, toifalar chastotasini hisoblashdan shartli hodisalarni vaqt oralig‘i bo‘yicha guruhlashga o‘tilishi mumkin. Muammoning kasbiy mavzusi yordamchi kontekst bo‘lib, asosiy pedagogik natija ma’lumotni ifodalash, sikl yoki shartni tanlash va ishchi kodni yaratishdagi mustaqillik bilan belgilanadi. O‘quvchi dasturini yozish jarayonida qaysi manbadan yoki maslahatdan foydalanganini belgilashi ham akademik halollik, ham diagnostik shaffoflik uchun foydalidir.

Beshinchi - testlash va refleksiya bosqichida o‘quvchi o‘z dasturiga nisbatan faqat tasdiqlovchi emas, balki xatoni yuzaga chiqarishi mumkin bo‘lgan misollarni ham tanlaydi. Oddiy holat, bo‘sh yoki chegaraviy holat hamda noto‘g‘ri formatdagi holat uchun kutilgan va amaldagi natijalar alohida qayd etiladi. Kod o‘zgartirilgan bo‘lsa, oldingi testlar takror bajariladi. Refleksiya qismida ‘qaysi qaror to‘g‘ri bo‘ldi?’, ‘qaysi shartda yechim ishlamaydi?’ va ‘keyingi safar nimani o‘zgartiraman?’ savollariga dalilli javob beriladi. Bu dasturlashni faqat mahsulot tayyorlash emas, ishonchli yechimni asoslash faoliyatiga aylantiradi.

Pedagogik yordam uch darajada tashkil etiladi: birinchi daraja - shart va natija mezoni; ikkinchi daraja - yo‘naltiruvchi savol yoki qisman to‘ldirilgan reja; uchinchi daraja - kod bloklari yoxud funksiya skeleti. Yordam miqdori kalendar muddatga qarab emas, o‘quvchining mustaqil harakatiga muvofiq kamaytiriladi. Bu tartib yuqori natijani pedagog yordami bilan olingan natijadan farqlash imkonini beradi.

Pedagogik yordamni kamaytirishning didaktik mantiqi yordamning uch darajasini bir xil vazifaga nisbatan qo‘llash orqali ochiladi. Dastlab o‘qituvchi masalaning sharti va muvaffaqiyat mezonini tushuntiradi; keyin o‘quvchining to‘xtab qolgan joyiga mos yo‘naltiruvchi savol beradi; zarurat tug‘ilgandagina funksiyaning skeleti yoki algoritmning qisman tuzilgan rejasi bilan ta’minlaydi. Agar o‘quvchi qo‘shimcha yordamdan keyin ishni bajarsa, ushbu natija ‘mustaqil bajardi’ deb qayd etilmaydi. Keyingi shunga yaqin vazifada yordam hajmi kamaytirilishi hamda o‘quvchining mustaqil bajarish imkoniyati qayta sinovdan o‘tkazilishi kerak.

Yordamni boshqarish uchun qisqa diagnostik kartadan foydalanish taklif etiladi. Unda “masala shartini izohladi”, “algoritmni tanladi”, “kod tuzdi”, “testlarni ishlab chiqdi”, “xatoni asoslab tuzatdi” ustunlari va har biri bo‘yicha “mustaqil”, “savol yordamida”, “namuna yordamida” belgilari qayd etiladi. Karta summativ ball o‘rnini bosa olmaydi, uning vazifasi ayni paytda qaysi tushunchaga tayanch kerakligini aniqlashdan iborat. Bir o‘quvchi ma’lumotlarni tuzilmalashda mustaqil, boshqasi esa testlashda kuchli bo‘lishi mumkin; shuning uchun yordam bir xil “kuchli-kuchsiz” yorlig‘i asosida emas, harakatlar kesimida ko‘rsatiladi.

Dasturlashni kasbiy yo‘naltirishda asosiy mezon vazifaning tashqi nomlanishi emas, balki predmet vaziyati algoritm tanloviga ta’sir qilishidir. Masalan, shartli murojaatlar reestrini tahlil qilishda o‘quvchi kirish ma’lumotlari maydonlarini, takroriy qaydni aniqlash qoidasini, bo‘sh maydonga munosabatni va natijaning chiqish shaklini mustaqil belgilaydi. Bu jarayonda ma’lumotlarni guruhlash, tekshirish va saralash amallari haqiqiy dasturlash vazifasiga aylanadi.

O‘quv masalasining birinchi variantida ma’lumotlar Python ro‘yxatida berilib, o‘quvchi toifalar bo‘yicha sanoqni amalga oshiradi. Keyingi variantda lug‘at orqali yig‘ma ma’lumot olinadi, uchinchi variantda noto‘g‘ri yoki takroriy qaydlar ajratiladi, oxirgi bosqichda fayldan o‘qish va natijani saqlash kiritiladi. Yangi talablar bittadan qo‘shilganida xato manbasini ajratish hamda kerakli tayanchni tanlash osonlashadi.

Kasbiy vaziyatni pedagogik masalaga aylantirishda uch pog‘onali transformatsiya zarur: birinchidan, real faoliyatdan umumiy intellektual amal ajratiladi; ikkinchidan, mazkur amal shaxsni identifikatsiyalamaydigan uydirma ma’lumotlarga ko‘chiriladi; uchinchidan, masala o‘quv dasturida o‘zlashtirilgan operator va tuzilmalar bilan yechiladigan shaklga keltiriladi. Masalan, xabarlarni saralashga oid umumiy ish jarayoni toifalar chastotasini hisoblash, kiritilgan qiymatni tekshirish va yakuniy xulosani taqdim etish algoritmiga aylantiriladi. Bunda haqiqiy murojaatlar, xizmat rekvizitlari yoki shaxsga doir ma’lumotlar o‘quv topshirig‘iga kiritilmaydi.

Python asosidagi amaliy mashqlarni murakkablashtirishda minimal ishlovchi yechim tamoyili qo‘llanadi. Dastlab o‘quvchi ro‘yxat elementlarini aylanib chiqish va hisoblagichdan

foydalanishni ko‘rsatadi. Keyin natijalarni lug‘atda jamlaydi; keyingi qadamda xato qiymatni aniqlash va bu haqda alohida xabar berish qo‘shiladi. Shundan so‘ng funksiyaga ajratish va sinovlarni avtomatlashtirishga o‘tish mumkin. Fayl bilan ishlash esa mazkur amallar egallangandan keyin kiritiladi. Bu izchillik murakkab interfeys yaratishga berilib ketmasdan algoritmning mohiyatini o‘zlashtirishga imkon yaratadi.

Masalan, kirish ma’lumoti “A, B, A, C, B, A” ko‘rinishida bo‘lganda toifalar chastotasini hisoblashning kutilgan natijasi A - 3, B - 2, C - 1 bo‘ladi. Lekin algoritmning barqarorligini tekshirish uchun bo‘sh ro‘yxat, noma’lum toifa, takroriy belgilar va ma’lumot shakli buzilgan holat ham alohida qaralishi kerak. Demak, asosiy savol “kod ishladimi?” emas, “kod qaysi shartlarda ishonchli ishlaydi?” tarzida qo‘yiladi.

Masalaning aniq spetsifikatsiyasi quyidagicha ishlab chiqilishi mumkin: kirish - A, B, C toifalaridan iborat shartli belgilarning ro‘yxati; chiqish - har bir toifaning chastotasi va xatoli belgilar soni; cheklov - bo‘sh satr va ruxsat etilmagan belgi alohida qayd qilinadi. Birinchi test sifatida [A, B, A, C, B, A] qabul qilinadi va mos ravishda 3, 2, 1 qiymatlari kutiladi. Ikkinchi test bo‘sh ro‘yxat bo‘lib, barcha chastotalar nolga teng bo‘lishi kerak. Uchinchi test [A, X, B] bo‘lib, X noma’lum belgi sifatida qayd etiladi. Qabul qilingan qoidalar oldindan berilganligi sababli har bir testning to‘g‘riligini o‘zaro taqqoslash mumkin.

Mazkur topshiriqning keyingi variantida bitta qayd bir necha maydondan tashkil topadi: shartli identifikator, toifa va sana. O‘quvchi identifikatorlarni takrorlanishiga doir qoidani aniqlaydi: bir xil identifikator va toifa takroriy yozuvmi yoki mustaqil hodisami? Ushbu savol dastur kodidan oldin ma’lumot modeli to‘g‘risida qaror qabul qilishni talab etadi. Baholashda aynan shu qaror mazmunan asoslanganligi tekshiriladi. Bir nechta to‘g‘ri algoritm mumkinligi e’tirof etiladi, ammo ularning har biri dastlab belgilangan qabul mezonlariga mos bo‘lishi shart. Shu orqali algoritmik tafakkurning muqobil yechimlarni qiyoslash elementi rivojlantiriladi.

Transfer topshirig‘i uchun o‘quvchiga avvalgi vazifa bilan tarkibiy jihatdan yaqin, lekin tashqi ko‘rinishi boshqa masala taqdim etiladi. Masalan, murojaatlar toifalari o‘rniga o‘quv laboratoriyasi inventaridagi jihoz turlari yoki kutubxona kitoblarining mavzu bo‘yicha sanog‘i olinadi. Asosiy algoritmik g‘oya o‘zgarmasa-da, o‘quvchi kirish ko‘rinishi va natija spetsifikatsiyasini qayta tuzishi kerak bo‘ladi. Agar u eski kodni shunchaki ko‘chirsa va yangi cheklovlarni hisobga olmasa, transfer amalga oshgani haqida xulosa qilish mumkin emas. Shunday qilib, yangi kontekst didaktik bezak emas, egallangan usulning umumlashganligini tekshirish vositasiga aylanadi.

Sintetik ma’lumotlardan foydalanish o‘quvchini real shaxsiy yoki xizmat axboroti bilan ishlash zaruratidan xalos etadi. Shu bilan birga, vazifaning kasbiy mazmuni yo‘qolmaydi: ma’lumotni aniqlashtirish, xatoni qayd qilish, axborot sifati uchun mas’uliyat va natijani izohlash kabi umumkasbiy ko‘nikmalar rivojlantiriladi. Bu esa ta’limning kasbiy yo‘nalganligi bilan axborot xavfsizligi o‘rtasida mutanosiblikni ta’minlaydi.

Juftlikda kodni tahlil qilish, bir-biriga qarshi test topish va qarorni dalillash o‘quv faoliyatini faollashtirishi mumkin. Ammo jamoaviy dastur bitta o‘quvchining mustaqil kompetensiyasi dalili bo‘lmaydi. Shu bois hamkorlikdagi mashqdan so‘ng har bir ishtirokchi yangi parametr, o‘zgargan shart yoki mustaqil testga ega qisqa individual vazifani bajaradi. O‘qituvchi kodning qaysi qismi yordam bilan, qaysi qismi mustaqil yaratilganini qayd etadi.

Xatoning pedagogik ahamiyati ham qayta talqin qilinadi. “Dastur ishlamadi” degan umumiy bayonot o‘rniga o‘quvchi xato yuz bergan kirish qiymatini, kutilgan natijani, amaldagi natijani va tuzatish sababini keltiradi. Shunda xatolarni tuzatish tasodifiy urinishdan dalilga asoslangan tahlilga o‘tadi; testlash esa dasturni bajarib ko‘rish bilan cheklanmaydi.

Xatolar tipologiyasini o‘quv maqsadiga mos holda to‘rtta guruhga ajratish maqsadga muvofiq: sintaktik xato, bajarilish vaqtidagi xato, mantiqiy xato va masalani noto‘g‘ri formallashtirish. Sintaktik xatoda Python yozilish qoidasi buziladi; bajarilish xatosida dastur ma’lum kirishda to‘xtaydi; mantiqiy xatoda dastur ishlaydi, ammo natija noto‘g‘ri bo‘ladi; formallashtirish xatosida esa kod noto‘g‘ri tushunilgan talabni bajarayotgan bo‘lishi mumkin. Ushbu farqlar o‘quvchiga ‘xatoni topish’ degan umumiy ko‘rsatmadan ko‘ra aniqroq refleksiya

yo‘nalishini beradi. Baholash jarayonida xato mavjudligining o‘zi emas, uning sababi va tuzatish usuli ham tahlil qilinadi.

Testlash madaniyatini shakllantirish uchun har bir individual amaliy ishga kichik test protokoli biriktiriladi. Protokolda test raqami, kirish qiymati, oldindan kutilgan natija, haqiqiy natija, moslik holati va aniqlangan muammo yoziladi. O‘quvchi tekshiruvdan keyin kodga o‘zgartirish kiritsa, xatoni aniqlagan testdan tashqari avval muvaffaqiyatli o‘tgan testlarni ham qayta bajaradi. Bu takroriy tekshiruv bitta tuzatish boshqa funksiyani buzmaganini aniqlashga yordam beradi. Maktab darajasida murakkab avtomatlashtirilgan test muhitini talab qilmasdan ham shu tartib orqali dalillilik va takrorlanish madaniyatini shakllantirish mumkin.

Dissertatsiyada individual amaliy ish uchun olti mezonli, umumiy 18 balllik baholash rubrikasi taklif qilingan. U baholashning mazmunini faqat kod natijasiga bog‘lab qo‘ymasdan, masala shartni anglash, yechimni rejalashtirish va tekshirish madaniyatiga ham tarqatadi. Mezonlarni qo‘llashdan avval ularning mazmun validligi, baholovchilar o‘rtasidagi kelishuvi va parallel topshiriqlarning qiyosiy qiyinligi pilot tekshiruvda aniqlanishi lozim.

Olti mezonli rubrikaning mazmunini quyidagi tarkibda aniqlashtirish taklif etiladi: (1) masalani formallashtirish; (2) algoritmik rejalashtirish; (3) kodning funksional to‘g‘riligi; (4) testlash qamrovi; (5) xatolarni tahlil qilish va tuzatish; (6) yechimni asoslash hamda mustaqillik. Har bir mezon 0 dan 3 ballgacha baholanadi. Nol ball - faoliyat dalili yo‘q yoki vazifa bajarilmagan; bir ball - qisman yoki jiddiy yordam bilan bajarilgan; ikki ball - asosiy talab bajarilgan, biroq ayrim izoh yoki tekshiruvlar yetishmaydi; uch ball - talablarga mos, dalillar bilan asoslangan va mustaqil bajarilgan natija. Ta’riflar vazifa turi bo‘yicha oldindan spetsifikatsiyalanishi shart.

Rubrikani qo‘llashda funksional to‘g‘rilik bilan pedagogik mustaqillikni bir-biridan farqlash muhim. Masalan, dastur barcha berilgan testlardan o‘tishi, ammo uning asosiy qismi o‘qituvchi tomonidan yozib berilgan bo‘lishi mumkin. Bunday ishning kodga tegishli texnik bahosi yuqori chiqsa ham, mustaqillik mezonida ayni holat aks ettiriladi. Aksincha, o‘quvchi mustaqil to‘g‘ri algoritm tuzib, kichik sintaktik xatoga yo‘l qo‘ysa, uning algoritmik rejalashtirish bo‘yicha natijasi kodning bajarilish natijasidan alohida baholanadi. Bu yondashuv summativ ballning diagnostik foydasini oshiradi.

Baholovchilar kelishuvini ta’minlash uchun ikki pedagog mustaqil ravishda bir nechta anonimlashtirilgan ishni rubrika bo‘yicha tekshirishi, keyin ball farqining sababini birgalikda muhokama qilishi mumkin. Mezonlar talqini bo‘yicha kelishmovchilik yuqori bo‘lgan joylarda tavsiflar va namunaviy ishlar qayta aniqlashtiriladi. Kichik hajmdagi kalibrlash jarayoni ham rubrikaning amaliy tushunarliligini oshiradi, biroq uning statistik ishonchliligini qat’iy tasdiqlamaydi. Buning uchun yetarli tanlanma, baholovchilararo moslik ko‘rsatkichi va takroriy tekshiruv zarur bo‘ladi. Maqoladagi rubrika shu nuqtayi nazardan sinab ko‘rilishi lozim bo‘lgan taklif sifatida taqdim etiladi.

Rubrika bo‘yicha 18 ballning o‘zi avtomatik tarzda «yuqori kompetensiya»ning universal chegarasini belgilamaydi. Bunday darajalarni joriy qilish uchun empirik taqsimot va mazmunli standart o‘rnatish jarayoni zarur. Shu bois baholashda o‘quvchining har bir harakati bo‘yicha dalilni ko‘rsatish, olingan yordam darajasini qayd etish va keyingi o‘quv qadamini belgilash ustuvor hisoblanadi.

Boshlang‘ich va yakuniy ishlarni taqqoslashda vazifalarning aynan bir xil bo‘lishi shart emas, lekin ular o‘lchaydigan harakatlar va murakkablik darajasi qiyoslanadigan bo‘lishi kerak. Birinchi vazifada ma’lumotlarni sanash, ikkinchisida esa fayl, interfeys va ma’lumotlar bazasini bir vaqtda qo‘llash talab qilinsa, ball farqi o‘quvchi rivojlanishi bilangina izohlanmaydi. Shuning uchun parallel shakllar spesifikatsiyasi avvaldan tayyorlanishi zarur.

Boshlang‘ich va yakuniy diagnostika uchun ikkita parallel topshiriq ishlab chiqishda nazorat qilinadigan xususiyatlar oldindan belgilanadi: dasturlash konstruksiyalari soni, ma’lumot turi, cheklovlar miqdori, algoritmik bosqichlar murakkabligi va talab etiladigan test turlari. Ikkala topshiriq turli predmet ssenariysiga ega bo‘lishi mumkin, lekin bir xil rubrika mezonlari bilan tekshirilishi zarur. Aks holda guruhlar yoki vaqt nuqtalari orasidagi ball tafovutiga faqat

metodikaning ta’siri sabab bo‘lganini asoslash qiyinlashadi. Diagnostik materiallarning teng qiyinligini aniqlash pilot tekshiruvda o‘quvchilar javoblari va bajarish vaqti bo‘yicha baholanishi maqsadga muvofiq.

Amaliy tajriba-sinov loyihalashtirilganda dastlabki tayyorgarlik, kompyuter bilan ishlash tajribasi, dars davomiyligi, texnik ta’minot va o‘qituvchi omili hisobga olinadi. Faqat yakuniy umumiy ballni taqqoslash yetarli bo‘lmaydi: alohida mezonlar bo‘yicha rivojlanish, topshiriqni bajarishga sarflangan vaqt va yordam darajasining o‘zgarishi ham tekshiriladi. O‘quvchilar bir sinfda ta’lim olgani sababli natijalar o‘zaro bog‘liq bo‘lishi mumkin; statistik tahlilda ushbu tuzilmani e’tibordan chetda qoldirmaslik kerak. Hozirgi maqolada bunday tajribaning birlamchi natijalari berilmagani uchun metodikaning sonli samaradorligi haqida xulosa chiqarilmaydi.

Muhokama. Taklif etilayotgan metodik modelning ilmiy-metodik afzalligini quyidagicha izohlashimiz mumkin:

- *birinchisi,* tushunchani bilish bilan uni yangi muammoga mustaqil tatbiq etishni ajratish;
- *ikkinchisi,* pedagogik yordam bilan bajarilgan faoliyatni mustaqil mahsulotdan farqlash;
- *uchinchisi,* mashg‘ulotni o‘tkazish faktini uning o‘quvchi kompetensiyasiga ta’siri haqidagi dalildan ajratishdir.

Mazkur farqlar baholashning mazmuniy aniqligini oshiradi. Metodikani amaliyotda tekshirish uchun tajriba va nazorat guruhlari bir xil o‘quv mazmuni hamda imkoni boricha teng o‘quv vaqti bilan ta’minlanishi, boshlang‘ich tayyorgarlik va texnik sharoit hisobga olinishi kerak. Umumiy balldan tashqari, ayrim harakatlar bo‘yicha natijalar, yordam turlari va ish fayllari qiyoslanishi lozim. Guruhlarning sinf doirasida o‘zaro bog‘liqligi, o‘qituvchi ta’siri va topshiriq variantlari farqini inkor etish mumkin emas. Demak, ishlab chiqilgan modelning tadqiqotchilik ahamiyati avvalambor uning tekshiriladigan va takrorlanadigan tarkibida: har bir darsning maqsadi, o‘quv harakati, vazifa spesifikatsiyasi, yordam darajasi, individual mahsulot va baholash dalili o‘zaro bog‘langan.

Taklif etilgan tizimning amaliy joriy etilishi uch darajada tashkil qilinishi mumkin. Dars darajasida o‘qituvchi vazifa, tayanch va qisqa baholashni boshqaradi; mavzu darajasida bir nechta dars topshiriqlari bir-biriga bog‘lanadi; bo‘lim darajasida esa o‘quvchi mustaqil dastur va uning test hisobotini yakuniy ish sifatida taqdim etadi. Shu ketma-ketlik har bir alohida mashqni umumiy kompetensiya maqsadiga bog‘laydi. Biroq dastur bo‘limining barcha soatlarini bitta algoritm turiga sarflash to‘g‘ri emas: asosiy konstruksiyalar bo‘yicha takrorlash, o‘zgartirish va yangi vaziyatga ko‘chirish orasidagi muvozanat saqlanishi kerak.

O‘qituvchining roli tayyor javobni namoyish etishdan muammoni o‘rganish uchun sharoit yaratishga o‘tadi. U noto‘g‘ri natijani darhol to‘g‘rilab bermasdan, o‘quvchidan test qiymatini ko‘rsatishni, kutgan va olgan natijalarini solishtirishni va taxminini izohlashni so‘raydi. Bunday muloqotning didaktik qiymati o‘quvchini passiv kuzatuvchidan tekshiruvchi subyektga aylantirish bilan belgilanadi. Shu bilan birga, ortiqcha yordam ham, mutlaqo yordamsiz qoldirish ham maqsadga muvofiq emas; metodikaning samarali qo‘llanishi o‘quvchining tayyorgarlik holatini muntazam kuzatishni talab qiladi.

Modelning qo‘llanish chegaralari ham e’tiborga olinishi lozim. Birinchidan, boshlang‘ich algoritmik bilimlari yetarli bo‘lmagan o‘quvchiga murakkab mustaqil loyiha vazifasini darhol berish kognitiv yuklamani oshirishi mumkin. Ikkinchidan, faqat juftlikdagi umumiy kodni baholash individual natijalarni yashiradi. Uchinchidan, sintetik kasbiy vaziyatlar real xizmat jarayonini to‘liq aks ettirishi shart emas va bunga intilish ham o‘quv maqsadidan chetlashtirishi mumkin. To‘rtinchidan, rubrika mezonlari mahalliy sinovsiz universal daraja chegarasi sifatida ishlatilmasligi kerak. Mazkur cheklovlarning ochiq ko‘rsatilishi modelni takomillashtirish va qayta tadqiq etish imkoniyatini oshiradi.

Keyingi tadqiqotlar uchun taklif etilayotgan tekshiruv mantig‘i quyidagicha: birinchi bosqichda mutaxassislar yordamida topshiriq va rubrika mazmuni ekspertizadan o‘tkaziladi; ikkinchi bosqichda pilot sinfda topshiriqlarning tushunarliligi va bajarilish muddati sinovdan o‘tkaziladi; uchinchi bosqichda parallel diagnostika vositalari bilan boshlang‘ich va yakuniy holat solishtiriladi; to‘rtinchi bosqichda natijalar yordam darajasi, individual xatolar va transfer

topshiriqlari bilan birgalikda talqin etiladi. Ana shunday ketma-ketlikda nazariy asoslangan g‘oyadan dalillar bilan tekshirilgan pedagogik xulosaga o‘tish mumkin.

Xulosa. Ichki ishlar akademik litseyi o‘quvchilarining dasturlash kompetensiyasini shakllantirish masalani formallashtirish, algoritmni rejalashtirish, kod yaratish, yechimni turli holatlarda testlash va natijani asoslashdan tashkil topgan yaxlit, kuzatiladigan amaliy faoliyat sifatida talqin qilinishi maqsadga muvofiq.

Python bo‘limi uchun asoslangan bosqichli metodika o‘quvchini kodni anglab o‘qishdan uni mustaqil o‘zgartirish va yangi dastur yaratishga olib boradi. Vazifaning kasbiy mazmuni sintetik ma’lumotlar orqali algoritmik qarorga ta’sir qiladi, pedagogik yordam esa o‘quvchining tayyorgarligi va ishlash darajasiga muvofiq kamaytiriladi. Bunday tashkil etishda o‘qitish usuli bilan mustaqillikni baholash mezoni o‘rtasida ichki muvofiqlik yuzaga keladi.

Olti mezonli 18 ballik rubrika va boshlang‘ich-yakuniy individual topshiriqlar kompetensiyani mezonli kuzatish uchun metodik asos yaratadi. Biroq ushbu vositalarning validligi, baholovchilar kelishuvi hamda metodikaning ta’sir kuchi haqidagi qat’iy xulosa to‘liq birlamchi ma’lumotlar asosidagi alohida empirik tekshiruvdan keyin berilishi kerak. Shunday qilib, taklif etilayotgan yechimning asosiy natijasi litsey dasturiga mos, pedagogik jihatdan asoslangan va ilmiy tekshirishga ochiq o‘qitish-baholash tizimining ishlab chiqilganidir. Shuningdek, metodikaning tashkiliy jihatdan amalga oshirilishi uchun o‘qituvchiga vazifa spetsifikatsiyasi, yordamni qayd etish varaqasi, test protokoli va mezonli rubrika kabi qisqa, lekin izchil hujjatlar tavsiya etiladi. Ular qo‘shimcha hujjatbozlik maqsadida emas, o‘quvchi faoliyati bilan baholash xulosasi o‘rtasidagi bog‘lanishni ochiq ko‘rsatish uchun xizmat qiladi. Yangi algoritmik vazifaga o‘tish faqat dastur ishlagani asosida emas, balki o‘quvchining yechimni tushunishi, tekshirishi va o‘zgargan shartda qo‘llay olishi haqidagi dalillar bilan asoslanadi.

Foydalanilgan adabiyotlar

1. IIV akademik litseylari aniq fanlar yo‘nalishi. Informatika va axborot texnologiyalari fanidan o‘quv dasturi. - Toshkent, 2025. – 22 B.

2. Wing J. M. Computational Thinking // Communications of the ACM. - 2006. Vol. 49, No. 3. - P. 33–35.

3. Saeli M., Perrenet J., Jochems W. M. G., Zwaneveld B. Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective // Informatics in Education. - 2011. -Vol. 10, No. 1. - P. 73-88.

4. Sentance S., Waite J., Kallia M. Teaching computer programming with PRIMM: a sociocultural perspective // Computer Science Education. - 2019. - Vol. 29, No. 2-3. -P. 136-176. DOI: 10.1080/08993408.2019.1608781.

5. Brennan K., Resnick M. New frameworks for studying and assessing the development of computational thinking. - AERA Annual Meeting, Vancouver, 2012.

6. Qian Y., Lehman J. Students’ Misconceptions and Other Difficulties in Introductory Programming: A Literature Review // ACM Transactions on Computing Education. - 2017. - Vol. 18, No. 1. - Article 1. DOI: 10.1145/3077618.

7. Vuorikari R., Kluzer S., Punie Y. DigComp 2.2: The Digital Competence Framework for Citizens. - European Commission, JRC, 2022. DOI: 10.2760/115376.