



BO‘LAJAK INFORMATIKA O‘QITUVCHILARIDA DASTURLASH KOMPETENSIYALARINI SHAKLLANTIRISHNING NAZARIY- METODIK ASOSLARI

Raximova Iroda Karim qizi

Buxoro davlat pedagogika instituti 2-bosqich magistranti

Annotatsiya

Mazkur tadqiqotda bo‘ljak informatika o‘qituvchilarida dasturlash kompetensiyalarini shakllantirishning nazariy-metodik asoslari tahlil qilindi. Tadqiqotning dolzarbligi raqamli transformatsiya, ta’limda sun’iy intellekt va hisoblash tafakkuri (computational thinking)ning kuchayib borayotgan ahamiyati bilan izohlanadi. Maqolada kompetensiyaviy yondashuv, algoritmik tafakkur, muammo yechish, dasturlashni o‘qitish metodikasi, raqamli ta’lim muhiti va blended learning kabi tushunchalar o‘zaro bog‘liq holda yoritildi. Metodologik jihatdan ish konseptual tahlil, qiyosiy tahlil, manbalarni sistematik saralash hamda zamonaviy empirik tadqiqotlar natijalarini sintez qilishga asoslandi. Natijalar shuni ko‘rsatdiki, dasturlash kompetensiyasi faqat sintaktik bilimlar yig‘indisi emas, balki algoritmlashtirish, abstraksiyalash, modellashtirish, debugging, refleksiya va pedagogik transforni qamrab oluvchi kompleks kasbiy kompetensiyadir. Problem-based learning, plugged va unplugged yondashuvlar uyg‘unligi, shuningdek, raqamli platformalar bilan qo‘llab-quvvatlangan blended learning modeli bo‘ljak o‘qituvchilarning dasturlashga oid bilim, ko‘nikma va o‘z-o‘zini baholash ko‘rsatkichlarini yaxshilashi ko‘rsatildi. Shuningdek, ta’lim dasturlarida ICT CFT va DigCompEdu kabi xalqaro kompetensiya ramkalari bilan uyg‘unlashgan metodik modullarni qo‘llash zarurligi asoslandi. Tadqiqot natijalari oliy pedagogik ta’limda informatika o‘qituvchilarini tayyorlash mazmunini yangilash hamda amaliy



mashg'ulotlarni qayta loyihalash uchun metodik asos bo'lib xizmat qiladi [1]–[5].

(ЮНЕСКО)

Kalit so'zlar: dasturlash kompetensiyasi, bo'lajak informatika o'qituvchisi, algoritmik tafakkur, computational thinking, problem-based learning, blended learning, raqamli ta'lim muhiti, kasbiy tayyorgarlik.

Kirish (Introduction)

Raqamli iqtisodiyot va ta'limning texnologik transformatsiyasi sharoitida bo'lajak informatika o'qituvchisidan nafaqat dasturlash tilini bilish, balki murakkab muammolarni algoritmlashtirish, yechimni modellashtirish, o'quv vaziyatiga moslashtirish va uni pedagogik jihatdan tushuntira olish talabi kuchaymoqda. Shu ma'noda dasturlash kompetensiyasi tor texnik ko'nikma emas, balki hisoblash tafakkuri, kasbiy refleksiya va didaktik loyihalashning integrativ natijasi sifatida qaralishi lozim [1]–[3]. (CMU School of Computer Science)

Jeannette Wing tomonidan ilgari surilgan computational thinking konsepsiyasi muammoni shakllantirish, tizimni loyihalash va yechimni kompyuter fanlari tamoyillari asosida tasavvur qilishni nazarda tutadi; zamonaviy talqinlarda esa bu yondashuv algoritmik tafakkur, abstraksiya, dekompozitsiya, naqshni aniqlash, testlash va debugging elementlari bilan boyitilgan [1], [7], [11]. O'qituvchi tayyorlash tizimi uchun bu ayniqsa muhim, chunki informatika o'qituvchisi dasturlashni "bajaruvchi" emas, balki uni "o'rgatuvchi va metodik talqin qiluvchi" subyekt sifatida shakllanishi zarur [7], [12], [14]. (CMU School of Computer Science)

Xalqaro darajada UNESCO ICT Competency Framework for Teachers hamda DigCompEdu kabi ramkalar o'qituvchining raqamli kompetensiyasini kasbiy faoliyat, ta'limni loyihalash, baholash va o'quvchilarni raqamli muhitda qo'llab-quvvatlash bilan bog'laydi [2], [3]. 2023-yilgi UNESCO GEM Report va OECD PISA 2022 materiallari esa texnologiya ta'limda samarali bo'lishi uchun u mazmun, pedagogika va insoniy o'zaro ta'sirga bo'ysundirilishi kerakligini ko'rsatadi; demak,



bo'lajak informatika o'qituvchisida dasturlash kompetensiyasini shakllantirish masalasi umumdidaktik va kasbiy tayyorgarlik doirasida ko'rilishi kerak [4], [5]. (ЮНЕСКО)

Shundan kelib chiqib, mazkur maqolaning maqsadi bo'lajak informatika o'qituvchilarida dasturlash kompetensiyalarini shakllantirishning nazariy-metodik asoslarini aniqlash, samarali pedagogik shart-sharoitlarni tavsiflash hamda empirik tadqiqotlar natijalari asosida amaliy tavsiyalar ishlab chiqishdan iboratdir [6], [14], [17].

Tadqiqot metodlari (Methods)

Mazkur tadqiqotda sifat va miqdoriy ilmiy adabiyotlar sintezi asosida konseptual-tahliliy yondashuv qo'llanildi. Manbalar sifatida 2018–2025 yillarda nashr etilgan Scopus/Web of Science indekslangan maqolalar, xalqaro kompetensiya ramkalari va ta'lim bo'yicha nufuzli hisobotlar tanlab olindi. Saralash mezonlari quyidagicha belgilandi: a) bo'lajak yoki amaldagi o'qituvchilarda computational thinking, programming, coding self-efficacy, blended learning yoki problem-based learning mavzulariga oid bo'lishi; b) metodologiyasi ochiq ko'rsatilgan bo'lishi; c) tekshiriladigan nashr va DOIga ega bo'lishi [2]–[6], [14], [17]. (ЮНЕСКО)

Tahlil bir necha bosqichda amalga oshirildi: birinchi bosqichda dasturlash kompetensiyasining tarkibiy komponentlari ajratildi; ikkinchi bosqichda pedagogik yondashuvlar — problem-based learning, experiential learning, plugged/unplugged integratsiyasi va blended learning — bo'yicha empirik natijalar qiyoslandi; uchinchi bosqichda bo'lajak o'qituvchi tayyorgarligiga ta'sir qiluvchi omillar, ya'ni self-efficacy, raqamli tayyorgarlik, pedagogik transfer va metodik refleksiya umumlashtirildi [7]–[18]. Ushbu maqoladagi "natijalar" bo'limi muallif tomonidan uydirilgan tajriba emas, balki mavjud empirik tadqiqotlar natijalari asosidagi ilmiy sintezdir [7], [8], [11], [12], [15]–[18]. (ScienceDirect)

Natijalar (Results)



Tahlil natijalari dasturlash kompetensiyasini kamida besh o‘zaro bog‘liq komponentda ko‘rish zarurligini ko‘rsatdi: 1) kognitiv-komputatsion komponent — algoritm, abstraksiya, dekompozitsiya va mantiqiy strukturalarni tushunish; 2) operatsion-amaliy komponent — kod yozish, testlash, debugging va iterativ takomillashtirish; 3) muammoli-faoliyat komponenti — real vaziyatni dasturlash masalasiga aylantirish; 4) pedagogik-metodik komponent — dasturlash mazmunini o‘quvchilarga tushuntirish va differensiallashtirish; 5) reflektiv-motivatsion komponent — self-efficacy, raqamli ishonch va kasbiy transfer [1], [7], [10], [13], [15]. ([CMU School of Computer Science](#))

Empirik tadqiqotlar shuni ko‘rsatadiki, problem-based learning va computational thinking integratsiyasi dasturlashga oid natijalarni sezilarli yaxshilaydi. Xususan, Ma va hammualliflarining tadqiqotida problem-solving instructional approach o‘quvchilarning CT ko‘nikmalari va self-efficacy ko‘rsatkichlarida ijobiy siljishni qayd etgan [8]. Kong, Lai va Sun esa uzluksiz professional rivojlantirish modeli asosida CT concepts va CT practices bo‘yicha katta effekt hajmi kuzatilganini ko‘rsatgan [7]. Broza va hammualliflari tadqiqotida esa pre-service teachers dasturlash faoliyati orqali abstraksiya, murakkab muammoni yechish va ijodkorlikni rivojlantirgani qayd etilgan [11]. ([digitalcommons.unf.edu](#))

Plugged va unplugged faoliyatlarni birlashtirgan modellar ham samarali ekanligi aniqlangan. Voon va hammualliflari 12 haftalik modulda 38 nafar bo‘lajak o‘qituvchining CT kompetensiyalari, ayniqsa CTni dars loyihasiga integratsiya qilish, asoslash va refleksiya qilish qobiliyatlari yaxshilanganini ko‘rsatgan [12]. Güler tadqiqotida algoritmik tafakkurni kompyutersiz o‘qitish prospective computer science teachers uchun foydali va zarur kurs sifatida baholangan [9]. Umutlu ishida esa bo‘lajak o‘qituvchilarning computational thinking va programming skills o‘rtasida mazmunli bog‘liqlik mavjudligi ko‘rsatilgan [10]. ([rptel.apsce.net](#))



Blended learning bo'yicha meta-tahlil natijalari ham ahamiyatli: Cao tadqiqotida blended learning talabalarning ko'rsatkichlari va o'quv yutuqlariga ijobiy ta'sir ko'rsatgani, umumiy akademik yutuqlar bo'yicha $d = 0.30$, umumiy ishlash natijasi bo'yicha esa $d = 0.50$ darajasida ustunlik qayd etilgani ko'rsatilgan [18]. Shu bilan birga, tayyorlik omili hal qiluvchi ekanligi aniqlangan: Çemçem, Korkmaz va Kukul ishlab chiqqan shkala o'qituvchining blended learningga tayyorligini 25 band va 4 omil orqali ishonchli baholash mumkinligini ko'rsatadi [16]. Demak, raqamli muhitning o'zi emas, balki unga metodik tayyorgarlik dasturlash kompetensiyasining shakllanishida asosiy vositachidir [4], [16], [18]. (Frontiers)

1-jadval. Dasturlash kompetensiyasini shakllantirishga oid ayrim empirik natijalar sintezi

Manba	Ishtirokchilar / dizayn	Yondashuv	Asosiy natija
Kong, Lai, Sun (2020) [7]	O'qituvchilar, ko'p bosqichli rivojlantirish dasturi	CT + programming pedagogy	CT concepts va CT practices bo'yicha sezilarli o'sish, katta effekt hajmi
Ma va boshq. (2021) [8]	Pretest–posttest, maktab konteksti	Problem-solving instructional approach	CT skills va self-efficacy ijobiy oshgan
Broza va boshq. (2023) [11]	Pre-service teachers	Constructionism + programming	Abstraksiya, complex problem solving va ijodkorlik rivojlangan
Voon va boshq. (2023) [12]	38 nafar bo'lajak o'qituvchi, 12 hafta	Plugged + unplugged experiential model	CTni dars dizayniga integratsiya qilish va refleksiya yaxshilangan



Cao (2023) [18]	Meta-tahlil, ko'p mamlakatli tadqiqotlar	Blended learning	Achievement: d=0.30; Performance: d=0.50
Sat, Kadirhan (2024) [15]	2 bosqichli validatsiya, CS teachers	Coding teaching self-efficacy	20 itemli CTSES, 4 o'lcham: motivatsiya, subject knowledge, class management, material development

Jadvaldagi ma'lumotlar ko'rsatadiki, dasturlash kompetensiyasini shakllantirishda yagona usul emas, balki muammoli faoliyat, refleksiya, raqamli platformalar va metodik transfering integratsiyasi yuqori samaradorlik beradi [7], [8], [11], [12], [15], [18]. ([ScienceDirect](#))

Muhokama (Discussion)

Olingan natijalar dasturlash kompetensiyasini shakllantirishda "til sintaksisini o'rgatish" modeli yetarli emasligini ko'rsatadi. Bu xulosa Wingning computational thinking haqidagi klassik qarashlari, UNESCO va DigCompEdu ramkalari, shuningdek Tondeur va hammualliflarining bo'lajak o'qituvchilarni raqamli texnologiyalar bilan o'qitishga tayyorlash bo'yicha yangilangan SQD-strategiyalariga mos keladi [1]–[3], [17]. Ayniqsa, pre-service teacher education bo'yicha adabiyotlar digital competence va pedagogical design birga rivojlantirilmasa, dasturlashga oid bilimlar o'quv amaliyotiga ko'chmay qolishini ko'rsatadi [14], [17]. ([CMU School of Computer Science](#))

Shuningdek, muhokama natijalaridan ko'rinadiki, PBL va blended learning samaradorligi avtomatik emas. UNESCO GEM Report texnologiya ta'limda maqsad emas, vosita ekanini ta'kidlaydi; OECD materiallari esa ijodiy va ilmiy muammo yechish ko'nikmalarining ochiq vazifalarda rivojlanishini ko'rsatadi [4], [5]. Demak, bo'lajak informatika o'qituvchilarida dasturlash kompetensiyasini rivojlantirish uchun o'quv topshiriqlari real muammoga yaqin, ko'p bosqichli,



refleksiv va baholash mezonlari aniq bo‘lishi zarur. Bu xulosa Hallingerning PBL bo‘yicha holat tahlili, Sundtjønn va hammualliflarining teacher educationdagi pedagogik amaliyotlar sharhi hamda Sat va Kadirhan tadqiqotidagi self-efficacy omillari bilan hamohangdir [6], [14], [15]. (ЮНЕСКО)

Xulosa (Conclusion)

Mazkur tadqiqotda bo‘lajak informatika o‘qituvchilarida dasturlash kompetensiyalarini shakllantirishning nazariy-metodik asoslari tizimlashtirildi. Xulosa sifatida quyidagilar aniqlandi: birinchidan, dasturlash kompetensiyasi algoritmik tafakkur, muammo yechish, kodlash, debugging, pedagogik transfer va refleksiyani qamrab oluvchi integrativ kasbiy tuzilma hisoblanadi; ikkinchidan, problem-based learning, experiential learning, plugged/unplugged integratsiyasi va blended learning mazkur kompetensiyani shakllantirishda yuqori metodik salohiyatga ega; uchinchidan, raqamli ta‘lim muhiti samarasi o‘qituvchining self-efficacy, digital readiness va metodik dizayn malakasi bilan belgilanadi; to‘rtinchidan, oliy pedagogik ta‘lim dasturlariga ICT CFT, DigCompEdu va CTga asoslangan modullarni kiritish amaliy ahamiyatga ega. Shu bois, informatika o‘qituvchilarini tayyorlashda “dasturlashni bilish”dan “dasturlashni o‘qita olish”ga yo‘naltirilgan kompetensiyaviy modelni joriy etish maqsadga muvofiqdir [2]–[5], [15]–[17]. (ЮНЕСКО)

Adabiyotlar ro‘yxati

1. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
2. UNESCO. (2018). UNESCO ICT Competency Framework for Teachers. Version 3. Paris: UNESCO.
3. Redecker, C. (2017). European Framework for the Digital Competence of Educators: DigCompEdu. Luxembourg: Publications Office of the European Union. <https://doi.org/10.2760/159770>



4. UNESCO. (2023). Global Education Monitoring Report 2023: Technology in education: A tool on whose terms? Paris: UNESCO.
5. OECD. (2024). PISA 2022 Results, Volume III: Creative Minds, Creative Schools. Paris: OECD Publishing.
6. Hallinger, P. (2023). Bibliometric review methodology and state of the science: Bibliometric review of research on problem-based learning, 2017–2021. *Interdisciplinary Journal of Problem-Based Learning*, 17(2), 1–27. <https://doi.org/10.14434/ijpbl.v17i2.35761>
7. Kong, S.-C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872. <https://doi.org/10.1016/j.compedu.2020.103872>
8. Ma, H., Zhao, M., Wang, H., Wan, X., Cavanaugh, T. W., & Liu, J. (2021). Promoting pupils' computational thinking skills and self-efficacy: A problem-solving instructional approach. *Educational Technology Research and Development*, 69(3), 1599–1616. <https://doi.org/10.1007/s11423-021-10016-5>
9. Güler, Ç. (2021). Algorithmic thinking skills without computers for prospective computer science teachers. *Journal of Theoretical Educational Science*, 14(4), 570–585. <https://doi.org/10.30831/akukeg.892869>
10. Umutlu, D. (2022). An exploratory study of pre-service teachers' computational thinking and programming skills. *Journal of Research on Technology in Education*, 54(2). <https://doi.org/10.1080/15391523.2021.1922105>
11. Broza, O., Biberman-Shalev, L., & Chamo, N. (2023). “Start from scratch”: Integrating computational thinking skills in teacher education program. *Thinking Skills and Creativity*, 48, 101285. <https://doi.org/10.1016/j.tsc.2023.101285>
12. Voon, X. P., Wong, S. L., Wong, L. H., Khambari, M. N. M., & Syed-Abdullah, S. I. S. (2023). Developing pre-service teachers' computational thinking through experiential learning: Hybridisation of plugged and unplugged approaches.



- Research and Practice in Technology Enhanced Learning, 18, 006.
<https://doi.org/10.58459/rptel.2023.18006>
13. Li, Q., McNary, S. W., & Boyd, T. (2023). Assessment of computational thinking: A study of preservice teachers' knowledge and beliefs. *Athens Journal of Sciences*, 10(2), 65–82. <https://doi.org/10.30958/ajs.10-2-1>
14. Sundtjønn, T., Aalbergsjø, S. G., Møller, T. E., & Schrøder, V. (2024). Review on pedagogical practices for computational thinking in teacher education: Characterizing an emerging field. *Nordic Journal of Comparative and International Education*, 8(4). <https://doi.org/10.7577/njcie.5742>
15. Sat, M., & Kadirhan, Z. (2024). Unveiling competence in the classroom: A multidimensional assessment of computer science teachers' self-efficacy in coding education. *Education and Information Technologies*, 29. <https://doi.org/10.1007/s10639-024-12503-7>
16. Çemçem, G. D., Korkmaz, Ö., & Kukul, V. (2024). Readiness of teachers for blended learning: A scale development study. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-024-12777-x>
17. Tondeur, J., Trevisan, O., Howard, S. K., & van Braak, J. (2025). Preparing preservice teachers to teach with digital technologies: An update of effective SQD-strategies. *Computers & Education*, 232, 105262. <https://doi.org/10.1016/j.compedu.2025.105262>
18. Cao, W. (2023). A meta-analysis of effects of blended learning on performance, attitude, achievement, and engagement across different countries. *Frontiers in Psychology*, 14, 1212056. <https://doi.org/10.3389/fpsyg.2023.1212056>
19. Lazarinis, F., Karachristos, C. V., Stavropoulos, E. C., & Verykios, V. S. (2019). A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies*, 24(2), 1237–1249. <https://doi.org/10.1007/s10639-018-9823-2>