



ANIQ FANLAR

C# DASTURLASH TILI MISOLIDA DASTURLASH ASOSLARI

Sherali O‘RINOV¹

¹ISFT instituti IKI kafedrası professori

Nurali MAVLONOV²

²ISFT instituti IKI kafedrası mudiri

Abdukarim KARIMOV³

³ISFT instituti IKI kafedrası o‘qituvchisi

KALIT SO‘ZLAR

Dasturlash, C#, o‘zgaruvchilar, konstantalar, operatorlar, massivlar, metodlar.

КЛЮЧЕВЫЕ СЛОВА

Программирование, C#, переменные, константы, операторы, массивы, методы.

KEY WORDS

Programming, C#, variables, constants, operators, arrays, methods.

ANNOTATSIYA

Ushbu maqolada C# dasturlash tili misolida dasturlash asoslari yoritilgan. Tadqiqotning maqsadi – C# tilida dasturlashning fundamental tamoyillarini ko‘rsatish va dasturchilarga bilimlarini oshirishda yordam berish. Maqola alifbodan boshlab, o‘zgaruvchilar, konstantalar, operatorlar va massivlar kabi asosiy mavzularni qamrab oladi. Natijalar C# tilining algoritmlar tuzishda qulayligini va OYD tamoyillari bilan zamonaviy dasturlar yaratishda keng qo‘llanilishini ko‘rsatadi. Maqolada dasturlash asoslari C# misolida ilmiy-amaliy yondashuv bilan ko‘rsatilgan.

АННОТАЦИЯ

В статье рассматриваются основы программирования на примере языка C#. Цель – показать, как фундаментальные принципы программирования работают на C#, и помочь программистам укрепить знания. Статья охватывает такие темы, как переменные, константы, операторы и массивы. Результаты показывают, что C# удобен для создания алгоритмов и широко используется с ООП при разработке современных программ. Статья демонстрирует научно-практические аспекты программирования на C#.

ABSTRACT

This article examines the basics of programming using the C# language. The study aims to demonstrate how core programming principles operate in C# and assist programmers in enhancing their knowledge. It covers topics such as variables, constants, operators, and arrays. The findings show that C# is effective for building algorithms and is widely used with OOP principles in modern software development. The article presents practical and scientific aspects of programming fundamentals using C#.

Kirish. Dasturlash bugungi kunning eng muhim texnologik yo‘nalishlaridan biri hisoblanadi. Zamonaviy dunyoda turli tillarda dasturlashni o‘rganish va qo‘llash dasturchilar uchun zarurat bo‘lib qolmoqda. C# dasturlash tili o‘zining yuqori samaradorligi va aniqligi

bilan boshqa tillardan ajralib turadi. Ushbu maqolada biz dasturlash asoslarini C# misolida yoritamiz, bunda alifbo, o‘zgaruvchilar, konstantalar, operatorlar va massivlar haqida batafsil ma‘lumot beriladi. Asosiy maqsad – dasturlashda yangi bo‘lganlar uchun C# tilidagi

asosiy tushunchalarni amaliy misollar bilan ko‘rsatishdir.

Tadqiqot metodologiyasi Ushbu maqolada dasturlash asoslari C# dasturlash tili misolida o‘rganiladi. Maqolani yozish jarayonida analitik yondashuv va eksperimental metodlardan foydalanildi. C# dasturlash tilining asosiy komponentlari haqida ma’lumotlar yig‘ildi va ular aniq misollar orqali sinovdan o‘tkazildi. Har bir misolni yozishda ilmiy-texnik adabiyotlar tahlil qilindi va dasturlash tamoyillari to‘g‘risida chuqur tadqiqotlar olib borildi. Maqolada nazariy bilimlarni amaliy misollar bilan bog‘lashga alohida e’tibor qaratildi, har bir kod misoli natijalari aniq chiqarildi va o‘rganildi.

Natijalar. C# dasturlash tilida dasturlashning asosiy tamoyillari va ularning amaliy qo‘llanilishi mazkur tadqiqot davomida batafsil o‘rganildi. Ushbu qismda o‘zgaruvchilar, konstantalar, operatorlar, massivlar va metodlar bo‘yicha aniq amaliyotlar va ularning natijalari ko‘rib chiqildi. Har bir tushuncha asosida keltirilgan kod misollari yordamida ularning ishlash jarayonlari tahlil qilindi. Bu natijalar C# dasturlash tilini o‘rganishda yangi boshlovchilar uchun amaliy ko‘rsatma sifatida foydali bo‘lishi mumkin [1; 123].

O‘zgaruvchilar va ularning dasturdagi roli

C# tilida o‘zgaruvchilarni aniqlash dasturiy jarayonlarning asosi hisoblanadi. O‘zgaruvchilar orqali ma’lumotlarni saqlash va ularga murojaat qilish amalga oshiriladi. O‘rganilgan natijalarga ko‘ra, C# tilida o‘zgaruvchilar bir nechta turga ega bo‘lib, ularning turiga qarab dastur samaradorligi o‘zgarishi mumkin. C# da o‘zgaruvchilar int, double, string, char kabi turli ma’lumot turlari bilan ishlaydi [2; 87].

O‘zgaruvchilarni qo‘llash misoli:

```
using System;
class Program
{
    static void Main ()
    {
        int yosh = 25;
        double maosh = 2500.75;
        string ism = "Dilshod";
        Console.WriteLine ("Ism: " + ism);
        Console.WriteLine ("Yosh: " + yosh);
        Console.WriteLine ("Maosh: " + maosh);
    }
}
```

Natija:

Ism: Dilshod

Yosh: 25

Maosh: 2500.75

Yuqoridagi natijada ko‘rsatilganidek, C# tilida o‘zgaruvchilarni ma’lumotlar turiga qarab deklaratsiya qilish mumkin va ular yordamida dasturda ma’lumotlarni saqlash osonlashadi. Ma’lumotlarni kerakli vaqtda dasturga kiritish orqali dastur samaradorligi oshiriladi [3; 202].

Konstantalar va ularning dasturdagi afzalliklari

Konstanta -- bu o‘zgarmas qiymat bo‘lib, dastur davomida bir marta aniqlanadi va keyinchalik o‘zgartirilmaydi. Bu kodni boshqarish va qayta o‘qilishi oson bo‘lishiga katta hissa qo‘shadi. C# da konstantalar const kalit so‘zi bilan belgilanadi [4; 156].

Konstantani ishlatish misoli:

```
using System;
class Program
{
    const double Pi = 3.14159;
    static void Main()
    {
        double radius = 5.0;
```

```
double yuzasi = Pi * radius * radius;  
Console.WriteLine("Doira yuzi: " + yuzasi);  
}  
}
```

Natija:

Doira yuzi: 78.53975

Yuqoridagi natijalar shuni ko‘rsatadiki, konstantalarni ishlatish dasturda doimiy ma’lumotlarni boshqarishni osonlashtiradi. Bundan tashqari, konstantalar yordamida dasturda noto‘g‘ri qiymatlarni kiritishdan himoyalaniş mumkin, chunki ular dastur davomida o‘zgartirilmaydi [5; 75].

Operatorlar va ularning qo‘llanilishi

C# dasturlash tilida operatorlar arifmetik, mantiqiy, taqqoslash va boshqa turli xil amallarni bajarish uchun ishlatiladi. Operatorlarni to‘g‘ri ishlatish dasturda murakkab matematik va mantiqiy amallarni sodda shaklda bajarishga imkon beradi [6; 142].

Arifmetik operatorlar misoli:

```
using System;  
class Program  
{  
static void Main()  
{  
int a = 10;  
int b = 3;  
int natija = a + b;  
Console.WriteLine ("Yig'indi: " + natija);  
}  
}
```

Natija:

Yig‘indi: 13

Yuqoridagi natija arifmetik operatorlarning oddiy masalalarda qanday ishlatilishini ko‘rsatadi. Bundan tashqari, boshqa operatorlar, masalan, bo‘lish (/), ko‘paytirish (*), ayirish (-) ham keng qo‘llaniladi [7; 194].

Mantiqiy operatorlar misoli:

```
using System;  
class Program  
{  
static void Main ()  
{  
bool a = true;  
bool b = false;  
bool natija = a && b;  
Console.WriteLine ("Natija: " + natija);  
}  
}
```

Natija:

Natija: False

Yuqoridagi misol shuni ko‘rsatadiki, mantiqiy operatorlar yordamida dasturda turli shartlarni tekshirish va natijaga ko‘ra amallar bajarish mumkin [8; 88].

Massivlar va ular bilan ishlash

Massivlar bir xil turdagi ma’lumotlarni bir joyda saqlash imkonini beruvchi ma’lumotlar tuzilmasidir. Tadqiqot davomida bir o‘lchamli va ko‘p o‘lchamli massivlar ko‘rib chiqildi va ularning dasturda ishlatilishi samaradorligini oshirishi aniqlandi [9; 213].

Bir o‘lchamli massivdan foydalanishga misol:

```
using System;  
class Program  
{  
static void Main()  
{  
int [] sonlar = { 1, 2, 3, 4, 5 };  
foreach (int son in sonlar)  
{  
Console.WriteLine(son);  
}  
}
```

Natija:

1
2
3
4
5

Yuqoridagi natijalar ko'rsatadiki, massivlar yordamida bir xil turdagi ma'lumotlarni saqlash va ular bilan ishlash ancha qulaydir. Foreach operatori yordamida massiv elementlariga murojaat qilish va ularni ko'rish osonlashadi [10; 45].

Ko'p o'lchamli massivdan foydalanishga misol:

```
using System;
class Program
{
    static void Main()
    {
        int [,] matritsa = { { 1, 2 }, { 3, 4 }, { 5, 6 }
    };
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.Write (matritsa [i, j] + " ");
            }
            Console.WriteLine ();
        }
    }
}
```

Natija:

1 2
3 4
5 6

Ko'p o'lchamli massivlar katta hajmdagi ma'lumotlarni tartibli saqlash va ularga tezkor murojaat qilishda samarali vosita hisoblanadi. Natijalardan ko'rinib turibdiki, massivlarni

amaliyotda ishlatish natijadorlikni oshiradi va kodni o'qish osonlashadi [11; 157].

Metodlar va ularning dasturda roli

Metodlar – bu bir yoki bir nechta kod qatorini bir joyda yig'ish imkonini beruvchi funksiyalardir. Tadqiqot davomida metodlarni qo'llash orqali kodni takrorlamaslik va dasturiy loyihani modular shaklida yaratish imkoniyati o'rganildi [12; 38].

Metoddan foydalanishga misol:

```
using System;
class Program
{
    static void Main ()
    {
        Salomlashish (“Dilshod”);
    }
    static void Salomlashish (string ism)
    {
        Console.WriteLine (“Salom, “ + ism);
    }
}
```

Natija:

Salom, Dilshod

Yuqoridagilardan ma'lum bo'ldiki, metodlar yordamida dasturiy kodni modular shaklga keltirish va kerakli joylarda chaqirish juda samarali usuldir. Bu dasturda kodni qayta ishlatishni osonlashtiradi va ishlab chiqish jarayonini tezlashtiradi [13; 244].

Xulosa. C# dasturlash tili zamonaviy dasturlarni yaratishda qulay va samarali vosita hisoblanadi. O'zgaruvchilar, konstantalar, operatorlar va massivlar kabi asosiy tushunchalar dasturiy jarayonlarning asosi hisoblanadi. Ushbu maqolada keltirilgan misollar C# da dasturlashning dastlabki tamoyillarini tushunishga yordam beradi. Dasturchilar ushbu fundamental bilimlarga tayanib, katta dasturiy loyihalarni tuzishda



www.isft.uz

“ISFT” ILMIY-USLUBIY JURNAL
“ISFT” НАУЧНО-МЕТОДИЧЕСКИЙ ЖУРНАЛ
“ISFT” SCIENTIFIC-METHODOLOGICAL JOURNAL

ISSN: 3030-329X

2024/3-son



www.jurnal.isft-ilm.uz

muvaffaqiyat qozonishlari mumkin. Dasturlash jarayonida C# tili keng imkoniyatlar yaratadi va

yangi boshlovchilar uchun oson o‘rganiladigan til sifatida o‘zini namoyon qiladi.

Foydalanilgan adabiyotlar ro‘yxati:

1. Albahari B., Albahari J. C# 8.0 in a Nutshell. – O’Reilly Media, 2020.
2. Richter J. CLR via C# (4th ed.). – Microsoft Press, 2012.
3. Troelsen A., Japikse P. Pro C# 7: With .NET and .NET Core. – Apress, 2017.
4. Skeet J. C# in Depth (4th ed.). – Manning Publications, 2019.
5. Wagner B. Effective C#: 50 Specific Ways to Improve Your C#. – Addison-Wesley, 2021.
6. Freeman A. Pro ASP.NET Core 3. – Apress, 2020.
7. Sharp J. Microsoft Visual C# Step by Step (9th ed.). – Microsoft Press, 2018.
8. Price J. C# 7 and .NET Core: Modern Cross-Platform Development. – Packt Publishing, 2017.
9. Hart P. Beginning C# and .NET. – Apress, 2020.
10. Levitin A. Introduction to the Design & Analysis of Algorithms (3rd ed.). – Pearson, 2018.
11. Vermeulen G. Exploring Advanced C#: Enhancing C# Skills Beyond the Basics. – BPB Publications, 2019.
12. Marczyk G., DeMatteo D., Festinger D. Essentials of Research Design and Methodology. – John Wiley & Sons, 2005.
13. Duffy J. Professional C# 4.0 and .NET 4. – Wrox, 2008.