

C# DASTURLASH TILIDA TO'PLAMLAR BILAN ISHLASH

Farmonov Sherzodbek Raxmonjonovich

*Farg'ona davlat universiteti amaliy matematika va informatika
kafedrası katta o'qituvchisi
farmonovsh@gmail.com*

Rustamova Humoraxon Sultonbek qizi

*Farg'ona davlat universiteti 2-kurs talabasi
rustamovahumoraxon@gmail.com*

Annotatsiya: Bu maqolada C# dasturlash tilida massivlar bilan ishlash va ularni algoritmik muammolarni yechishda qanday qilib ishlatish ko'rsatiladi. C# tilida massivlar, ko'p elementli ob'ektlar sifatida foydalaniladi. Ular yaratish usullari va ularga murojaat qilish ko'rsatiladi. Maqolada massiv yaratish va boshlang'ich qiymatlar bilan ishlov berishda ishlatiladigan usullar bayon etiladi. Algoritmik muammolar, C# tilida massivlar orqali oddiy algoritmalar orqali yechilishi ko'rsatiladi. Maqolada saralash algoritmlari, masalan, Bubble Sort va Selection Sort kabi, massivlarni tartiblashda ishlatiladigan usullar bayon etiladi. Max elementni topish, massivning o'rta arifmetigi va eng kichik elementni topish kabi algoritmik muammolar C# tilida massivlar orqali oddiy usullar bilan yechilishini ta'riflaydi. Maqola, C# dasturlash tilida massivlar bilan ishlash, ularga murojaat qilish va ular orqali algoritmik muammolar yechishning oddiy usullarini ta'riflaydi.

Kalit so'zlar: Massivlar, yaratish, murojaat qilish, indeks, algoritmik muammolar, saralash, Bubble Sort, Selection Sort, o'rta arifmetigi, max element, min element, elementlar.

C# tilida massivlar bilan ishlash va ularni algoritmik muammolarni yechishda qanday qilib ishlatishni o'rganish juda muhimdir. Massivlar, ko'p elementli ob'ektlar sifatida dasturlashda keng qo'llaniladi. Ular yaratish, ularga murojaat qilish va algoritmik muammolarni oddiy usullar bilan yechishga imkoniyat beradi.

Massivlar va ularning yaratilishi. C# tilida massivlar yaratish usullari, ko'p elementli ob'ektlarga murojaat qilish va ularga qiymat berishning oddiy usullari bayon etiladi. C# dasturlash tilida massivlar ko'p elementli ob'ektlar sifatida foydalaniladi. Ularni yaratish uchun, ko'p variantlar mavjud:

- Bo'sh massiv yaratish: Massivning o'lchamini belgilab, bo'sh massiv yaratish mumkin. `int[] raqamlar = new int[5];` deyilgan holda, raqamlar nomli bo'sh massiv yaratiladi.

- Boshlang'ich qiymatlar bilan massiv yaratish: Boshlang'ich qiymatlar ko'rsatilgan bo'lishi mumkin. `int[] raqamlar = new int[] { 1, 2, 3, 4, 5 };` deyilgan holda, raqamlar nomli massiv yaratiladi va ichiga boshlang'ich qiymatlar joylashadi.
- Odatiy massiv yaratish: Odatiy sintaksis orqali ham massiv yaratish mumkin. `int[] raqamlar = { 1, 2, 3, 4, 5 };` deyilgan holda ham raqamlar nomli massiv yaratiladi. Massivlar indekslar yordamida murojaat qilinadi. Indeksler noldan boshlanadi, ya'ni birinchi element indeksi 0 ga tengdir. Misol uchun, `raqamlar[0]` ifodasi massivning birinchi elementiga murojaat qilish uchun ishlatiladi.

C# dasturlash tilida massivlar, ko'p elementli ob'ektlar sifatida foydalaniladi. Ular yaratilishi, boshlang'ich qiymatlar bilan yaratilishi va ularga murojaat qilishning oddiy usullari mavjud.

Bu misollar C# tilida massivlar bilan ishlovchi algoritmlardan faqat bir qismini ko'rsatadi. Bu algoritmlar massivlar bilan ishlashda keng qo'llaniladi va ularga murojaat qilish oddiy.

- **Massiv yaratish:**

Massivning o'lchamini belgilab bo'sh massiv yaratish:

```
int[] raqamlar = new int[5]; // 5 elementli bo'sh massiv
```

Boshlang'ich qiymatlar bilan massiv yaratish:

```
int[] raqamlar = new int[] { 1, 2, 3, 4, 5 }; // Boshlang'ich qiymatlar bilan massiv
```

- **Murojaat:**

Massiv elementlariga murojaat indekslar orqali amalga oshiriladi:

```
int birinchiRaqam = raqamlar[0]; // Massivning birinchi elementiga murojaat
```

```
int ikkinchiRaqam = raqamlar[1]; // Massivning ikkinchi elementiga murojaat
```

Algoritmik Muammolar: Saralash va Qanday Yechish

- **Bubble Sort Algoritmi:**

Bubble Sort, massivni o'sish tartibida saralash uchun odatiy algoritmi hisoblanadi. Elementlar orasidagi tartibni solishtirib boradi va eng kichik qiymatlarni massivning boshiga olib boradi. Bubble Sort, massivni tartiblash uchun odatiy usuldir:

```
void BubbleSort(int[] array)
{
    int n = array.Length;
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (array[j] > array[j + 1])
            {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
        }
    }
}
```

```
        array[j + 1] = temp;
    }
}
}
```

- **Max/Min elementni topish:**

Eng katta element:

```
int FindMax(int[] array)
{
    int max = array[0];
    for (int i = 1; i < array.Length; i++)
    {
        if (array[i] > max)
        {
            max = array[i];
        }
    }
    return max;
}
```

Eng kichik element:

```
int FindMin(int[] array)
{
    int min = array[0];
    for (int i = 1; i < array.Length; i++)
    {
        if (array[i] < min)
        {
            min = array[i];
        }
    }
    return min;
}
```

- **Massivning O'rta Arifmetigini Topish:**

```
double FindMean(int[] array)
{
    int sum = 0;
    foreach (int num in array)
    {
        sum += num;
    }
}
```

```
}  
    return (double)sum / array.Length;  
}
```

Bu maqola, C# tilida massivlar va ular bilan ishlash, shuningdek algoritmik muammolar yechishning oddiy usullarini ko'rsatadi. Algoritmilar, massivlardagi ma'lumotlarni tuzatish va ulardan foydalanishda ko'p qo'llaniladi.

Xulosa. Massivlar dasturlashda muhim qat'iyatga ega, ular bilan ishlash va ularda ishlatiladigan algoritmilar dasturchilar uchun cheksiz imkoniyatlar yaratadi. Massivlar ko'p elementli ob'ektlar sifatida foydalaniladi. Bo'sh massiv yaratish, boshlang'ich qiymatlar bilan massiv yaratish va indekslar orqali massivlarga murojaat odatiy usullardan biri hisoblanadi. Algoritmilar, saralash usullari (masalan, Bubble Sort, Selection Sort) orqali massivlar saralashda ishlatiladi. Eng katta/eng kichik elementni topish va massivning o'rta arifmetigini topish ham oddiy algoritmik muammolardan bir qancha misollar. Maqolada misollar va C# kodlar yordamida massivlar va ularda ishlatiladigan algoritmilar o'rganishga imkoniyat beradi. Bu kodlar dasturchilarga tushunarli, amaliy misollar bilan ta'minlangan.

Foydalanilgan adabiyotlar ro'yhati.

1. John Paul Mueller. C# 10.0 All-in-One For Dummies. Published by: John Wiley & Sons, Inc., 2022. — 830 c.
2. A.Troelsen, R.Japikse. Pro C# 8 with .NET Core 3. Foundational Principles and Practices in Programming. Ninth Edition. Apress, 2020. – 1223 c.
3. A.Васильев. Программирование на C# для начинающих. Основные свидания.: – М.: "Эксмо", 2018. – 592 c.
4. Фленов М. Е. Библия C#. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2016. — 544 c.
5. Фармонов, Ш., & Камбарова, Д. (2022). КАК ПОМОЧЬ УЧЕНИКАМ РАЗВИТЬ ИНТЕРЕС К УЧЕБЕ. Центральноеазиатский журнал образования и инноваций, 1(2), 118-120.
6. Farmonov, S., & Rahmatjonov, M. (2023). FLUTTER YORDAMIDA PLATFORMALARARO KUTUBXONANI ISHLAB CHIQISH TECHNOLOGIYASI. Центральноеазиатский журнал образования и инноваций, 2(6 Part 2), 124-127.
7. Raхmonjonovich, F. S. (2023). USE OF BLENDED LEARNING TECHNOLOGY IN ORGANIZING INDEPENDENT EDUCATION OF STUDENTS. MATERIALLAR TO'PLAMI, 352.
8. Farmonov, S., & Karimova, M. (2023). MODERN METHODS TO DEVELOP MATHEMATICAL THINKING IN SCHOOLCHILDREN. Бюллетень педагогов нового Узбекистана, 1(6 Part 2), 28-38.

9. Tojiyev, T., Boynazarov, A., & Farmonov, S. (2022). PHARMACOKINETICS IS A DESCRIPTION OF DRUGS AND THEIR BEHAVIOR IN THE HUMAN BODY BY BUILDING A MATHEMATICAL MODEL. Евразийский журнал медицинских и естественных наук, 2(13), 146-149.
10. Фармонов, Ш., & Хайдарова, С. (2022). Обобщенный метод Бубнова-Галеркина для уравнений с дробно-дифференциальным оператором. Norwegian Journal of Development of the International Science, (99), 10-15.