O‘ZBEKISTON MILLIY UNIVERSITETI JIZZAX FILIALI

ISSN 2181-3884

ZAMONAVIY ILMIY-TEXNIK TADQIQOTLAR

xalqaro jurnali

Vol.2 N.5

2026

International Journal of Contemporary Scientific and Technical Research

Volume 2 Issue 5
2026

Zamonaviy ilmiy-texnik tadqiqotlar xalqaro jurnali

5-son
2026

Международный журнал современных научно-технических исследований

Том 2 Выпуск 5
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.

BO‘LAJAK INFORMATIKA O‘QITUVCHILARIDA ALGORITMIK FIKRLASHNI RIVOJLANTIRISHNING POLIPARADIGMAL METODIKASI

Abduqodirova Patmaxon Tursunboyevna

Andijon davlat pedagogika instituti, dotsent, (PhD).

Annotatsiya.

Maqolada bo‘lajak informatika o‘qituvchilarida algoritmik fikrlashni rivojlantirishning poliparadigmal metodikasi ilmiy-nazariy jihatdan yoritilgan. Unda “paradigma”, “dasturlash paradigmasi”, “poliparadigmallik” va “poliparadigmal metodika” tushunchalarining mazmun-mohiyati tahlil qilinib, ularning algoritmik fikrlashni shakllantirishdagi pedagogik imkoniyatlari ochib berilgan. Tadqiqotda imperativ-protseduraviy, obyektga yo‘naltirilgan, funksional, mantiqiy-deklarativ hamda hodisaviy-parallel dasturlash paradigmalarining o‘ziga xos xususiyatlari va ularni ta’lim jarayonida uyg‘unlashtirish masalalari ko‘rib chiqilgan. Shuningdek, poliparadigmal yondashuvning yaqin rivojlanish zonasi, aqliy harakatlarni bosqichma-bosqich shakllantirish, tajribaviy o‘rganish, bilimlarni transfer qilish va spiralsimon ta’lim kabi pedagogik-psixologik nazariyalar bilan o‘zaro aloqadorligi asoslangan.

Kalit so‘zlar: algoritmik fikrlash, poliparadigmal metodika, dasturlash paradigmasi, imperativ paradigma, obyektga yo‘naltirilgan dasturlash, funksional dasturlash, mantiqiy dasturlash, hodisaviy dasturlash, paradigmalararo transfer, bo‘lajak informatika o‘qituvchisi.

ПОЛИПАРАДИГМАЛЬНАЯ МЕТОДИКА РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ У БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ

Абдукодирова Патмахон Турсунбоевна

Андижанский государственный педагогический институт, доцент, (PhD).

Аннотация.

В статье с научно-теоретической точки зрения рассматривается полипарадигмальная методика развития алгоритмического мышления у будущих учителей информатики. Анализируются сущность и содержание понятий «парадигма», «парадигма программирования», «полипарадигмальность» и «полипарадигмальная методика», раскрываются их педагогические возможности в формировании алгоритмического мышления. В исследовании рассматриваются специфические особенности императивно-процедурной, объектно-ориентированной, функциональной, логико-декларативной и событийно-параллельной парадигм программирования, а также вопросы их интеграции в образовательный процесс. Кроме того, обосновывается взаимосвязь полипарадигмального подхода с такими психолого-педагогическими теориями, как теория зоны ближайшего развития, теория поэтапного формирования умственных действий, теория обучения на основе опыта, теория переноса знаний и концепция спирального обучения.

Ключевые слова: алгоритмическое мышление, полипарадигмальная методика, парадигма программирования, императивная парадигма, объектно-ориентированное программирование, функциональное программирование, логическое программирование, событийно-ориентированное программирование, межпарадигмальный перенос, будущий учитель информатики.

A MULTI-PARADIGM METHODOLOGY FOR DEVELOPING ALGORITHMIC THINKING IN PROSPECTIVE COMPUTER SCIENCE TEACHERS

Abduqodirova Patmaxon Tursunboyevna

Andijan State Pedagogical Institute, Associate Professor, (PhD).

Abstract.

This article provides a scientific and theoretical examination of a multi-paradigm methodology for developing algorithmic thinking in prospective computer science teachers. It analyzes the essence and content of the concepts of "paradigm," "programming paradigm," "multi-paradigm approach," and "multi-paradigm methodology," highlighting their pedagogical potential for fostering algorithmic thinking. The study examines the distinctive characteristics of imperative-procedural, object-oriented, functional, logic-declarative, and event-driven parallel programming paradigms, as well as the possibilities of integrating them into the educational process. Furthermore, the relationship between the multi-paradigm approach and psychological and pedagogical theories, including the zone of proximal development, the step-by-step formation of mental actions, experiential learning, knowledge transfer, and spiral learning, is theoretically substantiated.

Keywords: algorithmic thinking, multi-paradigm methodology, programming paradigm, imperative paradigm, object-oriented programming, functional programming, logic programming, event-driven programming, cross-paradigm transfer, prospective computer science teacher.

Kirish. "Paradigma" atamasi ilmiy muomalaga keng ma'noda T. Kun tomonidan kiritilgan bo'lib, u ilmiy hamjamiyat tomonidan qabul qilingan va ma'lum davr mobaynida muammolarni qo'yish hamda hal etishning namunasi bo'lib xizmat qiladigan nazariy-metodologik andozalar yig'indisini anglatadi [1]. Kun talqinida paradigma faqat nazariya emas, balki ilmiy faoliyat uslubi, qadriyatlar va namunaviy yechimlar tizimidir. Aynan shu jihati, ya'ni paradigmaning muammoni ko'rish va yechish usulini belgilashi uni pedagogik tadqiqotlarda, jumladan, algoritmik fikrlashni rivojlantirish metodikasida qo'llash uchun asos bo'ladi.

Informatikada "paradigma" tushunchasi R. Floyd tomonidan 1978-yilgi Tyuring mukofoti ma'ruzasida fanga kiritilgan. U dasturlash paradigmasini dasturchining yechim qurish usullari, ya'ni muammoni tahlil qilish va algoritmik shaklga keltirishdagi umumiy yondashuvlar majmui sifatida talqin qildi va dasturlash paradigmalarini o'zlashtirish dasturchining kasbiy ko'nikmalarini kengaytirishini ta'kidladi [2]. Keyingi tadqiqotlarda P. Van Roy dasturlash paradigmasini muayyan matematik nazariya yoki izchil tamoyillar to'plamiga asoslangan dasturlash yondashuvi sifatida ta'riflab, paradigmalar o'rtasidagi farq asosan hisoblash modeli (holat, hodisa, qaramlik, bir vaqtlilik kabi tushunchalarning bor-yo'qligi) bilan belgilanishini ko'rsatdi [3]. Demak, paradigmalar o'rtasidagi farq sintaksisdagi yoki kalit so'zlardagi farq emas, balki "hisoblash nima va qanday sodir bo'ladi" degan savolga berilgan turlicha konseptual javobdir.

Dasturlashning asosiy paradigmalari tarixiy va mantiqiy jihatdan quyidagicha shakllangan.

Imperativ-protseduraviy paradigma fon Neyman arxitekturasi bilan chambarchas bog'liq bo'lib, hisoblashni xotira holatini ketma-ket buyruqlar yordamida o'zgartirish jarayoni sifatida modellashtiradi. Uning asosiy tushunchalari: o'zgaruvchi, qiymat berish, tarmoqlanish, takrorlanish, protsedura. E. Deykstra va boshqalar tomonidan ishlab chiqilgan tuzilmali dasturlash g'oyasi bu paradigmani intizomlashtirdi: dastur uchta bazaviy boshqaruv tuzilmasi — ketma-ketlik, tanlash va takrorlash orqali quriladi, algoritmning to'g'riligi esa qadamma-qadam asoslanadi [4]. N. Virtning "Algoritmlar + ma'lumotlar tuzilmalari = dasturlar" formulasi algoritm va ma'lumotlar tuzilmasining uzviyligini ta'kidlab, imperativ paradigmaning markaziy didaktik g'oyasini ifodaladi [5].

Obyektga yo'naltirilgan paradigma murakkab tizimlarni o'zaro xabar almashuvchi obyektlar to'plami sifatida modellashtiradi. Inkapsulyatsiya, vorislik va polimorfizm kabi tushunchalar abstraksiya darajasini oshiradi, dasturiy yechimni predmet sohasining tushunchaviy tuzilmasiga yaqinlashtiradi [6]. Bu paradigma mas'uliyatni taqsimlash, interfeys va amalga oshirishni ajratish, loyihalash andozalari (design patterns) kabi yuqori darajadagi algoritmik-arxitekturaviy fikrlashni talab qiladi [7].

Funksional paradigma hisoblashni matematik funksiyalarni qo'llash va kompozitsiyalash jarayoni sifatida ko'radi. J. Bekus 1977-yilgi ma'ruzasida dasturlashni fon Neyman uslubidan "ozod qilish" zarurligini ko'rsatib, funksional uslubning algebraik xossalarini asosladi [8]. Funksional paradigmada o'zgarmaslik, sof funksiyalar, yuqori tartibli funksiyalar va rekursiya markaziy o'rin tutadi. J. Xyuz funksional dasturlashning ahamiyatini "modullilik" nuqtayi nazaridan asoslab, kichik funksiyalarni "yopishtiruvchi" mexanizmlar (yuqori tartibli funksiyalar va "dangasa" hisoblash) yechimning tuzilmasini tubdan o'zgartirishini ko'rsatgan [9]. H. Abelson va J. Sassmanning "Kompyuter dasturlarining tuzilishi va talqini" asari esa funksional va abstraksiyaga asoslangan fikrlashni informatika ta'limining markaziga qo'ygan [10].

Mantiqiy-deklarativ paradigma masalani "qanday hisoblash kerak" degan savol orqali emas, balki "nima haqiqat" va "qanday munosabatlar o'rinli" degan savollar orqali ifodalaydi. R. Kovalski o'zining mashhur formulasida "Algoritm = Mantiq + Boshqaruv" deya, algoritmning ikki mustaqil tarkibiy qismini — masalaning mantiqiy tavsifi va uni bajarish strategiyasini ajratdi [11]. Prologga o'xshash tillarda dasturchi faktlar va qoidalarni bayon qiladi, yechimni esa mantiqiy xulosa chiqarish mexanizmi (unifikatsiya va orqaga qaytish) topadi [12]. Bu paradigma bo'lajak o'qituvchida algoritmning "tavsif" va "bajarish" jihatlarini farqlash qobiliyatini rivojlantiradi.

Hodisaviy va parallel (bir vaqtda bajariladigan) paradigma hisoblashni hodisalarga javob beruvchi, o‘zaro ta’sirlashuvchi, ba’zan bir vaqtda bajariluvchi jarayonlar sifatida ko‘radi. Foydalanuvchi interfeysi, o‘yinlar, tarmoq ilovalari va robototexnika bilan ishlashda aynan shu paradigma yetakchi o‘rin tutadi. Scratch kabi blokli muhitlarda bolalar hodisa, parallellik va sinxronizatsiya tushunchalari bilan dastlab aynan shu paradigma orqali tanishadi [13].

Bu paradigmalar bir-birini inkor etmaydi. Zamonaviy amaliyotda kuchli dasturchi masalaning tabiatiga qarab paradigmalarni uyg‘unlashtiradi: Python, JavaScript, C#, Kotlin, Scala kabi tillar bir nechta paradigmani qo‘llab-quvvatlaydi. Shu sababli P. Van Roy dasturchi uchun bitta paradigmani bilish yetarli emasligini, turli paradigmalarning asosiy tushunchalarini bilgan mutaxassis esa masalani tabiiyroq va kamroq xato bilan yechishini ta’kidlaydi [14]. Krishnamurti va Fisler paradigmalarni o‘rgatish masalasini tahlil qilib, paradigmani faqat “til xususiyatlari to‘plami” emas, balki fikrlash usuli sifatida o‘rgatish dasturlash ta’limining yetuk bosqichi ekanini ko‘rsatadilar [15].

Yuqoridagilarga asoslanib, tadqiqotda quyidagi tushunchalar ishchi ta’rif sifatida qabul qilindi.

Dasturlash paradigmasi — hisoblashning muayyan modeliga tayanuvchi, masalani tahlil qilish, yechimni ifodalash va uning to‘g‘riligini asoslashning izchil tushunchalar va usullar tizimi.

Poliparadigmallik — bir nechta dasturlash paradigmasining tushunchaviy asoslarini o‘zlashtirish, ularni masalaning tabiatiga mos ravishda tanlash, bir paradigmadagi yechimni boshqasiga o‘tkazish (paradigmalararo transfer) va ularni uyg‘unlashtirish qobiliyati.

Poliparadigmal metodika — ta’lim maqsadlari, mazmuni, metodlari, vositalari va baholash tizimi shunday loyihalangan pedagogik tizimki, unda har bir asosiy dasturlash paradigmasi alohida “tafakkur obyektivi” sifatida o‘rganiladi, ularning ketma-ketligi va o‘zaro bog‘liqligi algoritmik fikrlashning yuqori darajalariga bosqichma-bosqich ko‘tarilishni ta’minlaydi.

Poliparadigmal yondashuvning pedagogik-psixologik asoslari bir necha nazariyaga tayanadi. Birinchidan, L. S. Vigotskiyning yaqin rivojlanish zonasi va ichkilashtirish g‘oyasiga ko‘ra, yuqori psixik funksiyalar ijtimoiy-madaniy vositalar (bu yerda — turli paradigmalar va ularning tillari) orqali shakllanadi [16]. Har bir paradigma muayyan fikrlash vositalari to‘plamini beradi, ularning ko‘pligi esa tafakkurning moslashuvchanligini oshiradi. Ikkinchidan, P. Ya. Galperin va N. F. Talizinalarning aqliy harakatlarni bosqichma-bosqich shakllantirish nazariyasiga muvofiq, harakat avval moddiylashtirilgan (yoki moddiy) shaklda, so‘ng nutq va aqlda bajariladi; bunda harakatning indikativ asosi qanchalik to‘liq bo‘lsa, o‘zlashtirish shunchalik sifatli bo‘ladi [17]. Paradigmaning hisoblash modeli aynan shunday indikativ asos vazifasini o‘taydi. Uchinchidan, konstruktivizm va D. Kolbning tajribaviy o‘rganish sikli (aniq tajriba — refleksiv kuzatish — mavhum konseptuallashtirish — faol tajriba) paradigmalararo taqqoslash orqali chuqur tushunishga erishish mumkinligini asoslaydi [18]. To‘rtinchidan, D. Perkins va G. Salomonning o‘rganishni ko‘chirish (transfer) nazariyasi ko‘rsatadiki, yuqori darajadagi (“high-road”) transfer ongli abstraksiyalash va turli kontekstlarda qo‘llashni talab qiladi [19]; poliparadigmal metodikada bu aynan paradigmalararo tarjima vazifalari orqali ta’minlanadi. Beshinchidan, J. Bruner tomonidan asoslangan spiralsimon o‘quv dasturi tamoyiliga ko‘ra, asosiy g‘oyalar o‘quv jarayonida murakkablik darajasi ortib borishi bilan qayta-qayta yoritiladi [20]; bu tamoyil paradigmalar ketma-ketligini loyihalashda tayanch bo‘ladi.

Shuningdek, poliparadigmal yondashuv tilning sintaksisiga emas, balki “notional machine” (tushunchaviy hisoblash mashinasi)ga e’tibor qaratadi. B. du Buley dasturlashni o‘rganishdagi qiyinchiliklar orasida aynan o‘quvchi uchun ko‘rinmas bo‘lgan “mashina”ning ishlash modelini tushunish eng muhim ekanini ko‘rsatgan [21]. J. Sorva esa tushunchaviy mashinani dasturni bajaruvchi abstrakt kompyuter sifatida ta’riflab, uni aniq o‘rgatish zarurligini asoslagan [22]. Har bir paradigma o‘z tushunchaviy mashinasiga ega: imperativ paradigmada — xotira katakchalari va ko‘rsatmalar ketma-ketligi; funksional paradigmada — ifodalarni almashtirish (reduksiya) modeli; mantiqiy paradigmada — unifikatsiya va qidiruv daraxti; hodisaviy paradigmada — hodisalar navbati va ishlov beruvchilar. Bo‘lajak o‘qituvchining bu modellarni farqlay olishi o‘quvchilarning xatolarini ularning noto‘g‘ri tushunchaviy modeli nuqtayi nazaridan tahlil qila olishining shartidir.

Poliparadigmal yondashuvning metodik qimmati yana shundaki, u kasbiy tayyorgarlikning “texnologik-pedagogik-predmet” bilimlari integratsiyasi (TPACK) g‘oyasiga mos keladi [23]. L. Shulmanning “pedagogik predmet bilimi” konsepsiyasiga ko‘ra, o‘qituvchi predmetni bilishi bilan birga uni o‘quvchilar uchun “o‘qitiladigan” shaklga keltira olishi lozim [24]. Informatika o‘qituvchisi uchun bu muayyan algoritmik g‘oyani turli ko‘rinishlarda (blokli, imperativ, funksional, mantiqiy) tushuntira olish, ularning har birining kuchli va kuchsiz tomonlarini bilish demakdir.

Demak, poliparadigmal yondashuv algoritmik fikrlashning bir tomonlamaligini bartaraf etishga, uni “bitta uslubdagi texnikaviy ko‘nikma”dan “uslublar orasida erkin harakatlanuvchi kasbiy tafakkur”ga aylantirishga xizmat qiladi.

Poliparadigmal yondashuvga nisbatan ilmiy-pedagogik hamjamiyatda bir qator asosli e’tirozlar bildirilishi mumkin; ularni oldindan tahlil qilish metodikaning cheklovlarini aniqlash va loyihada hisobga olish uchun zarur.

Birinchi e’tiroz — kognitiv yuklama. Bir necha paradigmani, ularning tillari va muhitlarini o‘rganish boshlovchi dasturchini ortiqcha yuklashi, natijada bilimlar yuzaki bo‘lib qolishi mumkin. Bu xavf haqiqiy, ammo u asosan paradigmalar bir vaqtda va til sintaksisi darajasida o‘rgatilganda yuzaga keladi. Metodikada esa paradigmalar ketma-ket, tushunchaviy mashina darajasida, sintaksis yukini kamaytiruvchi usullar (Parsons masalalari, tayyor namunalar, qisman to‘ldiriladigan kodlar) bilan kiritiladi; asosiy muhit sifatida bitta til (Python) saqlanadi [25].

Ikkinchi e’tiroz — chuqurlikning yo‘qolishi. “Hammasidan oz-ozdan” o‘rgatish hech bir paradigmani chuqur

o‘zlashtirmaslikka olib keladi, degan fikr bildiriladi. Metodikada bu xavf ikki yo‘l bilan kamaytiriladi: (a) har bir modul paradigmaning asosiy g‘oyalari (hisoblash modeli, tushunchaviy mashina, 4–6 ta markaziy tushuncha) bilan cheklanadi, texnik tafsilotlar kengaytiriladigan trayektoriyaga o‘tkaziladi; (b) bo‘lajak o‘qituvchi uchun maqsad “har bir paradigmada professional dasturchi” bo‘lish emas, balki paradigmalarning mohiyatini tushunish va ularni o‘qitish qobiliyatini egallashdir.

Uchinchi e’tiroz — maktab amaliyotiga mosligi. Maktab informatikasida funksional yoki mantiqiy paradigma kam uchraydi, shuning uchun ular o‘qituvchi uchun “ortiqcha” ko‘rinishi mumkin. Biroq ular maktabda bevosita o‘qitilmasa ham, o‘qituvchining fikrlash doirasini kengaytiradi: rekursiya, ro‘yxatlar bilan ishlash, deklarativ so‘rovlar (ma’lumotlar bazasi), qoidalarga asoslangan tizimlar maktab kursida turli shakllarda uchraydi; ularni chuqur tushungan o‘qituvchi bu mavzularni to‘g‘riroq tushuntiradi [26].

To‘rtinchi e’tiroz — vaqt cheklovi. O‘quv rejasida soatlar cheklangan bo‘lib, 252 soatlik kurs hamma muassasalarda mavjud bo‘lmasligi mumkin. Shu sababli metodikada “yadroviy” (M1, M2, M3, M6) va “kengaytiriladigan” (M4, M5) modullar ajratilishi mumkin; bunday holatda IV darajaga erishish yadro modullar doirasida (imperativ, obyektli, funksional paradigmalar) tashkil etiladi, mantiqiy va hodisaviy paradigmalar esa tanlov yoki qisqartirilgan shaklda o‘rgatiladi.

Bu e’tirozlarning tahlili ikki xulosaga olib keladi. Birinchidan, poliparadigmal metodikaning samaradorligi unda qancha paradigma qamrab olinishiga emas, balki qanday tashkil etilishiga — ketma-ketlik, tushunchaviy mashina ustuvorligi, transfer vazifalari va kognitiv yuklamani boshqarishga bog‘liq. Ikkinchidan, metodikaning moslashuvchan tuzilmasi (yadroviy va kengaytiriladigan modullar, uch trayektoriya) uni turli resurs imkoniyatlariga ega muassasalarda qo‘llash uchun asos yaratadi.

Algoritmik fikrlash tushunchasining talqinlari xilma-xil. D. Knut algoritmik fikrlashni matematik fikrlashdan farqlab, uning o‘ziga xos xususiyatlari sifatida jarayonni qadamlar ketma-ketligi shaklida ifodalash, holatlar o‘zgarishini kuzatish va dinamik tuzilmalar bilan ishlashni ko‘rsatgan [27]. S. Peypert o‘zining “Mindstorms” asarida kompyuterni “tafakkur uchun vosita” sifatida talqin qilib, bolalar dasturlash orqali o‘z fikrlash usullari haqida fikr yuritishni o‘rganishini asoslagan [28]. J. Ving 2006-yilda “hisoblash tafakkuri” tushunchasini ommalashtirib, uni muammolarni hal etish, tizimlarni loyihalash va inson xulq-atvorini tushunishda informatika fanining asosiy tushunchalaridan foydalanish sifatida ta’rifladi [29].

G. Fuchek algoritmik fikrlashni informatikani tushunishning kaliti sifatida ko‘rib, uning tarkibiga quyidagi qobiliyatlarni kiritgan: berilgan masalani tahlil qilish, masalani aniq bayon qilish, masalaga mos elementar harakatlarni topish, ularning yordamida to‘g‘ri algoritm qurish, barcha oddiy va maxsus holatlarni inobatga olish hamda algoritm samaradorligini oshirish [30]. K. Selbi va J. Vulard esa hisoblash tafakkurining ta’rifini aniqlashtirib, uning tarkibiy qismlari sifatida abstraksiya, dekompozitsiya, algoritmik fikrlash, baholash va umumlashtirishni ajratgan [31]. K. Brennan va M. Resnik blokli dasturlash muhitida hisoblash tafakkurining uch o‘lchamini — hisoblash tushunchalari, hisoblash amaliyotlari va hisoblash nuqtayi nazarlarini ajratgan [32]. Grover va Pea K–12 tizimidagi tadqiqotlarni sharhlab, hisoblash tafakkurining asosiy elementlari sifatida abstraksiya, algoritmlar, avtomatlashtirish, nosozliklarni topish va tuzatish, simulyatsiya va umumlashtirishni sanagan [33]. Shute va hamkasblari esa shu tushunchaning ko‘p sonli ta’riflarini tahlil qilib, uni “masalalarni hal etishning mantiqiy uslubi” sifatida dekompozitsiya, abstraksiya, algoritm dizayni, nosozliklarni tuzatish, takrorlash va umumlashtirish kabi tarkibiy qismlar bilan ifodalagan [34].

Mazkur ta’riflarni umumlashtirib, ular ikki umumiy cheklovga ega ekanini qayd etish mumkin. Birinchidan, ular asosan o‘quvchi yoki umumiy foydalanuvchi tafakkurini ta’riflaydi va o‘qituvchining kasbiy tomonini — algoritmik g‘oyani o‘qitish jarayonida qayta ishlash qobiliyatini alohida ko‘rib chiqmaydi. Ikkinchidan, ular paradigmalarning farqiga e’tibor qaratmaydi: “algoritm qurish” ko‘pincha yagona (odatda imperativ) uslub sifatida tushuniladi. Algoritmik fikrlashning poliparadigmal taksonomiyasi aynan shu ikki bo‘shliqni to‘ldirishga qaratilgan.

Mahalliy va postsovet ta’lim an’anasida algoritmik tafakkur masalasi A. P. Yershovning “dasturlash — ikkinchi savodxonlik” g‘oyasi bilan bog‘liq holda rivojlantirilgan: unda algoritmik madaniyat o‘qish va yozish bilan teng ahamiyatga ega umumiy madaniyat elementi sifatida talqin qilingan va maktab informatikasining mazmuni shu g‘oya asosida shakllantirilgan [35]. Informatika o‘qitish metodikasi bo‘yicha pedagogik oliy o‘quv yurtlari uchun yaratilgan darsliklarda ham algoritmik fikrlash maktab informatikasining bosh maqsadlaridan biri sifatida ko‘rsatiladi [36]. Biroq bu manbalarda poliparadigmal jihat, ya’ni turli hisoblash modellari orqali fikrlashni rivojlantirish alohida tadqiqot predmeti sifatida qaralmagan.

FOYDALANILGAN ADABIYOTLAR

1. Kuhn T. S. The Structure of Scientific Revolutions. – Chicago: University of Chicago Press, 1962.
2. Floyd R. W. The Paradigms of Programming // Communications of the ACM. – 1979. – Vol. 22, № 8. – P. 455–460.
3. Van Roy P. Programming Paradigms for Dummies: What Every Programmer Should Know // New Computational Paradigms for Computer Music. – Paris: Delatour France / IRCAM, 2009. – P. 9–47.
4. Dahl O.-J., Dijkstra E. W., Hoare C. A. R. Structured Programming. – London: Academic Press, 1972.
5. Wirth N. Algorithms + Data Structures = Programs. – Englewood Cliffs: Prentice-Hall, 1976.

6. Meyer B. Object-Oriented Software Construction. – 2nd ed. – Upper Saddle River: Prentice Hall, 1997.

7. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Reading, MA: Addison-Wesley, 1994.

8. Backus J. Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs // Communications of the ACM. – 1978. – Vol. 21, № 8. – P. 613–641.

9. Hughes J. Why Functional Programming Matters // The Computer Journal. – 1989. – Vol. 32, № 2. – P. 98–107.

10. Abelson H., Sussman G. J. Structure and Interpretation of Computer Programs. – 2nd ed. – Cambridge, MA: MIT Press, 1996.

11. Kowalski R. Algorithm = Logic + Control // Communications of the ACM. – 1979. – Vol. 22, № 7. – P. 424–436.

12. Sterling L., Shapiro E. The Art of Prolog: Advanced Programming Techniques. – Cambridge, MA: MIT Press, 1986.

13. Resnick M., Maloney J., Monroy-Hernández A., Rusk N., Eastmond E., Brennan K., Millner A., Rosenbaum E., Silver J., Silverman B., Kafai Y. Scratch: Programming for All // Communications of the ACM. – 2009. – Vol. 52, № 11. – P. 60–67.

14. Van Roy P. Programming Paradigms for Dummies: What Every Programmer Should Know // New Computational Paradigms for Computer Music. – Paris: Delatour France / IRCAM, 2009. – P. 9–47.

15. Krishnamurthi S., Fisler K. Programming Paradigms and Beyond // The Cambridge Handbook of Computing Education Research / eds. S. A. Fincher, A. V. Robins. – Cambridge: Cambridge University Press, 2019. – P. 377–413.

16. Vygotsky L. S. Mind in Society: The Development of Higher Psychological Processes. – Cambridge, MA: Harvard University Press, 1978.

17. Ершов А. П. Школьная информатика (концепции, состояние, перспективы). – Новосибирск: ВЦ СО АН СССР, 1987.

18. Kolb D. A. Experiential Learning: Experience as the Source of Learning and Development. – Englewood Cliffs: Prentice-Hall, 1984.

19. Perkins D. N., Salomon G. Transfer of Learning // International Encyclopedia of Education. – 2nd ed. – Oxford: Pergamon Press, 1992.

20. Bruner J. S. The Process of Education. – Cambridge, MA: Harvard University Press, 1960.

21. du Boulay B. Some Difficulties of Learning to Program // Journal of Educational Computing Research. – 1986. – Vol. 2, № 1. – P. 57–73.

22. Sorva J. Notional Machines and Introductory Programming Education // ACM Transactions on Computing Education. – 2013. – Vol. 13, № 2. – Article 8.

23. Mishra P., Koehler M. J. Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge // Teachers College Record. – 2006. – Vol. 108, № 6. – P. 1017–1054.

24. Shulman L. S. Those Who Understand: Knowledge Growth in Teaching // Educational Researcher. – 1986. – Vol. 15, № 2. – P. 4–14.

25. Parsons D., Haden P. Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses // Proceedings of the 8th Australasian Computing Education Conference (ACE 2006). – 2006. – P. 157–163.

26. Krishnamurthi S., Fisler K. Programming Paradigms and Beyond // The Cambridge Handbook of Computing Education Research / eds. S. A. Fincher, A. V. Robins. – Cambridge: Cambridge University Press, 2019. – P. 377–413.

27. Knuth D. E. Algorithmic Thinking and Mathematical Thinking // The American Mathematical Monthly. – 1985. – Vol. 92, № 3. – P. 170–181.

28. Papert S. Mindstorms: Children, Computers, and Powerful Ideas. – New York: Basic Books, 1980. – 230 p.

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

30. Futschek G. Algorithmic Thinking: The Key for Understanding Computer Science // Informatics Education – The Bridge between Using and Understanding Computers (ISSEP 2006). – Lecture Notes in Computer Science. – Vol. 4226. – Berlin: Springer, 2006. – P. 159–168.

31. Selby C., Woollard J. Computational Thinking: The Developing Definition. – Southampton: University of Southampton, 2013.

32. Brennan K., Resnick M. New Frameworks for Studying and Assessing the Development of Computational Thinking // Proceedings of the AERA Annual Meeting. – Vancouver, 2012.

33. Flavell J. H. Metacognition and Cognitive Monitoring: A New Area of Cognitive-Developmental Inquiry // American Psychologist. – 1979. – Vol. 34, № 10. – P. 906–911.

34. Shute V. J., Sun C., Asbell-Clarke J. Demystifying Computational Thinking // Educational Research Review. – 2017. – Vol. 22. – P. 142–158.

35. Ершов А. П. Школьная информатика (концепции, состояние, перспективы). – Новосибирск: ВЦ СО АН СССР, 1987.

36. Лапчик М. П., Семакин И. Г., Хеннер Е. К. Методика преподавания информатики. – Москва: Академия, 2001.