



**KOLLEKSIYALAR. COLLECTION VA ITERATOR
INTERFEYSLARI. RO‘YXATLAR. RO‘YXATLAR MASSIVI.
TO‘PLAMLAR. DARAXT VA XESH TO‘PLAMLARI. IZLASH VA
SARALASH.**

*Farg‘ona davlat universiteti Amaliy matematika va informatika kafedrası
katta o‘qituvchisi p.f.b.d (PhD)*

Umarov Bekzod Azizovich

baumarov@mail.ru

Mamatojiyeva Umidaxon Muxtorjon qizi

Farg‘ona davlat universiteti talabasi

mamatojiyeva05@gmail.com

Annotatsiya. Ushbu maqolada Java dasturlash tilidagi Collection va Iterator interfeyslari, ro‘yxatlar (List va massivlar), to‘plamlar (Set, TreeSet, HashSet), shuningdek, ularning mobil ilovalar ishlab chiqish jarayonida qo‘llanilishi chuqur tahlil qilinadi. Shuningdek, izlash va saralash algoritmlarining samaradorligi va kolleksiyalar bilan integratsiyasi ilmiy yondashuv asosida yoritiladi. Maqolada zamonaviy dasturlash yondashuvlariga asoslangan holda kolleksiyalarni to‘g‘ri tanlash va ulardan samarali foydalanish bo‘yicha tavsiyalar beriladi.

Kalit so‘zlar: Java, kolleksiyalar, iterator, ro‘yxat, massiv, to‘plam, daraxt to‘plami, xesh to‘plam, izlash, saralash, mobil ilovalar

Аннотация. В данной статье проводится глубокий анализ интерфейсов Collection и Iterator в языке программирования Java, а также списков (List и массивов), множеств (Set, TreeSet, HashSet) и их применения в разработке мобильных приложений. Рассматриваются алгоритмы поиска и сортировки, их эффективность и интеграция с коллекциями. Статья предоставляет рекомендации по эффективному выбору и использованию коллекций на основе современных подходов программирования.



Ключевые слова: Java, коллекции, итератор, список, массив, множество, дерево-множество, хеш-множество, поиск, сортировка, мобильные приложения

Annotation. *This article provides a comprehensive analysis of Collection and Iterator interfaces in Java programming, including lists (arrays and Lists), sets (Set, TreeSet, HashSet), and their application in mobile application development. It also explores the efficiency and integration of searching and sorting algorithms with collections. The article offers recommendations on the optimal use of collections based on modern programming practices.*

Keywords: Java, collections, iterator, list, array, set, tree set, hash set, searching, sorting, mobile applications

Zamonaviy dasturlash tizimlarining muvaffaqiyati ko'p jihatdan ma'lumotlar bilan qanday ishlanishiga bog'liq. Ma'lumotlar tuzilmalari va ularga ishlov berish algoritmlari har qanday dasturiy ta'minotning, xususan, mobil ilovalarning asosiy tayanchi hisoblanadi. Mobil qurilmalar cheklangan resurslarga ega bo'lganligi sababli (masalan, xotira, protsessor quvvati, batareya sarfi), bu turdagi ilovalar samaradorlik va optimallikni maksimal darajada talab qiladi. Shu sababli, dasturlashda kolleksiyalar (collections) va ularni boshqarish interfeyslari, xususan Collection va Iterator, juda katta ahamiyatga ega.

Java dasturlash tili mobil ilovalar ishlab chiqishda eng ommabop texnologiyalardan biri hisoblanadi. Aynan Android ilovalari yaratishda Java (va undan keyin Kotlin) asosiy dasturlash tillari sifatida tan olingan. Java tilida kolleksiyalarni boshqarish uchun mo'ljallangan java.util kutubxonasi mavjud bo'lib, unda ro'yxatlar (List), to'plamlar (Set, HashSet, TreeSet), xaritalar (Map) va ularning turli implementatsiyalari mavjud. Ushbu ma'lumotlar tuzilmalarini to'g'ri tanlash va ularni samarali boshqarish mobil ilovalarning tezkor ishlashida hal qiluvchi rol o'ynaydi.



Kolleksiyaalar tushunchasi ma'lumotlar to'plamlarini saqlash, ularga ishlov berish, izlash, qo'shish, o'chirish va tartiblash kabi amallarni amalga oshirishni osonlashtiradi. Java tilidagi Collection interfeysi barcha kolleksiya turlarining umumiy otasi sifatida qaraladi. Iterator interfeysi esa kolleksiyadagi elementlar ustida ketma-ket yurish (iteratsiya) qilish imkonini beradi. Bu yondashuvlar orqali foydalanuvchi mobil ilovadagi katta hajmdagi ma'lumotlar ustida tezkor va resursni tejovchi operatsiyalarni amalga oshirishi mumkin bo'ladi.

Mobil ilovalar uchun yaratilgan dasturiy arxitekturaning ko'rib chiqqanda, foydalanuvchi interfeysi va ma'lumotlar bazasi o'rtasidagi vositachilik vazifasini aynan kolleksiyalar bajaradi. Misol uchun, foydalanuvchi kontaktlar ro'yxatini ko'rayotgan bo'lsa, ushbu ro'yxat ichki jihatdan ArrayList yoki LinkedList ko'rinishida saqlanishi mumkin. Foydalanuvchi yangi kontakt qo'shganida, bu element kolleksiyaga dinamik tarzda qo'shiladi va interfeys real vaqt rejimida yangilanadi. Bu kabi funktsionallikni samarali ta'minlash uchun dasturchi ro'yxatlar bilan bir qatorda saralash (Collections.sort()), izlash (binarySearch), filtrlashtirish (stream().filter()) kabi funktsiyalarni to'g'ri qo'llay bilishi lozim.

To'plamlar — ya'ni Set implementatsiyalari — takrorlanuvchi bo'lmagan elementlar bilan ishlash uchun mo'ljallangan bo'lib, foydalanuvchi avtorizatsiyasi, ruxsat darajalari, noyob identifikatorlar va boshqa ko'plab funktsional sohalarda qo'llaniladi. HashSet orqali tez izlash va qo'shish amallari bajariladi, TreeSet esa elementlarni tartiblangan holatda saqlaydi. Bu to'plamlar mobil ilovalarda xavfsizlik va tartib intizomini saqlashda muhim rol o'ynaydi.

Bundan tashqari, Iterator interfeysining o'rni ham juda katta. Chunki mobil ilovalarda ko'plab ma'lumotlar ro'yxatlari bilan ishlashda oddiy for-each sikllar yetarli bo'lmaydi. Ba'zi hollarda foydalanuvchi kolleksiya ichidagi ma'lumotlar ustida shartli iteratsiya yoki kontekst asosida o'chirish kabi operatsiyalarni amalga oshirishi kerak bo'ladi. Bunday hollarda Iterator interfeysi eng mos vosita hisoblanadi.



Izlash va saralash algoritmlarining mobil ilovalarda tutgan oʻrni ham beqiyos. Mobil foydalanuvchi doim tezkor javobni kutadi. Masalan, kontaktlar boʻyicha qidiruvda binarySearch algoritmi yoki xeshlash asosida tezkor kirish (HashMap, HashSet) orqali foydalanuvchi talabiga darhol javob berish imkoniyati yaratiladi. Shu bilan birga, Collections.sort() yordamida roʻyxatlarni foydalanuvchiga qulay koʻrinishda tartiblab berish mumkin.

Maqolaning ushbu kirish qismida kolleksiyalar tushunchasining asosiy nazariy asoslari, ularning interfeyslari va mobil ilovalar kontekstida ahamiyati yoritildi. Keyingi boʻlimlarda esa mazkur maʼlumotlar tuzilmalarining texnik jihatlari, real hayotdagi qoʻllanilishi, algoritmik samaradorligi va ularni toʻgʻri tanlash boʻyicha ilmiy asoslangan tavsiyalar beriladi. Bundan koʻrinadiki, kolleksiyalar nafaqat nazariy tushuncha, balki amaliy dasturiy ishlab chiqishning ajralmas boʻlagidir. Ayniqsa, bugungi tezkor va resursga tejamkor mobil texnologiyalar davrida ular mutlaqo strategik ahamiyat kasb etadi.

Kolleksiyalar mobil ilovalar ishlab chiqishda muhim oʻrin egallaydi. Ular yordamida turli maʼlumotlarni samarali saqlash, tartiblash va ularga ishlov berish mumkin boʻladi. Mobil qurilmalarda resurslar cheklanganligi sababli, maʼlumotlar tuzilmasini toʻgʻri tanlash va optimal foydalanish ayniqsa dolzarbdir. Java dasturlash tilida kolleksiyalar toifasi bu masalani yechishda qulay vosita boʻlib xizmat qiladi. Ayniqsa, foydalanuvchi maʼlumotlarini saqlash, interfeysda dinamik roʻyxatlar bilan ishlash, tanlab olish va saralash jarayonlarida kolleksiyalarning oʻrni beqiyos.

Java tilida Collection interfeysi orqali barcha kolleksiya turidagi maʼlumotlar tuzilmalarini boshqarish mumkin. Bu interfeys yordamida maʼlumotlar toʻplamini yaratish, ularga element qoʻshish, olib tashlash, mavjudligini tekshirish va oʻlchamini aniqlash imkoniyatlari mavjud. Kolleksiyalar ustida iteratsiya qilishda esa Iterator interfeysi muhim ahamiyatga ega boʻlib, u orqali ketma-ket yurish, keyingi elementni olish yoki mavjud elementni olib tashlash funksiyalari bajariladi.



Aynan mobil ilovalar foydalanuvchi bilan interaktiv ishlovchi muhit bo'lganligi sababli, bu ikki interfeys – Collection va Iterator – juda ko'p uchraydi.

Kolleksiyalar turli sinflar orqali implementatsiya qilinadi. Ularning ichida List, Set va Map toifalari ayniqsa mashhurdir. List turiga mansub bo'lgan ArrayList va LinkedList sinflari foydalanuvchi interfeyslarida ro'yxatlar bilan ishlash uchun keng qo'llaniladi. ArrayList dinamik massiv bo'lib, unda indeks orqali elementlarga tezda murojaat qilish mumkin. Bu esa foydalanuvchi tajribasini tezkor va qulay qiladi. LinkedList esa bog'langan ro'yxat bo'lib, unda elementlar bir-biriga havola orqali ulanadi. Bu tuzilma, ayniqsa, elementlarni ro'yxat boshiga yoki o'rtasiga qo'shish yoki olib tashlash zarur bo'lganda foydalidir.

Massivlar ham ma'lumotlar saqlashning keng tarqalgan usullaridan biri bo'lib, ular bir xil turdagi ma'lumotlarni ketma-ket saqlaydi. Massivlar statik bo'lib, dastlabki o'lchami aniqlangach, uni keyinchalik o'zgartirib bo'lmaydi. Biroq ular resursni kam talab qilgani sababli mobil qurilmalarda juda foydali. Shunga qaramay, List kolleksiyalaridan foydalanish foydalanuvchiga dinamik imkoniyatlar yaratib beradi. Misol uchun, foydalanuvchi kontaktlarini saqlashda, yangi kontakt qo'shilganda ro'yxat avtomatik kengayadi va interfeys yangilanadi.

Set interfeysi orqali takrorlanuvchi bo'lmagan elementlar to'plamini boshqarish mumkin. HashSet, TreeSet va LinkedHashSet kabi sinflar ushbu interfeysni turli usullarda implementatsiya qiladi. HashSet elementlarni hash jadval yordamida tartibsiz saqlaydi va bu orqali tezkor izlash va qo'shish imkoniyatini beradi. Bu ayniqsa noyob identifikatorlar yoki foydalanuvchi IDlarini saqlashda foydalidir. TreeSet esa elementlarni tabiiy tartibda yoki komparator asosida saralangan holatda saqlaydi. Bu mahsulotlar yoki kontaktlar ro'yxatini nom, sana yoki narx bo'yicha saralashda foydali bo'ladi. LinkedHashSet esa elementlarni ular qo'shilgan tartibda saqlaydi, bu foydalanuvchi tomonidan oxirgi ko'rilgan elementlarni ketma-ket ko'rsatish imkonini beradi.



Ushbu kolleksiyalar orasidagi tanlov mobil ilovaning xususiyatiga bog'liq. Agar maqsad tezkor izlash va takrorlanuvchi bo'lmagan ma'lumotlarni saqlash bo'lsa, HashSet ideal variant hisoblanadi. Agar foydalanuvchiga ma'lumotlarni tartiblangan shaklda taqdim etish muhim bo'lsa, TreeSet tanlanadi. Oxirgi foydalanilgan tartibda ma'lumotlarni saqlash esa LinkedHashSet yordamida amalga oshiriladi.

Java'da kolleksiyalar ustida izlash va saralash ham samarali yo'lga qo'yilgan. Chiziqli izlash oddiy massiv yoki ro'yxatda har bir elementni tekshirib chiqish orqali ishlaydi, lekin bu nisbatan sekin usul bo'lib, $O(n)$ vaqt murakkabligiga ega. Saralangan ma'lumotlar ustida ishlaganda binarySearch metodidan foydalanish esa $O(\log n)$ murakkablikni ta'minlaydi. Java tilida mavjud bo'lgan Collections.binarySearch() metodidan foydalangan holda ro'yxatlardan tezda kerakli elementni topish mumkin.

Saralash algoritmlariga kelganda, Bubble sort eng oddiy usullardan bo'lib, lekin u katta hajmdagi ma'lumotlar uchun mos emas. Quick sort va Merge sort esa murakkabroq, lekin tezroq ishlovchi algoritmlardir. Java Collections.sort() metodi orqali ro'yxatlarni samarali tartiblash imkonini beradi. Ushbu metod Timsort algoritmiga asoslangan bo'lib, bu algoritm real hayotdagi ma'lumotlar tuzilmalariga moslashtirilgan.

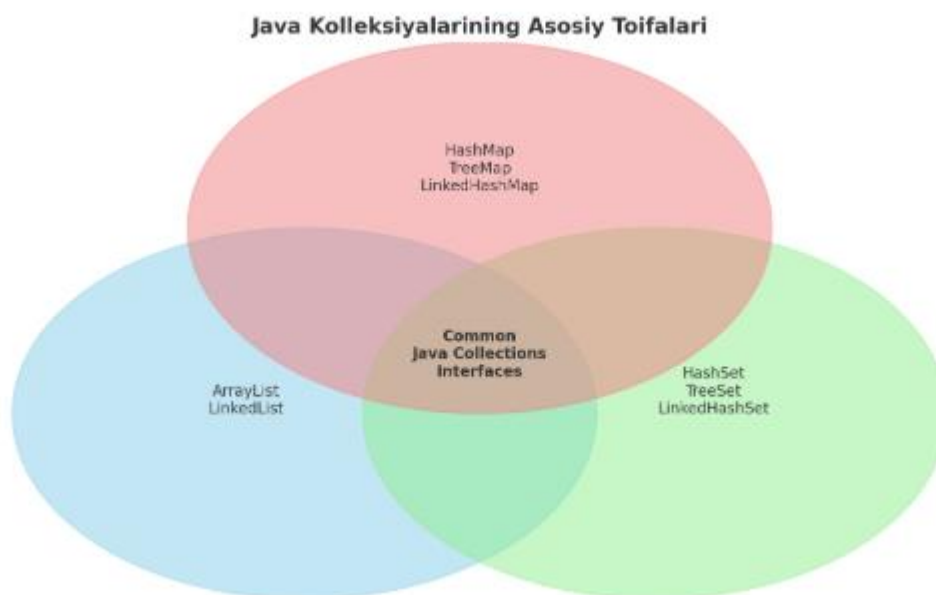
Mobil ilovalarda kolleksiyalardan foydalanishning ko'plab amaliy misollarini ko'rish mumkin. Masalan, kontaktlar ilovasida foydalanuvchining barcha kontaktlari ArrayList orqali saqlanadi. Yangi kontakt qo'shilganda yoki mavjudini tahrirlaganda bu ro'yxat avtomatik ravishda yangilanadi. Elektron do'kon ilovasida esa foydalanuvchi tanlagan mahsulotlar HashSet orqali saqlanishi mumkin, bu esa bir xil mahsulot bir necha bor kiritilishining oldini oladi. Mahsulotlar narxi bo'yicha saralanadigan sahifalarda esa TreeSet yordamida ma'lumotlar tartiblangan shaklda foydalanuvchiga ko'rsatiladi.



Shuningdek, so‘nggi ko‘rilgan mahsulotlar, eslatmalar yoki tarixlarni saqlash uchun LinkedHashSet qo‘llaniladi. Bu foydalanuvchiga aynan oxirgi ishlatilgan yoki ko‘rilgan elementlarni tartib bilan ko‘rsatish imkonini beradi. Misol uchun, yangiliklar ilovasida foydalanuvchi o‘qigan maqolalar ushbu turdagi kolleksiya orqali qayd etiladi.

Kolleksiyalarni tanlashda ularning tezligi, tartib saqlash xususiyati, qo‘shilish va o‘chirishdagi samaradorligi inobatga olinadi. Chunki mobil ilovalarda foydalanuvchi interfeysi juda muhim va har bir ish tezkor amalga oshirilishi kerak. Har bir kolleksiya turi o‘zining afzallik va kamchiliklariga ega, shuning uchun aniq vazifaga mos kolleksiyani tanlash eng muhim qarorlardan biridir.

Shuningdek, kolleksiyalarning xotira boshqaruvi bilan bog‘liq jihatlari ham ahamiyatlidir. Android platformasida ishlayotgan ilovalar uchun bu juda muhim, chunki xotira to‘lishi ilovaning to‘xtashiga yoki ishdan chiqishiga sabab bo‘lishi mumkin. Shu sababli, ArrayList kabi kolleksiyalar bilan ishlaganda, kerak bo‘lmagan elementlarni nullga o‘rnatish yoki kolleksiyani clear() metodi orqali tozalash orqali xotirani boshqarish tavsiya etiladi.



Shu tarzda, Java kolleksiyalari mobil ilovalarni samarali, interaktiv va foydalanuvchi uchun qulay shaklda ishlab chiqishga xizmat qiladi. Har bir



loyihaning ehtiyojidan kelib chiqib, to'g'ri kolleksiya turini tanlash orqali ilova funkcionalligini, tezligini va ishonchliligini oshirish mumkin. Ularning kombinatsiyalangan ishlatilishi esa yanada murakkab, ammo foydalanuvchiga qulay funksiyalarni yaratish imkonini beradi.

Kolleksiyalar dasturlashda ma'lumotlarni tartiblangan va samarali saqlash, boshqarish hamda qayta ishlash uchun muhim vosita hisoblanadi. Ular ma'lumotlar bazasini yaratishda, algoritmlar va ma'lumotlar tuzilmalari yordamida murakkab masalalarni yechishda asosiy rol o'ynaydi. Kolleksiyalar tushunchasi, asosan, turli tipdagi elementlarni bir joyda saqlash imkonini beradi va ularga qulay tarzda murojaat qilishni ta'minlaydi.

Collection interfeysi bu konseptning asosiy elementi bo'lib, unda umumiy maqsadlar uchun mo'ljallangan metodlar to'plami joylashgan. Collection interfeysi yordamida ro'yxatlarni, to'plamlarni va boshqa ko'plab ma'lumotlar tuzilmalarini yaratish va ularga amallar bajarish mumkin. Iterator interfeysi esa kolleksiyadagi elementlarni ketma-ket ko'rib chiqish uchun qulay vositadir. Iterator yordamida elementlarni bitta-bitta aylanib chiqish, ularni o'qish yoki o'zgartirish osonlashadi.

Ro'yxatlar (List) – elementlar ketma-ketligini saqlaydigan kolleksiya turi bo'lib, unda takroriy elementlar mavjud bo'lishi mumkin. Ro'yxatning eng keng tarqalgan realizatsiyasi – massiv (array) asosidagi ro'yxat bo'lib, u indeks orqali elementlarga tezkor murojaat qilish imkonini beradi. Massiv ro'yxatlarining afzalligi shundaki, ular xotirada uzluksiz joy egallaydi va elementlarga kirish tez amalga oshadi. Biroq, massivlar o'lchami dastlab belgilangani sababli, dinamik o'sish yoki qisqarish qiyinchilik tug'diradi. Shuning uchun ko'plab dasturlash tillari va kutubxonalarida dinamik ro'yxatlar yaratish imkoniyati ham mavjud.

To'plamlar (Set) esa noyob elementlar to'plamini ifodalaydi. Ular takroriy elementlarni qabul qilmaydi va ko'pincha ma'lumotlarni tezkor tekshirish, qo'shish yoki o'chirish vazifalarida ishlatiladi. To'plamlar tarkibida murakkab daraxt yoki xesh jadval (hash table) asosida qurilgan tuzilmalar bo'lishi mumkin.



Daraxt to'plamlari (masalan, qizil-qora daraxtlar yoki AVL daraxtlari) ma'lumotlarni tartiblangan holda saqlaydi, shu bilan birga, izlash, qo'shish va o'chirish operatsiyalarini samarali bajaradi. Daraxt to'plamlari ma'lumotlarning tartiblangan ko'rinishda bo'lishini ta'minlagani sababli, ular sortlash va tartiblash talab qilinadigan vazifalarda qulay hisoblanadi.

Xesh to'plamlari (HashSet) esa xesh funksiyasi yordamida elementlarni tartibsiz, lekin juda tezkor qidirish imkonini beradigan to'plamlar hisoblanadi. Xesh to'plamlari ayniqsa katta hajmdagi ma'lumotlarni samarali boshqarishda qo'llaniladi, chunki ular o'rtacha holatda elementlarni qo'shish, o'chirish va qidirishni $O(1)$ vaqt ichida bajaradi.

Ma'lumotlarni izlash va saralash (sorting) kolleksiyalar bilan ishlashda juda muhim jarayonlardandir. Izlash algoritmlari ma'lumotlar to'plamidan kerakli elementni topishga xizmat qiladi. Kolleksiyalar ichida oddiy chiziqli izlashdan tortib, samaraliroq ikkiyuzli izlash (binary search) kabi algoritmlargacha foydalaniladi. Saralash esa ma'lumotlarni tartiblab chiqarish uchun kerak bo'lib, bu ham ma'lumotni tezroq qidirish va tartiblangan ko'rinishda ishlash imkonini beradi. Saralash algoritmlariga tez saralash (quick sort), birlashtirish orqali saralash (merge sort), tanlab olish saralash (selection sort) kabi ko'plab algoritmlar kiradi.

Umuman olganda, kolleksiyalar va ular bilan bog'liq interfeyslar dasturlashning asosiy tamoyillaridan biridir. Ular yordamida ma'lumotlar samarali saqlanadi, ularni boshqarish va qayta ishlash imkoniyati kengayadi. Collection va Iterator interfeyslari esa dasturiy ta'minotda o'zaro moslashuvchanlik va kengaytiriluvchanlikni ta'minlaydi. Ro'yxatlar, to'plamlar, daraxt va xesh to'plamlari kabi ma'lumotlar tuzilmalarining o'ziga xos afzalliklari va qo'llanish sohalari mavjud. To'g'ri kolleksiyalarni tanlash va samarali izlash-saralash usullarini qo'llash dastur samaradorligini sezilarli darajada oshiradi va murakkab vazifalarni yechishda yordam beradi.



Shuning uchun har bir dasturchi uchun bu mavzuni chuqur tushunish, turli kolleksiyalarni to'g'ri tanlash va ularning ishlash printsiplarini yaxshi bilish juda muhimdir. Bu bilim dasturiy ta'minot sifatini oshirish, optimallashtirish va murakkab tizimlarni samarali boshqarishda asosiy omil bo'ladi.

Foydalanilgan adabiyotlar

1. Umarov B. RAQAMLI TEXNOLOGIYALAR VOSITASIDA PEDAGOGLARNING PROFESSIONAL KOMPETENTLIGINI RIVOJLANTIRISH MAZMUNI //Евразийский журнал математической теории и компьютерных наук. – 2023. – Т. 3. – №. 5. – С. 87-93.
2. Azizovich U. B. PRINCIPLES OF FORMING TEACHER COMPETENCE THROUGH INNOVATIVE TECHNOLOGIES. Finland International Scientific Journal of Education //Social Science & Humanities. – 2023. – Т. 11. – №. 5. – С. 823-828.
3. Azizovich U. B. PEDAGOGICAL-PSYCHOLOGICAL PRINCIPLES OF THE FORMATION OF PROFESSIONAL COMPETENCE //Confrencea. – 2023. – Т. 6. – №. 6. – С. 204-212.
4. Azizovich U. B., Zarifjon o'g'li X. N. BULUT TEXNOLOGIYALARINING AFZALLIKLARI VA KAMCHILIKLARI //TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI. – 2024. – Т. 1. – №. 1. – С. 46-54.
5. Azizovich U. B., Rustamjon o'g'li R. Z. MA'LUMOTLARNI SHIRFLASH TENALOGIYALARI VA XAVFSIZLIK STANDARTLARI //TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI. – 2024. – Т. 1. – №. 1. – С. 105-108.
6. Azizovich U. B. et al. OLAP TIZIMLARINING ASOSIY PRINSIPLARI //TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI. – 2024. – Т. 1. – №. 1. – С. 81-86.



7. Azizovich U. B. THE DEVELOPMENT OF PROFESSIONAL COMPETENCY OF TEACHERS IN EDUCATIONAL TECHNOLOGY BASED ON DIGITAL TECHNOLOGIES //Eurasian Journal of Mathematical Theory and Computer Sciences. – 2024. – Т. 4. – №. 7. – С. 11-14.
8. Azizovich U. B. et al. MASHINALI O ‘QITISHDA REGRESSIYA ENG KICHIK KVADRATLAR USULINI QO ‘LLASH //INNOVATION IN THE MODERN EDUCATION SYSTEM. – 2024. – Т. 5. – №. 46. – С. 266-270.