JOURNAL OF INTERNATIONAL SCIENTIFIC RESEARCH
Volume 2, Issue 8, April, 2025 Online ISSN: 3030-3508
https://spaceknowladge.com

MASSIVLAR USTIDA AMALLAR BAJARISHDA ARRAY SINFIDAN FOYDALANISH

Shermatova Xilola Mirzayevna [1]
[1] *FarDU Axborot texnologiyalari kafedrasi dotsenti*
shermatovahilola1978@gmail.com
G'ofurova Guli Abduvohidjon qizi [1]
[1] *Farg'ona davlat universiteti Axborot tizimlari va texnologiyalari yo'nalishi 1-kurs talabasi*
guligofurova36@gmail.com

MAQOLA MALUMOTI

MAQOLA TARIXI:
Received:07.04.2025
Revised: 08.04.2025
Accepted:09.04.2025

KALIT SO'ZLAR:
Array sinfi, massiv, tartiblash, statik massiv, dasturlash, samaradorchilik, element qo`shish.

ANNOTATSIYA:

Ushbu maqolada Array sinfining massivlar ustida amallar bajarishda tutgan o'rni va ahamiyati tahlil qilinadi. Array sinfi yordamida massiv yaratish, unga elementlar qo'shish, o'chirish, tartiblash, va boshqa amallarni bajarish usullari ko'rib chiqiladi. Maqolada statik va dinamik massivlar, shuningdek, ularning ishlash printsiplari haqida tushuncha beriladi. Array sinfining imkoniyatlarini to'liq tushunish dasturlashda samaradorlikni oshirishga yordam beradi.

KIRISH. Hozirgi kunda dasturlash jarayonida ma'lumotlarni samarali saqlash va qayta ishlash katta ahamiyatga ega. Dasturlash tillari orasida C++ tili o'zining yuqori samaradorligi va qulay imkoniyatlari bilan ajralib turadi. Ma'lumotlar ustida samarali amallar bajarish uchun massivlar muhim vositalardan biri hisoblanadi. Massivlar yordamida ma'lumotlarni tartib bilan saqlash, ularga tezkor murojaat qilish va turli hisob-kitoblarni bajarish mumkin.

C# dasturlash tilida massivlar bilan ishlashni soddalashtirish va yanada samarali bajarish uchun Array sinfidan foydalanish mumkin. Array sinfi oddiy massivlarga nisbatan qulayroq ishlash imkonini beradi, chunki u turli funksiyalar orqali massivlarni boshqarish, qayta ishlash va ularga turli amallarni bajarish imkonini yaratadi.

Mazkur maqolada C# tilida Array sinfi yordamida massivlar bilan ishlash jarayonlari, massivlarni yaratish, ularga qiymat berish, o‘qish va yozish algoritmlari, massivlar ustida amallar bajarish usullari, shuningdek, Array sinfining qulayliklari va afzalliklari yoritiladi.

Massivlar ustida amallar haqida tushuncha

C# dasturlash tilida kompyuter xotirasiga bir o’zgaruvchi yordamida bir nechta qiymatlarda foydalanishga to’g’ri keladi. Bir o’zgaruvchi bilan bir nechta qiymat ustida amallar bajarish uchun berilgan ma’lumotlar bir turga mansub bo’lishi kerak. C# dasturlash tilida bir o’zgaruvchi yordamida bir nechta qiymatlardan foydalanish uchun massiv degan turdan foydalaniladi. Dasturlash tillarida ro’yxat yoki jadval ko’rinishidagi ma’lumotlarni massiv deb atashadi. Massiv so’zining ma’nosi o’lcham, o’lchov demakdir. Massivning barcha elementlari bitta turga mansub bo’lib, ular bitta nom bilan nomlanadi va bir-birlaridan nomerlari (indekslari) bilan farq qiladi.

Endi dasturdagi ma’lumot strukturalari bilan tanishishni boshlaymiz. Dasturda ikki asosiy tur ma’lumot strukturalari mavjuddir. Birinchisi statik, ikkinchisi dinamikdir. Statik deganimizda xotirada egallagan joyi o’zgarmas, dastur boshida beriladigan strukturalarni nazarda tutamiz. Statik massivlar elementlar soni oldindan ma’lum bo’lgan va initsializatsiyalangan (qiymat belgilangan) massivlar hisoblanadi.

Dinamik ma’lumot tiplari dastur davomida o’z hajmini, egallagan xotirasini o’zgartirishi mumkin. Dinamik massivlar esa elementlari soni oldindan ma’lum bo’lishi va uni initsializatsiyalash (qiymat belgilash) shart emas. Statik massivlarning kamchiliki shundaki, agar ularning o’lchamini oldindan juda katta olinsa-yu, uning ko’p qismi keraksiz qolib ketsa, u holda xotira behuda sarflanishiga olib keladi. Shu muammoni hal qilish maqsadida massivlar C# tilida asosan dinamik tarzda e’lon qilinadi. Massivlar dasturlashda eng ko’p qo’laniladigan ma’lumot tiplaridir. Massivlar hotirada ketma-ket joylashgan, bir tipdagi o’zgaruvchilar guruhidir. Alohida bir o’zgaruvchini ko’rsatish uchun massiv nomi va kerakli o’zgaruvchi indeksini yoziladi.

Bir turga mansub bo`lgan yagona nom bilan saqlanuvchi tartiblangan ma`lumotlar majmuasi **massiv** deyiladi.

Massivlar yagona o’zgaruvchi bilan kompyuter xotirasiga saqlanadi, uning elementlari ma’lum bir indekslar bilan tartiblab joylashtiriladi. Massivlar yagona nom bilan bir nechta

qiymatni o’zida mujassamlashtiradi, bularga matematikadagi vektorlarni misol keltirish mumkin. Vektor ham yagona nom bilan saqlanib uning tarkibida bir nechta qiymatni o’zida mujassamlashadi. Vektorning ham elementlari bir turga mansub va tartiblangan bo’ladi.

Bir o’lchovli massivlar

Odatda massivlar zarurat, katta hajmdagi tartiblangan, lekin chekli elementlarga oid masalalarni hal etishda yuzaga keladi. Dastur ishlatilishi davomida massivlar aniq nomga ega bo’lishi va uning elementlari ma’lum bir turda bo’lishi kerak. Bir o’lchovli massivlar kompyuter xotirasiga quyidagi shaklda saqlanadi

Massiv tarkibida elementlar mavjud bo’ladi. Massivning eng ko’pi bilan ketishi mumkin bo’lgan elementlar soni uning o’lchamini bildiradi. Massivning elementi turgan o’rni uning indeksi deyiladi. Massivning elementiga uning indeksi orqali murojaat qilinadi. Massivning indeksi sifatida butun sonlar xizmat qiladi. Har bir massiv o’zining individual nomiga ega bo’lishi kerak, ya’ni bir xil nomdagi massivlar bo’lmaydi. Ularning nomi oldin e’lon qilingan oddiy o’zgaruvchi nomi bilan ustma-ust tushmasligi kerak.

Statik massivlarni e’lon qilishning umumiy ko’rinishi quyidagicha:

<tip> []<massiv_nomi>={boshlang’ich qiymatlar}

Bunda {boshlang’ich qiymatlar} albatta bo’lishi kerak.

Misollar:

```
int []A={1,4,3,1};
string[] B = { “olma”, “gilos”, “anor”};
double[] C = { 0.005, 1.234, 12.5, 13.5, 10.6 };
```

Yuqoridagi massivlarda massivning o’lchami uning initsializatsiya qismida qatnashgan elementlar soni bilan aniqlanadi. C# tilida xuddi C# da bo’lgani kabi element indeksi 0 dan boshlanadi. A[0] indeksli element 1 ga teng, B[1] indeksli element esa “gilos” ga teng va hakozo.

Indekslar massiv elementlariga murojat qilish uchun ishlatiladi. Indeks massivdagi element sonini bildiradi .Massivdagi to’rtinchi elementga murojat qilish uchun biz 3 indeksidan foydalanishimiz kerak. Misol uchun: son[3]. Massiv elementlarining qiymatlarini olish va o’rnatish uchun indekslardan foydalanamiz.

```
int[] sonlar=new int [4];
sonlar [0]=1;
sonlar [1]=2;
sonlar [2]=3;
```

sonlar [3]=5;
Console.ReadLine(sonlar [3]); //5

Bizda faqat 4 ta element uchun belgilangan massiv mavjud bo'lgani uchun , masalan oltinchi elementni qo'llay olmaymiz sonlar [5]=5;. Agar biz buni qilishga harakat qilsak biz IndexOutOfRangeException-ni oladi.

Array - bu bir xil toifali, chekli qiymatlarning tartiblangan to'plamidir(Array o'zbek tilida massiv deb yuritiladi). Massivlarga misol qilib matematika kursidan ma'lum bo'lgan vektorlar, matritsalarni ko'rsatish mumkin. Massivlarga hayotdan misol qilib biror tuman yoki shaharlarning qaysidir kichik hududlarida bir xil turda qurilgan, aholi yashash punktlarini olishimiz mumkin. Ba'zi yangiliklarda massiv deb atalishini ko'p eshtganmiz. Bu misolimizda ya'ni massivda joylashgan ko'p qavatli uylarni massivlar deb olsak, har bir xonadan o'z raqamiga ya'ni indexiga ega. Har bir raqamga ega xonadonda esa xonadon egalari istiomat qilishiadi ularni massiv elementlari deb qarashimiz mumkin.

XULOSA

Ushbu maqolada C# dasturlash tilida massivlar ustida amallar bajarishda Array sinfidan foydalanishning asosiy jihatlari yoritildi. Massivlar bilan ishlash dasturlash jarayonining ajralmas qismi hisoblanadi, chunki ular katta hajmdagi ma'lumotlarni samarali saqlash, ularga tezkor murojaat qilish va turli amallarni bajarish imkonini beradi. C# dasturlash tilida massivlar bilan ishlash uchun array sinfi muhim qulayliklarni taqdim etadi.

Maqolada Array sinfining asosiy xususiyatlari, uning oddiy massivlardan farqi hamda foydalanish afzalliklari ko‘rib chiqildi. Array odatiy massivlardan farqli ravishda, C# STL (Standard Template Library) tarkibiga kiradi va xavfsizlik, samaradorlik hamda qulay interfeys kabi afzalliklarga ega. Array sinfi statik uzunlikka ega bo‘lib, kompilatsiya vaqtida uning o‘lchami aniq bo‘lishi talab etiladi.

Array sinfidan foydalanish massiv bilan ishlash jarayonlarini sodda va tushunarli qiladi. Unda massiv elementlariga indeks orqali murojaat qilish, size() metodi yordamida o‘lchamni aniqlash, at() metodi orqali indeks chegaralarini tekshirish imkoniyatlari mavjud. Shuningdek, fill(), swap() kabi metodlar yordamida ma'lumotlar bilan samarali ishlash ta'minlanadi.

Shuningdek, maqolada vector bilan array sinflarining taqqoslanishi ham yoritilib, ularning ishlash prinsiplari va qo‘llanilish holatlari muhokama qilindi. vector dinamik o‘lchamga ega bo‘lsa, array statik uzunlikka ega bo‘lib, resurslardan yanada samarali foydalanishga imkon beradi. C# tilida massivlar bilan ishlashni mukammal o‘zlashtirish dasturlar samaradorligini oshirishga yordam beradi. Dasturchilar array sinfining

imkoniyatlarini chuqur o‘rganish orqali kodni yanada tushunarli, xavfsiz va samarali qilishi mumkin.

Yakunda aytish mumkinki, array sinfi massivlarni ishlatish jarayonini soddalashtirib, kodning xavfsizligi va ishlash tezligini oshiradi. Dasturchilar massivlarni samarali tashkil qilish orqali dasturlarning ishonchliligini ta’minlashi va resurslardan optimal foydalanishi mumkin.

Foydalanilgan adabiyotlar:

1. Stroustrup B. The C++ Programming Language – Addison-Wesley, 2013.
2. Lippman S.B., Lajoie J., Moo B.E. C++ Primer – Addison-Wesley, 2012.
3. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 – O’Reilly Media, 2014.
4. Josuttis N.M. The C++ Standard Library: A Tutorial and Reference – Addison-Wesley, 2012.
5. Kerrisk M. The Linux Programming Interface – No Starch Press, 2010.
6. Overland B. C++ Without Fear: A Beginner’s Guide That Makes You Feel Smart – Addison-Wesley, 2015.
7. Koenig A., Moo B.E. Accelerated C++: Practical Programming by Example – Addison-Wesley, 2000.
8. ISO/IEC 14882:2017. Programming Languages — C++ – International Organization for Standardization, 2017.
9. Eckel B. Thinking in C++ – Prentice Hall, 2000.
10. Balagurusamy E. Object-Oriented Programming with C++ – McGraw-Hill Education, 2013.
11. Sutter H. Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions – Addison-Wesley, 2000.resurslardan optimal foydalanishi mumkin.