

QO‘QON DAVLAT  
PEDAGOGIKA INSTITUTI  
ILMIY XABARLARI  
(2025-yil 2-son)



PEDAGOGIKA  
PEDAGOGY

UDK 372.862

Orcid: 0000-0002-1925-3708

## KOMPYUTER FANLARINI O‘QITISHDA JUFTLIK DASTURLASH PEDAGOGIK USULNING QO‘LLANILISHI

*Jamoldinov Saydillo Xasanovich,*  
Andijon Davlat Universiteti, assistent,  
[jsx78@mail.ru](mailto:jsx78@mail.ru), tel: +99890-545-50-25  
*Abdulatibov Mahammadtemur Sirojiddin o‘g‘li,*  
Andijon Davlat Universiteti, assistent,  
[jsx@rambler.ru](mailto:jsx@rambler.ru), tel: +99890-526-52-55  
*Yuldashev Madaminjon Muhammadqul o‘g‘li,*  
Andijon Davlat Universiteti, assistent,  
[yuldashevmadaminjon92@gmail.com](mailto:yuldashevmadaminjon92@gmail.com), tel: +99897-599-99-89

**Annotatsiya:** Ushbu maqolada biz turli didaktik modellar (semiotik zinapoya, kognitiv maqsadlar taksonomiyasi, muammolarni hal qilish) va zamonaviy o‘qitish usuli "juft dasturlash" yordamida dasturlash bilan bog‘liq kompetensiyalarga qanday erishish mumkinligini taqdim etamiz. Juftlik dasturlash standart usullardan farq qiladi. Ushbu usuldan foydalanish natijalari ham taqdim etiladi va tahlil qilinadi: dasturlash jarayonida qoniqish jihatlari va olingan bilim darajasi. Natijalar odatda ijobiydir va bu o‘qitish usulidan foydalanish imkoniyatlarini ko‘rsatadi.

**Kalit so‘zlar:** Indeks, trening, metod, juft dasturlash, semiotika, taksonomiya, kognitiv, muammo, ekstremal dasturlash, aspekt, istiqbol, dasturlash tillari.

## ПРИМЕНЕНИЕ ПЕДАГОГИЧЕСКОГО МЕТОДА ПАРНОГО ПРОГРАММИРОВАНИЯ В ПРЕПОДАВАНИИ КОМПЬЮТЕРНЫХ НАУК

**Аннотация:** В статье мы представляем, как достичь компетенций, связанных с программированием, используя различные дидактические модели (семиотическая лестница, таксономия когнитивных целей, решение проблем) и современный метод обучения «парное программирование». Парное программирование отличается от стандартных методов. Также представлены и проанализированы результаты использования этого метода: аспекты удовлетворенности во время программирования

и уровень полученных знаний. Результаты в целом положительные и показывают перспективность использования этого метода обучения.

**Ключевые слова:** Индекс, обучение, метод, парное программирование, семиотика, таксономия, когнитив, проблема, экстремальное программирование, аспект, перспективность, языки программирования.

## **APPLICATION OF THE PEDAGOGICAL METHOD OF PAIR PROGRAMMING IN TEACHING COMPUTER SCIENCES**

**Abstract:** In this article, we present how to achieve programming-related competencies using various didactic models (semiotic ladder, taxonomy of cognitive goals, problem solving) and a modern teaching method "pair programming". Pair programming differs from standard methods. The results of using this method are also presented and analyzed: aspects of satisfaction during programming and the level of knowledge gained. The results are generally positive and show the prospects for using this teaching method.

**Keywords:** Index, training, method, pair programming, semiotics, taxonomy, cognitive, problem, extreme programming, aspect, prospects, programming languages.

### **KIRISH.**

Hozirda ko‘plab axborot texnologiyalari bo‘limiga ega barcha oliy o‘quv yurtlarida kompyuter injiniringi yo‘nalishi bo‘yicha bo‘lajak professional o‘qituvchilar tayyorlanmoqda. Talabalar didaktik, pedagogik, matematik va boshqa fanlararo bilimlar bilan bir qatorda kompyuter injiniringi sohasida professional o‘qituvchilar uchun muhim bo‘lgan dasturlash bilim va ko‘nikmalarini egallaydi. Buning uchun biz o‘z tadqiqotimizda kompyuter injiniringi, ayniqsa, dasturlashni o‘rgatish bilan bog‘liq bo‘lgan professional o‘qituvchilarning bilim va ko‘nikmalarini belgilab beruvchi tanlangan kompetensiyalar to‘plamiga erishilganligini tahlil qilamiz. Kompyuter injiniringi sohasidagi professional o‘qituvchilar profiliga xos bo‘lgan kompetensiyalar qatorida informatika faniga oid umumiy kompetensiyalar va fanga oid kompetensiyalar mavjud [5].

Bo‘lajak kompyuter sohasi mutaxassislarini tarbiyalashda muhim mavzu dasturlashni o‘rgatishdir. Dasturlash va dasturlashni o‘rgatish butun ta’lim jarayonida turli fanlar orqali amalga oshiriladi. IT fakultetida "Dasturlashga kirish" fanining birinchi yilida dasturlash bo‘yicha kirish kursi amalga oshiriladi. Talabalar bilimi turli xil majburiy kurslar (masalan, Dasturlash-1 va dasturlash-2, algoritmlar va ma’lumotlar tuzilmalari) va tanlov kurslari (Java va Web dasturlash, Ta’limda axborot tizimlari va boshqalar) orqali chuqurlashtiriladi.

Ushbu tadqiqotda biz dasturlashni o‘rgatishning didaktik modellarini keltiramiz. Biz juft dasturlash (JD) kabi zamonaviy o‘qitish usullarini joriy etish bo‘yicha tajribamizni tushuntiramiz.

JD dasturiy ta’minotni ishlab chiqish uchun tezkor texnologiya va ekstremal dasturlashning muhim xususiyatidir [10]. Pedagogik nuqtai nazardan, JD hamkorlikda

o‘rganish va konstruktivizm bilan bog‘liq. So‘nggi bir necha yil ichida JD bir nechta nazorat ostida tajribalar mavzusi bo‘ldi. Bu "oddiy koddagi oddiy muammolar" uchun yaxshi ekanligi xabar qilingan. Oddiy muammolar uchun JD bitta dasturlashdan ko‘ra kamroq xatolarga olib keladi. Juftlashgan jamoalar tomonidan ishlab chiqarilgan kodni o‘qish va tushunish osonroq ekanligi taklif qilindi. Bu xatolarni aniqlash va texnik xizmat ko‘rsatishni osonlashtiradi. Adabiyotlarga ko‘ra, JD istiqbolli ish shaklidir, chunki juftlikda ishlaydigan talabalar imtihon va loyiha ballariga yakka o‘zi ishlaydigan talabalarga teng yoki undan yuqori ball oladilar, ular hamkorlikka ijobiy munosabatda bo‘lishadi va bir yildan so‘ng kompyuter fanlari bilan bog‘liq mutaxassisliklar bo‘yicha ro‘yxatdan o‘tish ehtimoli sezilarli darajada oshadi. Tadqiqot natijalari, shuningdek, juftlikda ishlaydigan talabalar mustaqil dasturlashni talab qiladigan keyingi dasturlash darslarida muvaffaqiyat qozonishda davom etishlarini ko‘rsatadi. Bizning holatda, JD tajribasi dasturlashning kirish kursida amalga oshirildi va so‘rov o‘tkazmoqdamiz. Biz JD usulidan foydalanish natijalarini turli tomonlardan baholaymiz: o‘quvchilarning shaxsiy qoniqishi, ularning dasturlash uchun keyingi motivatsiyasi, dasturchilarning o‘ziga ishonchi va juftlikda birgalikda ishlash natijasida olingan bilimlarning baholangan darajasi.

### **ADABIYOTLAR TAHLILI VA METODLAR.**

Axborot-kommunikatsiya texnologiyalari (AKT) va ta‘lim dasturlari endi ta‘lim jarayonlariga integratsiyalashgan. Garchi AKT kompyuter injiniringi ta‘limida muhim mavzu bo‘lsa-da [1], dasturlash boshidanoq o‘rgatish qiyin bo‘lgan fan hisoblangan.

Didaktika va kognitiv psixologiya sohalari tadqiqotchilari muvaffaqiyatsizliklar sabablarini o‘rganmoqdalar va ular asosiy sabablar o‘tmishda dasturlashni o‘rgatishda foydalanilgan didaktik modellarning nomukammalligida ekanligini aniqladilar. Universitet darajasida dasturlashni o‘rgatishning muvaffaqiyatli didaktik modellarini izlash ko‘plab zamonaviy tadqiqotlar mavzusiga aylandi. Ushbu maqolada "dasturlash didaktikasi" atamasi kengroq kontekstda qo‘llaniladi. Didaktik model quyidagilarga qaratilgan: [6].

- o‘quvchilarga o‘z bilimlarini og‘zaki bayon qilishlari uchun o‘rgatish kerak bo‘lgan meta-model;
- kursda o‘qituvchilarga o‘rgatish kerak bo‘lgan o‘qitish modeli, ular o‘z faoliyatini o‘qituvchilar bilan muvofiqlashtirishlari mumkin;
- dasturlashni o‘qitishni keyingi o‘rganish uchun tadqiqot savollarini shakllantirish uchun asos.

Nosek JD bo‘yicha dastlabki tadqiqotida ma‘lumotlar bazasining mustahkamligini tekshirish bo‘yicha professional dasturchilar, 5 kishi va 15 juftlikni o‘rgangan. Vazifani bajarish uchun zarur bo‘lgan vaqt 45 daqiqa bilan cheklangan. Barcha juftliklar individual dasturchilarni ortda qoldirdi. Jismoniy shaxslar uchun o‘rtacha bajarish vaqti 12 daqiqadan ko‘proq (41%) bo‘lsa-da, farq 5% darajasida statistik ahamiyatga ega emas edi [18].

Uilyams o‘zining dastlabki tajribalarida juft dasturchilar yakkaxon dasturchilarga qaraganda o‘rtacha 15% sekinroq ishlashlarini aniqladi, lekin ular 15% kamroq xatoga yo‘l

qo‘yishdi. Dasturlashda xatolarni qayta ishlash juda muhim masala va ular disk raskadrovka va sinov vaqtida qimmatga tushishi mumkin.

O‘z tadqiqotida Nawrocki va Voyciechowski kompyuter fanlari talabalari Shaxsiy dasturiy ta‘minot jarayonini dasturlash kursining dastlabki to‘rtta topshirig‘ini o‘rgandilar. Talabalar uch guruhga bo‘lingan: ulardan biri JDdan foydalangan. Umuman olganda, JD guruhi boshqa ikki guruhdan tezroq emas edi. Bu natija Uilyams va boshqalarning tadqiqotlaridan farq qiladi. Biroq, juft dasturlash guruhidagi o‘zgaruvchanlik boshqa ikkita guruhga qaraganda kamroq edi [18], [21].

IT bo‘limlarida dasturlashni o‘rgatish dasturlash tili sintaksisini tushunish, semantikani o‘rganish, algoritmik fikrlash, dastur yozish ko‘nikmalarini o‘rganish, misollar bo‘yicha o‘rganish, “muammo yechish”, guruh dasturlash va dasturlashni o‘rganishga asoslangan. Amalda biz quyidagi modellarning kombinatsiyasidan foydalanamiz:

*Semiotik zina* dasturlash tillarining tilga o‘xshash xususiyatlariga asoslanadi. O‘rganish va o‘qitish ketma-ketligi sintaksidan boshlanib, semantikaga o‘tadi va tilga o‘xshash vositalarning pragmatikasi bilan davom etadi. Uning mantiqiy asosi shundaki, sintaktik bilim biror narsani ifodalash uchun zarur va shuning uchun u lingvistik konstruktsiyalarning ma‘nosini o‘rganishdan oldin bo‘lishi kerak.

*Kognitiv maqsadlar taksonomiyasi* Bloomning kognitiv maqsadlar taksonomiyasiga o‘xshash o‘qitish strategiyasidan foydalangan. Trening ketma-ketligi dastur dasturidan foydalanish, dasturni o‘qish va dasturni o‘zgartirishni o‘z ichiga oladi. "Dastur yaratish" faolligini ham qo‘shish mumkin. Ushbu model bo‘lajak informatika o‘qituvchilari uchun mos keladi, chunki u algoritmlarning ma‘nosini bilvosita tushuntiradi. Bu ob'ektga yo‘naltirilgan dasturlashni o‘rgatish uchun ushbu modeldan foydalanish mumkinligi bilan izohlanadi;

*Muammoni hal qilish* o‘rganish qanday sodir bo‘lishiga konstruktivistik nuqtai nazardan kuchli turtki bo‘ladi. “Muammolarni yechish orqali talabalar o‘z tajribasi va amaliyot repertuarini kengaytirishlari kerak, jarayonning asosini esa dasturlash sohasidagi bilimlar tuzilishi tashkil etadi. Muammoni hal qilish jarayoni usullar va muhitlar tomonidan boshqariladi.” [16] Amalda biz tushuntirilgan modellarning kombinatsiyasidan foydalanamiz.

**Juftlik dasturlash va ekstremal dasturlash.** Juftlik dasturlash (JD) ikkita dasturchining bir kompyuterda birgalikda ishlash, bir loyiha, algoritmi, kod yoki test ustida hamkorlik qilish amaliyotini anglatadi.

AT bo‘limlari hamkorlikni o‘z ichiga olgan amaliyotlar to‘plamiga kiritish niyatida JDni sinfda rasmiy va norasmiy ravishda tobora ko‘proq tajriba o‘tkazmoqda. Shu bilan birga, JD - bu tezkor dasturiy ta‘minotni ishlab chiqish yoki ekstremal dasturlashning maxsus holati (XP) [10] Ekstremal dasturlash - bu dasturiy ta‘minotni ishlab chiqishning "engil metodologiyasi" deb ataladigan intizomi. U kichik guruhlar uchun mo‘ljallangan va oddiy amaliyotlar mavjud bo‘lganda butun jamoani birlashtirish orqali ishlaydi, jamoaga ularning qayerdaligini ko‘rish va amaliyotlarni o‘ziga xos vaziyatga moslashtirish uchun yetarli fikr-mulohazalarga ega.

JD hamkorlikda dasturlash uchun ishlatiladi. Juftlik dasturlashning o‘ziga xos xususiyati dasturchilar o‘rtasidagi psixologik va ijtimoiy o‘zaro ta’sirni o‘z ichiga oladi va shu bilan o‘yning biz odatiy yakkaxon dasturlash modelida ko‘rmaydigan noyob elementni olib keladi. Adabiyotlarga ko‘ra, o‘xshash va qarama-qarshi guruhlar o‘rtasidagi taqqoslash qarama-qarshi guruhning ko‘rsatkichlari o‘xshash guruhga qaraganda yuqori ekanligini tasdiqladi [12].

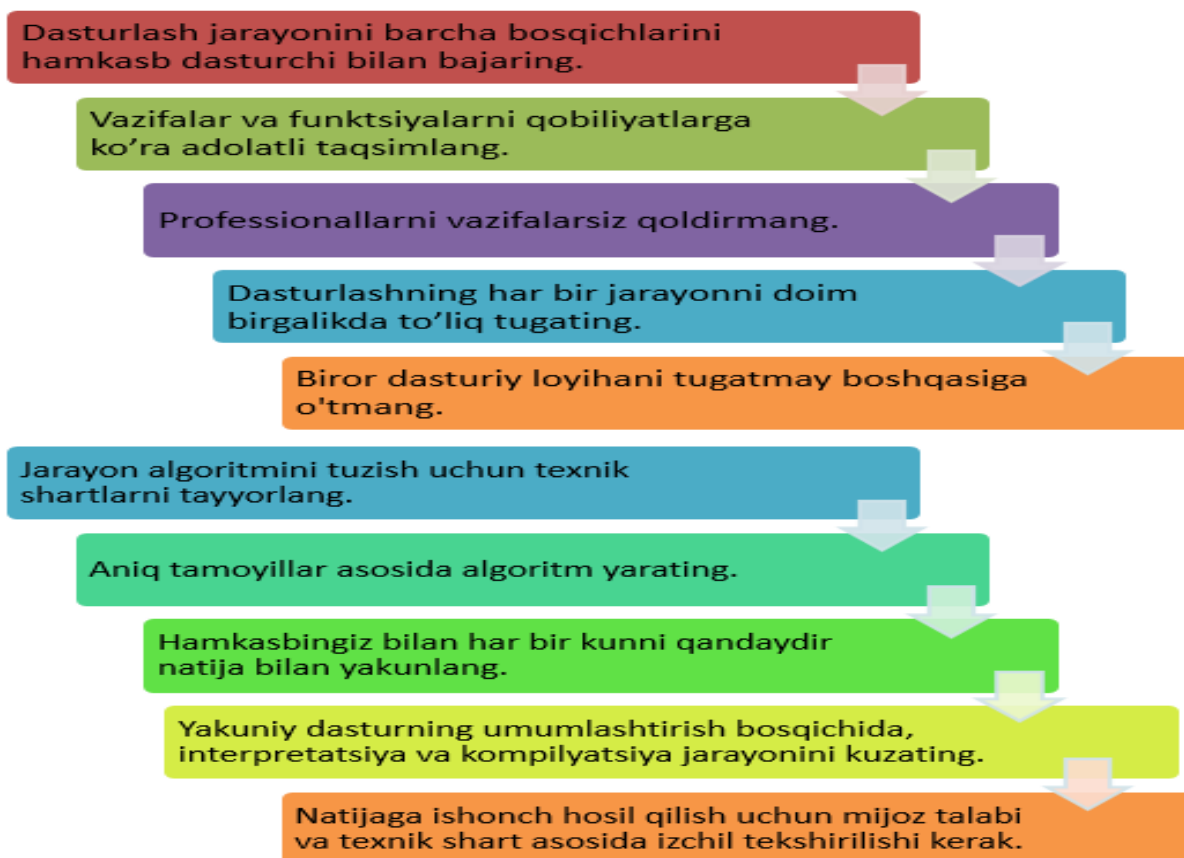
JD tamoyillari hamkorlikda o‘rganish tamoyillari bilan mos keladi:

- *Shaxsiyat va muloqot muhim ahamiyatga ega.*
- *Ishchi dasturiy ta’minot to‘liq hujjatlardan ko‘ra muhimroqdir.*
- *Ishtirokchilarni kiritish (hamkorlik) shartnoma muzokaralaridan ko‘ra muhimroqdir.*
- *O‘zgarishlarni hisobga olish jadvalga rioya qilishdan ko‘ra muhimroqdir.*

Pedagogik ishning ushbu shaklida ishtirokchilar muloqot, soddalashtirish, qayta aloqa, jasorat va ishonch kabi qobiliyat, ko‘nikma va qadriyatlarni rivojlantiradilar. Ular bo‘lajak informatika professor-o‘qituvchilarining malakasini oshirishda muhim ahamiyatga ega.

### NATIJALAR VA MUHOKAMA.

*Zamonaviy juftlik dasturlash.* Bu juftlik kompyuterda faol ravishda matn terish yoki dizayn yozuvchi haydovchi va haydovchining ishini kuzatuvchi va muammolarni sinchiklab aniqlab, takliflar bildiruvchi navigatoridan iborat. Ikkalasi ham doimiy ravishda miya hujumi sheriklaridir. Ushbu yo‘nalish muhim qoidalarini biz quyida keltirib o‘tamiz.(1-rasm)



**1-rasm.** Zamonaviy juftlik dasturlash eng muhim qoidalari.

*Manba : Tadqiqotlar asosidagi mualliflar ishlanmasi .*

*Juftlik dasturlashning afzalliklari va kamchiliklari.* JDni amalga oshirish natijasida, tadqiqotlarga ko‘ra, biz quyidagi afzalliklarni/kamchiliklarni tahlil qildik.(1-jadval)

1-jadval.

Juftlik dasturlashning afzalliklari va kamchiliklari.

<b>Afzalliklari:</b>	<b>Kamchiliklari :</b>
<ul style="list-style-type: none"> <li>- Ko‘proq intizom</li> <li>- Yaxshiroq kod (kamroq xatolar, tushunish osonroq)</li> <li>- Agile dasturiy ta‘minotni ishlab chiqish</li> <li>- Hamkorlar o‘rtasida bilim almashish</li> <li>- Yaxshi atmosfera</li> <li>- O‘zaro shaxsiy manbalar</li> <li>- Jamoa hamjihatligi ikkinchi</li> <li>- Bu juftlik tashqi muhitning buzilishlariga nisbatan kam sezgir</li> <li>- Kamroq kompyuter kerakligi</li> </ul>	<ul style="list-style-type: none"> <li>- Tajribasiz odamlarga ko‘rsatmalar berish</li> <li>- Tajribali dasturchilar mustaqil ravishda yaxshiroq ishlaydi</li> <li>- Tajribali dasturchi xatosiz kod yozadi va juftlikda bo‘lish foydasiz</li> <li>- Dasturlash uslubidagi farqlar konfliktlarning sababi hisoblanadi</li> <li>- Juftlik dasturlovchilar bir kishiga qaraganda kuniga bir soatdan kamroq dasturlashlari mumkin</li> <li>- SW korxonalarida uydan ishlaydigan dasturchilar JDni tushunishda qiynaladi</li> </ul>

*Manba : Tadqiqotlar asosidagi mualliflar ishlanmasi .*

*Dasturlashning kirish kursida juft dasturlashning qo‘llanilishi.*

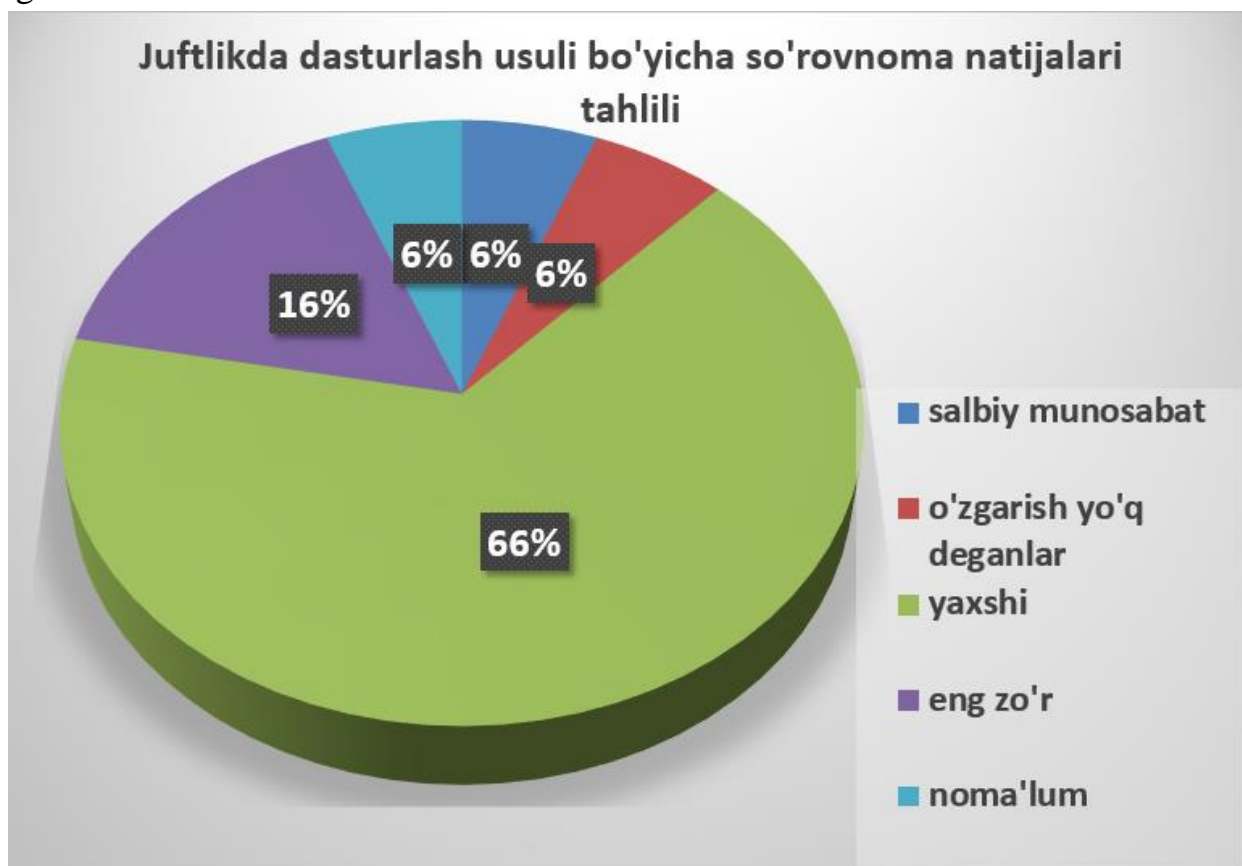
JD eksperimenti Andijon davlat universitetining IT fakultetida dasturiy ta‘minot bo‘yicha amaliy mashg‘ulotlar jarayonida (informatika fanining ikki fanidan o‘qishning birinchi kursi) o‘tkazildi. Kursning ikkinchi qismida (bizning yangi Boloniya o‘quv rejalarida dasturlash kursiga kirish qismiga aylantiriladi) biz o‘quvchilarimizni dasturiy ta‘minot ishlab chiqish bilan tanishtiramiz. Talabalar sxemalar va Python dasturlash tilining sintaksisi va semantikasi bo‘yicha oldindan o‘qitiladi. Kognitiv maqsadlar taksonomiyasi deb nomlangan dasturlash o‘qitishning didaktik modeliga ko‘ra, talabalar dasturlarni o‘zgartirishni va o‘z kodlarini yozishni o‘rganadilar. JD eksperimentini boshlashdan oldin biz talabalarning boshlang‘ich bilimlarini baholash uchun veb-test o‘tkazdik. Juftlik dasturlash paytida xulq-atvor qoidalarini muhokama qilishni davom ettirdik. Ikkita istisnolardan tashqari, talabalar mustaqil ravishda 37 juftlikka birlashtirildi. Er-xotinlar o‘zlarining ismlarini tanladilar. Tasodifiy generatordan foydalanib, biz har bir juftlikka keyingi ikki soat davomida bajarish uchun turli xil dasturlash mashqlarini tayinladik. Bir hafta o‘tgach, er-xotinlar o‘z dasturlarini taqdim etdilar va JD bilan o‘z tajribalarini bildirdilar, keyin esa ijobiy va salbiy tomonlarini muhokama qilishdi.

Onlayn so‘rovni JD tajribasida ishtirok etgan 74 talabaning 62 nafari to‘ldirdi.

*So‘rov natijalarini tahlil qilish.* Talabalarning JD haqidagi veb-saytini olish va JD talabalarning ba‘zi dasturlash tushunchalari haqidagi tushunchalarini yaxshilaganligini baholash uchun veb-so‘rov o‘tkazildi. Talabalar bilimi ikki ball bo‘yicha baholandi: JD usulini

qo‘llashdan oldin va keyin. So‘rovdan talabalar JD qoidalariga rioya qilganliklarini aniqlashga harakat qildik.(2-rasm)

Talabalar aniqlagan eng ijobiy tajriba bu shaxsiy dasturlash jihatini juftlikda kengaytirish edi. Ular ta’kidlagan eng ijobiy tajribalar orasida ish va shaxsiy ehtiyojlarni uyg‘unlashtirish, muloqot va ba’zi holatlarda odamlarning tajribasizligi tufayli zerikarli va ko‘p vaqt talab qiladigan dasturlash kiradi.



2-rasm. JD usuli bo‘yicha so‘rov natijalari.

Manba : Tadqiqotlar asosidagi mualliflar ishlanmasi .

JD usulining umumiy ko‘rinishini tahlil qilish biz uchun qiziqarli (2-rasm). Biz 82% JD bilan ijobiy tajribaga ega ekanligini aniqladik yoki ular JDni yaxshi yoki mukammal deb topdilar. Shu bilan birga, JD eksperimentida ishtirok etgan talabalarining 60% JDni amalga oshirgandan so‘ng, tez moslashish testini yechgandan so‘ng yaxshiroq test natijalarini ko‘rsatdi, bu esa dasturlash tushunchalarini yaxshiroq tushunishga erishilganligini ko‘rsatishi mumkin.

### XULOSA.

Samarali va sifatli dasturlash bo‘yicha treninglar katta muammodir. Ko‘pgina o‘qituvchilar o‘quvchilarning dasturlashga kirish kurslaridan keyin kompetentsiyaga erishishiga shubha qilishadi. Bularning barchasiga qaramay, ko‘plab talabalar o‘z majburiyatlarini ma’lum darajada bajaradilar. Bu holatni yangi didaktik va pedagogik usullarni joriy etish orqali hal qilish mumkin.

Ushbu maqolada biz dasturlashni o‘rgatishning eng mashhur didaktik modellarini taqdim etamiz va juft dasturlash deb ataladigan hamkorlik shaklini tushuntirdik.

JD ekstremal dasturlashning muhim xususiyati hisoblanadi. Talabalar guruhlarini bilan JD bilan ishlash tajribamiz shuni ko‘rsatdiki, o‘quvchilar juftlikda ishlashni foydali deb bilishadi. Xususan, biror xususiyatni amalga oshirishdagi harakatni baholash va kerakli va kerakli xususiyatlarni farqlash talabalar uchun juda qiyin edi. Biroq, biz talabalar JDdan foyda olishlarini aniqladik.

JD hamkorlikda ishlash usuli sifatida loyiha ishiga yaxshi kirishdir. Talabalar nafaqat zavqlanishadi, balki test ballari bo‘yicha yanada yaxshi natijalarga erishadilar. Ta‘lim nuqtai nazaridan, biz JD ham foydali ekanligini aniqladik, chunki u hamkorlik, integratsiya va bilimlarni moslashtirish bilan bog‘liq kompetentsiyalarni rivojlantiradi.

Shularni hisobga olib, biz kurslarimizda JD dan foydalanishni davom ettirishga va bu usulning afzalliklari va kamchiliklarini batafsilroq o‘rganishga qaror qildik.

### **ADABIYOTLAR RO‘YXATI.**

1. Lavrič, Osnove visokošolske didaktike (program za izpopolnjevanje): Uporaba informacijske komunikacijske tehnologije v izobraževanju, gradivo za udeležence, 4. del – V modul, CPI, FF, Ljubljana, maj 2007 (in Slovene).
2. ACM, Computing Curricula 2001, Computer Science — Final Report, The Joint Task Force on Computing Curricula, IEEE Computer Society Association for Computing Machinery, ACM, 15. 12. 2001.
3. Razdevšek Pučko in J. Rugelj, Kompetence v izobraževanju učiteljev, Vzgoja izob., 37(1), 34–41, 2006 (in Slovene).
4. E. Arisholm, H. Gallis, T. Dyba et al., Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise, Software Engineering, IEEE Transactions on Software Engineering, 33(2), 65–86, 2007.
5. Nančovska Šerbec i J. Rugelj, Analiza pridobljenih in zaželenih kompetenc dvopredmetnega študijskega programa »Računalništvo in ...«, Prispevki k posodobitvi pedagoških študijskih programov, Pedagoška fakulteta, Ljubljana, 78–90, 2006 (in Slovene).
6. J.J. Kaasbóll, Exploring Didactic Models for Programming, Tapir, 195–203, 1998.
7. J.J.Kaasbóll, Learning Programming, materials for rsubject Informatics didactics, University of Oslo, 2002.
8. J.J.Kaasbóll, Teaching Critical Thinking and Problem Defining Skills, Education and Information Technologies, 3(2), 101–117, 1998. [doi:10.1023/A:1009682924511](https://doi.org/10.1023/A:1009682924511)
9. J.Nawrocki and A. Wojciechowski, “Experimental Evaluation of Pair Programming”, Proceedings of the 12th European Software Control and Metrics s Conference, London, April, 2001, pp. 269– 276.
10. K. Beck, Extreme Programming Explained: Embrace Change, Reading, MA,



Addison-Wesley Professional, 1999.

11. K. M. Lui in K. C. C. Chan, Pair Programming Productivity: Novice-novice vs. expert-expert, *International Journal of Human Computer Studies*, 64, 915–925, 2006. [doi:10.1016/j.ijhcs.2006.04.010](https://doi.org/10.1016/j.ijhcs.2006.04.010)

12. K. S. Choi, F.P. Deek, I. Im, Exploring the Underlying Aspects of Pair Programming: The Impact of Personality, Information and Software Technology (2007),

13. L. A. Williams in R. R. Kessler, All I Ever Need to Know about Pair Programming I Learned in Kindergarten, *Communications of the ACM*, 43(5), 108–114, 2000. [doi:10.1145/332833.332848](https://doi.org/10.1145/332833.332848)

14. L. A. Williams, K. Yang, E. Wiebe, M. Ferzli, C. Miller, In Support of Pair Programming in Introductory Computer Science Course, *Computer Science Education*, 12(3), 197–202, 2002. [doi:10.1076/csed.12.3.197.8618](https://doi.org/10.1076/csed.12.3.197.8618)

15. L. A. Williams, C. McDowell, N. Nagappan, J. Fernald, L. Werner, Building Pair Programming Knowledge through a Family of Experiments, *Empirical Software Engineering*, 2003. ISESE 2003. Proceedings. 2003 International Symposium on Volume , Issue , 30 Sept.- 1 Oct. 2003, 143 – 152.

16. M. Pedroni, Teaching Introductory Programming with the Inverted Curriculum Approach, Diploma Thesis, Department of Computer Science, ETH Zurich, 2003.

17. Расулова, Т. П., Жамолдинов, С. Х., & Атаханов, К. (2020). Основы функционирования и эффективность внедрения систем мониторинга транспорта UzGPS в Узбекистане. *Universum: технические науки*, (7-1 (76)), 18-22.

18. Эркин, К., Жамолдинов, С. Х., & Бурунов, Б. А. У. (2018). Результаты внедрения современных информационных продуктов в сфере пассажирского транспорта. *Universum: технические науки*, (2 (47)), 5-8.

19. Jamoldinov S. Axborot mahsulotlari baholarining belgilanish xususiyatlari. “Mashinasozlik” ilmiy-texnika jurnali. №1, 2023 yil. 222-228.