OBYEKTGA YO‘NALTIRILGAN DASTURLASHGA NISBATAN KAM DARS AJRATILGAN SHAROITDA TALABALARDA OBYEKTGA YO‘NALTIRILGAN DASTURLASH KOMPETENSIYASINI SHAKLLANTIRISHDA REFAKTORING METODIKASINING SAMARADORLIGI

Orzimurod Hoshimov

Navoiy davlat universiteti tayanch doktoranti

orzimurodhoshimov78@gmail.com

Annotatsiya. Ushbu ilmiy tezisda respublikamiz oliy ta’lim muassasalaridagi "Axborot texnologiyalari va dasturlash" yo‘nalishlari o‘quv rejalarining real xususiyati — Obyektga yo‘naltirilgan dasturlash (OYD) alohida fan emas, balki umumiy dasturlash kurslari tarkibida 2-3 dars (4-6 soat) ko‘rinishida o‘tilishi sharoitidagi pedagogik muammolar tahlil qilingan. Vaqt tanqisligi sharoitida talabalarda protsedurali fikrlashdan OYD fikrlash tarziga o‘tishdagi to‘siqlarni bartaraf etish uchun mikro-loyiha metodikasi taklif etilgan.

Kalit so‘zlar: Obyektga yo‘naltirilgan dasturlash, vaqt tanqisligi, paradigma almashinuvi, Refaktoring, UML modellashtirish, kognitiv yuklama, amaliy kompetensiya.

Abstract. This scientific thesis analyzes the pedagogical challenges arising from the realistic curriculum structure in higher education institutions of Uzbekistan, where Object-Oriented Programming (OOP) is taught not as an independent course, but as a compressed module of 2-3 lectures (4-6 hours) within general programming modules. To overcome the paradigm shift obstacles under strict time constraints, a specialized methodology based on "Visual-First and Refactoring" techniques is proposed.

Keywords: Object-Oriented Programming, didactic compression, time constraint, paradigm shift, Refactoring, UML modeling, cognitive load, practical competence.

Zamonaviy dasturiy ta’minot muhandisligi va IT-sanoatining jadal rivojlanishi oliy ta’lim tizimida tayyorlanayotgan kadrlar sifatiga yangicha talablarni qo‘ymoqda. Bugungi kunda yirik dasturiy majmualar va korporativ tizimlarning aksariyati Obyektga yo‘naltirilgan dasturlash (OYD) tamoyillari va arxitekturaviy shablonlari asosida yaratilmoqda. Shu sababli, bo‘lajak dasturchilarda OYD muhitida ilovalarni loyihalash va ishlab chiqish bo‘yicha kasbiy kompetentlikni shakllantirish ta’lim tizimining ustuvor vazifalaridan biridir.

Biroq, oliy o‘quv yurtlari amaliyotida OYD asoslarini o‘qitish jarayoni bir qancha pedagogik hamda didaktik muammolarga duch kelmoqda. Ko‘plab talabalar dasturlashning bazaviy sintaksisini o‘zlashtirgan bo‘lsalar-da, real loyihalarni obyektlar, klasslar va ular o‘rtasidagi iyerarxik munosabatlar ko‘rinishida modellashtirishda jiddiy qiyinchiliklarga duch kelishmoqda. Bunga asosiy sabablardan biri — an’anaviy didaktik yondashuvlarning zamonaviy dasturlash paradigmalariga to‘liq mos kelmasligidir.

Respublikamiz OTMlarining "Dasturiy injeniring", "Axborot tizimlari va texnologiyalari" hamda "Kompyuter muhandisligi" bakalavriat yo‘nalishlaridagi amaldagi o‘quv rejalari tahlili shuni ko‘rsatadiki, aksariyat hollarda OYD fani alohida semestrlik fan sifatida o‘qitilmaydi. U, odatda, "Dasturlash 1-2" yoki "Algoritmlash va dasturlash asoslari" fanlarining yakuniy qismida atigi 2-3 ma’ruza hamda 2-3 amaliy mashg‘ulot (jami 4-6 akademik soat) hajmidagi qisqa modul ko‘rinishida o‘tiladi.

Aksariyat hollarda o‘qitish jarayoni hanuzgacha algoritmlashning ketma-ket (protsedurali) strukturasiga tayangan holda va passiv ma’ruza-amaliyot shaklida tashkil etilmoqda. Ushbu maqolada OYDni o‘qitishdagi an’anaviy didaktik metodlarning cheklangan jihatlari ochib beriladi hamda ularni bartaraf etishning

samarali metodik modeli taklif etiladi. Ushbu holat pedagogik jarayonda jiddiy ob'ektiv ziddiyatni (paradoksni) yuzaga keltiradi:

- **O‘quv fani murakkabligi:** OYD yuqori darajadagi kognitiv abstraksiyani, fikrlash tarzining tubdan o‘zgarishini (*protsedurali fikrlashdan obyektli fikrlashga o‘tish*) va murakkab sintaksisni talab qiladi.
- **Vaqt va resurs chegaralanganligi:** O‘qituvchida ushbu konseptual burilishni talaba shuurida shakllantirish uchun atigi 2-3 ta mashg‘ulot vaqti berilgan, xolos.

An'anaviy didaktika asosan o‘qituvchidan talabaga bilimlarni tayyor holda uzatish va akademik misollar orqali tushunchalarni mustahkamlashga tayanadi. OYD mavzusi esa yuqori darajadagi abstrakt va tizimli fikrlashni talab qiladi.

An'anaviy didaktika ushbu vaqt tanqisligi sharoitida ojiz qolmoqda. O‘qituvchilar 4-6 soat ichida OYDning 4 ta ustunini (Abstraksiya, Inkapsulatsiya, Vorislik, Polimorfizm) tezkor ravishda ma'ruza qilib berishga va standart sintaktik koding misollarini ko‘rsatishga majbur bo‘lishadi. Natijada talabalarda chuqur amaliy kompetensiya emas, balki yuzaki, chalkash va amaliyotda qo‘llab bo‘lmaydigan nazariy bilimlar yig‘indisi shakllanadi.

Shu sababli, amaldagi o‘quv rejalari bergan tig'iz vaqt sharoitiga moslashtirilgan, talabalarda OYD bo‘yicha bazaviy va arxitekturaviy kompetentlikni tezkor shakllantiruvchi yangicha didaktik yondashuvni ishlab chiqish pedagogika fani oldida turgan dolzarb vazifadir.

O‘quv rejasiga ko‘ra, talabalar OYDni o‘rganishdan oldin, odatda, C++, Java yoki Python kabi tillarda protsedurali (funksional) dasturlashni o‘rganadilar. Protsedurali yondashuvda diqqat-e'tibor *xatti-harakat va algoritmik ketma-ketlikka* (ma'lumot + algoritm = dastur) qaratiladi.

Vaqt va mazmun mutanosibligi buzilgan sharoitda an'anaviy ta'lim usullarini qo‘llash quyidagi salbiy pedagogik oqibatlarga olib keladi:

Kognitiv zo‘riqish: Insondagi qisqa muddatli ishchi xotira hajmi chegaralangan. Bir vaqtning o‘zida klass, obyekt, inkapsulyatsiya, merosxo'rlik,

polimorfizm, konstruktor, public/private kabi 8-10 lab yangi va abstrakt tushunchalarni 80 minutlik ma’ruzada sig‘dirishga urinish talabada axborot qabul qilish blokirovkasini keltirib chiqaradi.

Paradigma almashinuvining barbod bo‘lishi: Semestrning 80% qismida talaba protsedurali dasturlashni (ketma-ket ijro, oddiy funksiyalar, massivlar) o‘rganadi va uning mantiqiy xotirasida protsedurali mental model o‘rnashib qoladi. 14-haftaga kelib 2 ta darsda OYD o‘tilganda, talaba o‘z fikrlash tarzini o‘zgartirishga ulgurmaydi. U OYDni shunchaki *C++ yoki Python funksiyalarini klass degan qobiqqa o‘rab qo‘yish* deb tushunadi.

O‘qituvchi ko’p hollarda vaqt ozligi sababli "Moshina — bu klass, Spark — bu obyekt", "Odam — bu klass, Ali — bu obyekt" degan yuzaki hayotiy analogiyalar berish bilan cheklanadi. Natijada talaba real dasturlashda Mashina klassi interfeys yoki ma’lumotlar bazasi bilan qanday bog‘lanishini tushunmaydi va amaliy kodlashdan uzilib qoladi.

Laboratoriya ishlarida vaqt kamligi sababli talaba topshiriqni bitta main() funksiyasi ichida OYD tamoyillarini tamoman buzgan holda (barcha o‘zgaruvchilar public, inkapsulatsiyasiz) yozsa ham, dastur to‘g‘ri natija chiqarsa — maksimal ball qo‘yiladi. Bu esa talabada OYD arxitekturasiga nisbatan bepisandlikni shakllantiradi.

O‘quv rejasidagi 2-3 darslik (4-6 soatlik) modul doirasida maksimum amaliy natijaga erishish uchun biz "Visual-First va Refaktoring" (VFR) metodik modelini taklif etamiz. Ushbu model an’anaviy "nazariyadan-kodga" sxemasidan voz kechib, quyidagi 3 bosqichli amaliy mikro-siklga tayanadi:

1-bosqich. Sintaksisni yodlatishdan avval talabalarga tayyor dastur interfeysi ko‘rsatiladi. Talaba birinchi daqiqadanoq kod yozmaydi, balki tizim obyektlarini vizual ko‘rinishda ajratadi: Dastlab UML Class Diagram vositasida Klass uning atributlari hamda metodlari grafik tarzda chiziladi. Talaba xotirada klass va obyektning farqini, xotiradan joy ajratilishini diagrammalar orqali ko‘radi. Bu

kognitiv yuklamani keskin kamaytiradi va abstrakt tushunchalarni ko‘rgazmali qiladi.

2-bosqich. "Mashina/Hayvon" kabi mavhum misollar o‘rniga talabaga semestr davomida o‘rgangan bilimlariga mos bo‘lgan "Bank hisob-raqami va xavfsiz o‘tkazmalar" mikro-loyihasi beriladi. O‘qituvchi balance o‘zgaruvchisini public qilib qo‘yganda xaker yoki tashqi funksiya uni osongina manfiy qiymatga (-1000) o‘zgartirib qo‘yish xavfini amalda ko‘rsatadi. Talaba inkapsulatsiyani nazariy ta’rif sifatida emas, balki ma’lumot xavfsizligining amaliy ehtiyoji sifatida anglaydi va private hamda getter/setter usullarini anglagan holda qo‘llaydi.

3-bosqich. Talabalarga OYDni noldan yozdirish 2 soat ichida ko‘p vaqt yo‘qotilishiga olib keladi. Shu sababli Refaktoring (kodni qayta strukturalash) mashqi qo‘llaniladi: Talabalarga o‘zlari oldingi haftalarda yozgan protsedurali kodi (masalan, bitta ulkan *main()* funksiyali va global o‘zgaruvchili kod) tarqatiladi.

Topshiriq: Protsedurali kodni OYD tamoyillari (klasslar, atributlar va metodlar) bo‘yicha qayta strukturalash. Bu usul talabada protsedurali fikrlashdan OYD fikrlashiga o‘tishni bir necha soat ichida yaqqol namoyon etadi. Talaba protsedurali va OYD uslubining farqini, OYDning kodni qayta ishlatishdagi qulayligini amalda ko‘radi.

An’anaviy nazorat tizimida talabaning kodi asosan ikki mezon bo‘yicha baholanadi:

- ✓ Kodda sintaktik xatolik bormi yoki yo‘qmi?
- ✓ Dastur kutilgan natijani (output) to‘g‘ri chiqarmoqdami?

Ushbu yondashuv OYD mohiyatiga zid keladi. Dastur to‘g‘ri ishlashi mumkin, ammo u OYD tamoyillarini tamoman buzgan holda (masalan, barcha o‘zgaruvchilar *public* qilib qo‘yilgan, inkapsulyatsiya yo‘q, bitta klassga 10 xil mas’uliyat yuklatilgan) yozilgan bo‘lishi mumkin. An’anaviy baholash koding arxitekturasi va sifatini hisobga olmasligi sababli talabalarda noto‘g‘ri dasturlash ko‘nikmalari ildiz otadi.

Yuqorida ko‘rsatilgan didaktik to‘siqlarni bartaraf etish va talabalarning OYD muhitida ilovalarni ishlab chiqishga oid kompetentligini rivojlantirish uchun integratsiyalashgan metodik model taklif etiladi. Ushbu metodika quyidagi 4 ta asosiy ustunga tayanadi:

Protsedurali ta’limdan OYDga silliq o‘tish uchun "Didaktik inversiya" usulini qo‘llash tavsiya etiladi. Bunda o‘qitish sintaksisdan emas, balki tayyor obyektlar bilan ishlashdan boshlanadi. Talabalarga sintaksis o‘rgatilishidan oldin vizual muhitda tayyor obyektlar (masalan, GUI komponentlar — tugma, matn maydoni, shakllar) beriladi va ularning xususiyatlari (Properties) hamda metodlari (Methods) bilan shug’ullanish taklif etiladi. Obyekt tushunchasi aniq his qilingach, ularni birlashtiruvchi shablon — Klass tushunchasi kiritiladi. Shundan so‘nggina inkapsulyatsiya va ma’lumotlarni yashirish zaruriyati amaliy ehtiyoj sifatida tushuntiriladi.

Mavhum tushunchalarni bartaraf etishning eng samarali vositasi — UML (Unified Modeling Language) Class diagrammalaridir.

UML (Unified Modeling Language) Class diagrammasi obyektga yo‘naltirilgan tizimning arxitekturasini to‘g‘ri to‘rtburchaklar (bloklar) hamda ularni bog‘lovchi grafik chiziqlar orqali aks ettiruvchi strukturaviy diagrammadir.

- O‘qituvchi kod yozishdan oldin, talabalarga o‘quv topshirig‘ining UML diagrammasini chizdirishi lozim.
- Talaba klasslar o‘rtasidagi bog‘liqlikni avval vizual sxemada ko‘radi.
- Hozirgi zamonaviy IDE lardagi "Reverse Engineering" (koddan diagramma yoki diagrammadan kod generatsiya qilish) imkoniyatlaridan foydalanish nazariya va amaliyot o‘rtasidagi uzilishni yo‘qotadi.

Mavzu bo‘yicha alohida-alohida kichik mashqlarni bajarish o‘rniga, butun semestr davomida rivojlantirib boriladigan yagona o‘quv-amaliy loyihasi berilishi lozim (Masalan: *"Universitet kutubxonasini boshqarish tizimi"* yoki *"Onlayn do‘kon moduli"*). Mustaqil ta’lim va amaliy mashg‘ulotlarda ham tizim

obyektlarini aniqlash va bazaviy klasslarni yaratish, inkapsulyatsiya va ma’lumotlar xavfsizligini ta’minlash, vorislik orqali foydalanuvchi tiplarini (Admin, Talaba, O‘qituvchi) ajratish, polimorfizm va interfeyslar orqali to‘lov tizimlari yoki hisobot shakllarini ko‘p variantli qilish, ma’lumotlar bazasi va fayllar bilan integratsiya qilish, GUI interfeysini yakunlash kabi mashqlarni bajarish lozim.

Bunday bosqichma-bosqich loyiha yondashuvi talabada OYD shunchaki nazariya emas, balki real dasturiy mahsulot yaratish texnologiyasi ekanligi haqida mustahkam nuqtayi nazar va kompetentlikni shakllantiradi. Talabalarda arxitekturaviy fikrlashni rivojlantirish uchun ularga atayin "yomon yozilgan" dastur kodlari beriladi. Talaba ushbu koddagi OYD tamoyillari buzilgan bandlarni topishi, To‘g‘ri tamoyillarga moslab qayta qurishi, qayta qurilgan kodning moslashuvchanligi va o‘qishga oson ekanligini asoslab berishi kerak bo‘ladi. Ushbu amaliyot talabaning tanqidiy va mantiqiy fikrlashini keskin oshiradi.

Xulosa. Obyektga yo‘naltirilgan dasturlash mavzusi oddiy dasturlash sintaksisini o‘rgatish emas, balki yangi tizimli fikrlash va loyihalash falsafasidir. Shu sababli, uni an’anaviy didaktik usullar — mavhum analogiyalar, faktologik ma’ruzalar va tor doiradagi sintaktik baholash orqali o‘qitish kutilgan kasbiy natijani bermaydi. Protsedurali dasturlash va OYD o‘rtasidagi didaktik ko‘prikni to‘g‘ri qurish, ushbu mavzularni o‘quv rejalariga chuqurroq integratsiya qilish, talaba ishini baholashda faqat funksional natijaga emas, balki kodning OYD tamoyillariga mosligi, arxitekturaviy toza yozilganligiga e’tibor qaratish, mashg‘ulotlarni yakka tartibdagi kichik mashqlardan semestrlik yaxlit loyihalarga hamda tayyor kodlarni qayta ko‘rib chiqish shakliga o‘tkazgan ma’qul.

Oliy ta’lim muassasalarining dasturlash kurslarida OYDga atigi 2-3 dars ajratilishi sharoitida an’anaviy ma’ruza usuli va yuzaki "Mashina/Hayvon" analogiyalari samarasiz hisoblanadi.

Darsni murakkab sintaksisdan emas, balki UML Class Diagram va tayyor obyektlarning vizual modelini tahlil qilishdan boshlash lozim. Talabalarga OYDni noldan yozdirishga vaqt sarflamay, ularning oldingi bo‘limlarda yozgan protsedurali kodlarini OYD arxitekturasiga o‘tkazish topshiriqlarini berish lozim. Amaliy topshiriqlarni baholashda faqat funksional natijaga emas, balki koddagi inkapsulatsiya va arxitekturaviy tozalikka asosiy e’tiborni qaratish zarur.

Ushbu didaktik metodikaning amaliyotga joriy etilishi bo‘lajak IT-mutaxassislarning OYD muhitida amaliy ilovalar yaratish kompetentligini sifat jihatidan yangi bosqichga olib chiqadi.

Foydalanilgan adabiyotlar

1. Xodjabayev F. Talabalarning obyektga yo‘naltirilgan dasturlash tillariga oid kompetentligini rivojlantirish modeli // slib.uz. – 2024.
2. Mirsanov U.M. Uzluksiz ta’lim tizimida dasturlash tillarini o‘qitish nazariyasi va amaliyoti // Monografiya. – Navoiy, 2023. – 168 b.
3. Hoshimov O.P. Obyektga yo‘naltirilgan dasturlashni o‘qitishda kognitiv qiyinchiliklar: mavhum tushunchalarni anglash muammolari // *Scientific progress / Ilmiy tadqiqotlar jurnali*. — 2024. — Vol. 5, Issue 2. — B. 45-52.
4. Siddiqov B.X., Abdullayev N.M. *Dasturlash tillari va OYD texnologiyalari.* Darslik. — Toshkent: "TA'LIM", 2022. — 312 b.
5. Qosimov S.S. *Algoritmlash va dasturlash asoslari.* O‘quv qo‘llanma. — Toshkent: "Aloqachi", 2021. — 280 b.
6. Boltayev Sh.J. *Dasturlash fanlarida vaqt tanqisligi sharoitida loyihaviy ta'limni qo‘llash usullari.* // *Pedagogik mahorat*. — Buxoro, 2023. — № 4. — B. 115-120.
7. Holland, S., Griffiths, R., & Woodman, M. (1997). *Avoiding misconceptions for novices facing object-oriented design.* ACM SIGCSE Bulletin, 29(1), 218-221.
8. Ragonis, N., & Ben-Ari, M. (2005). *A theory of the concepts of object-oriented programming (Java).* Computer Science Education, 15(2), 119-135.