INNOVATIVE PUBLICATION

Journal of Effective Learning and Sustainable Innovation

Vol.3 №5 (2025). May

innovativepublication.uz

C# dasturlash tilida turli ma’lumotlar turlari va amallar bilan ishlash

Mirsaid Yusupov Abdulaziz og’li

Farg‘ona davlat unversiteti matematika kafedrasi o‘qituvchisi

@Mirsaidbek_Yusupov

Saydolimova Gulshanoy Umidjon qizi

Farg‘ona davlat unversiteti talabasi

saydolimovagulshanoy@gmail.com

Anotatsiya: Ushbu maqolada C# dasturlash tilidagi ma’lumotlar turlarining tasnifi,ularning xotira bilan ishlashdagi o‘rni va dasturlashdagi amaliy ahamiyati yoritilgan. Qiymat(value) va havola (reference) turlarining ishlash mexanizmi,ularning farqlari va dastur samaradorligiga ta’siri tahlil qilingan.Har bir ma’lumot turi misollar bilan yoritilib,ularning dastur tuzilmasidagi funksional roli ko‘rsatilib berilgan.Shuningdek,maqolada dasturiy resurslardan oqilona foydalanish, xususan,Garbage Collector tizimining afzalliklari muhokama qilingan.Mazkur tahlillar C# dasturlash tilida samarali va barqaror dasturlar yaratishda ma’lumotlar turlarini chuqur anglashning ahamiyatini ko‘rsatadi.

Kalit so‘zlar: qiymat turlari, havola turlari, xotira boshqaruvi,struct, class, ma’lumotlar turlari, Garbage Collector,dastur samaradorligi

Abstract: This article discusses the classification of data types in the C# programming language, their role in working with memory, and their practical importance in programming. The working mechanism of value and reference types, their differences,

and their impact on program performance are analyzed.Each data type is illustrated with examples and their functional role in the program structure is shown. The article also discusses the rational use of program recources, in particular, the advantages of the Garbage Collector system.These analyses demonstrate the importance of a deep understanding of data types in creating effective and stable programs in the C# programming language.

Keywords: value types,reference types, memory management, struct, class, data types, Garbage Collector, program performance

Аннотация: В статье рассмотривается классификация типов данных в языке програмирования C#, их роль в работе с памятью и практическое значение в програмировании. Анализируется механизм работы значемых и ссылочных типов, их различия и влияние на производитьеьность программы. Каждый тип данных проиллюстрирован примерами и показана их функциональная роль в структуре программы. В статье также рассматриваются вопросы рационального использования ресурсов программного обеспечения, в частности преимуществ системы Garbage Collector. Эти анализы демонстрируют и стабильных программ на языке программирования C#.

Ключевые слова: типы значений, ссылочные типы, управление памятью,

Структура,класс,типы данных, сборщик мусора, эффективность программы

C# dasturlash tili kuchli va obyektga yo‘naltirilgan ma’lumotlar haqida ko‘plab imkoniyatlar taqdim etuvchi dasturlash tilidir. C# dasturlash tilida asosan ma’lumotlar turlari ikkiga ga bo‘linadi.

1.Qiymat turlari (Value Types). Qiymat turlari ma’lumotlarni o‘zida to‘g’ridan to‘g’ri saqlaydi,ya’ni qiymat turidagi o‘zgaruvchi o‘zini qiymatini saqlaydi.Agar qiymat turlari o‘zgartirilsa,yangi qiymat kiritiladi va asl qiymat o‘zgarishsiz qoladi.Bu esa odatda xotiradan oson foydalanishga imkon beradi.

C# dasturlash tilida qiymatlar quyidagicha bo‘ladi:

int- butun sonlar tipi bo‘lib,butun sonlarni saqlash uchun ishlatiladi va U 32 bit (4 bayt) xotira egallaydi va -2,147,483,648 dan 2,147,483,647 gacha bo‘lgan qiymatlarni saqlay oladi. Ushbu tip asosan indekslar, hisoblagichlar yoki umumiy butun sonli qiymatlar uchun ishlatiladi.Uning xususiyatlariga quyidagilar kiradi:

Xotira o‘lchami: 4 bayt (32 bit)

Qiymatlar diapazoni: -2,147,483,648 dan 2,147,483,647 gacha

Butun sonlar bilan ishlash uchun qulay arifmetik amallar (qo‘shish, ayirish va hokazo) bajariladi.

Misol:

```
int age = 25;
int year = 2025;
int sum = age + year; // Natija: 2050
```

float- bu tip suzuvchi nuqtali sonlarni saqlash uchun ishlatiladi. Bu tip 32 bitli xotirani egallaydi va taxminan 6-7 ta raqam aniqlikni ta’minlaydi. Katta diapazonli, lekin nisbatan kam aniqlikdagi o‘nlik sonlar uchun qo‘llaniladi.Uning xususiyatlari:

Xotira o‘lchami: 4 bayt (32 bit)

Aniqlik: taxminan 6-7 raqam;

Katta va kichik kasrli sonlarni ifodalash uchun;

Misol uchun ilmiy hisob-kitoblarda qo‘llanilishi mumkin;

```
float pi = 3.14159f;
float tezlik = 12.5f;
```

float qiymatiga oxirida f qo‘yish kerak, chunki standartda suzuvchi nuqtali sonlar double sifatida qabul qilinadi.

double- bu qiymat turlarining suzuvchi nuqtali sonlarning aniqroq va kengroq diapazonga ega turi. U 64 bit (8 bayt) xotira egallaydi va taxminan 15-16 raqam aniqlikka ega bo‘lib, ko‘proq aniq ilmiy va muhandislik hisoblashlarida ishlatiladi.

Xususiyatlari:

Xotira o‘lchami: 8 bayt (64 bit);

Aniqlik: taxminan 15-16 raqam;

Katta diapazonli sonlar uchun qulay;

float ga qaraganda aniqroq va kengroq diapazonga ega ;

Misol:

```
double gravitatsiya = 9.80665;
double radius = 12345.6789;
```

decimal-yuqori aniqlikdagi o‘nlik sonlarni saqlash uchun mo‘ljallangan tip bo‘lib, ayniqsa moliyaviy hisob-kitoblarda keng qo‘llaniladi, chunki u raqamlarning qoldiqsiz va aniq ifodalanganini ta’minlaydi. decimal 128 bit (16 bayt) xotira egallaydi va taxminan 28-29 raqam aniqlikka ega.Uning xususiyatlari:

Xotira o‘lchami: 16 bayt (128 bit);

Aniqlik: 28-29 raqam;

Moliyaviy va tijorat hisob-kitoblar uchun ideal;

Qiymatlar o‘rtasida aniq taqqoslash va hisoblash imkonini beradi;

Misol:

```
decimal summa = 10000.75m;
decimal foiz = 0.045m;
```

decimal qiymatiga oxirida m yoki M qo‘yiladi;

char- ushbu tur bitta Unicode belgini saqlash uchun mo‘ljallangan qiymat turi bo‘lib,u 16 bitli xotira egallaydi va har qanday yozuv yoki maxsus belgini ifodalashi mumkin,u belgilar bilan ishlashda, masalan, harflar, raqamlar yoki maxsus belgilarni saqlashda ishlatiladi.

Xususiyatlari:

Xotira o‘lchami: 2 bayt (16 bit);

Unicode belgilarni saqlaydi;

Belgilar bilan bog‘liq operatsiyalar uchun ishlatiladi;

Misol:

```
char harf = 'A';
char raqam = '7';
char belgi = '#';
```

bool- ushbu tur mantiqiy qiymatlarni saqlash uchun mo‘ljallangan qiymat bo‘lib, faqat ikkita qiymatga ega bo‘lishi mumkin: true (rost) yoki false (yolg‘on) va u shartli ifodalar, sikllar va mantiqiy amallarda keng qo‘llaniladi.

Xususiyatlari:

Qiymatlar: true (rost) yoki false (yolg‘on);

Mantiqiy tekshiruvlar va shartlarni ifodalashda ishlatiladi;

Misol:

```
bool is True = true;
bool is Active = false;
```

struct-foydalanuvchi tomonidan aniqlangan qiymat turi hisoblanib, bir nechta bog‘langan ma’lumot maydonlarini (field) birlashtirish uchun ishlatiladi. Bu kichik, ko‘p maydonli ma’lumotlar to‘plamini yaratishda foydalidir. Struct qiymat turi hisoblanadi, shuning uchun u stackda saqlanadi va samarali ishlaydi.

Xususiyatlari:

Qiymat turi (value type);

Bir nechta bog‘langan maydonlarni o‘z ichiga oladi;

Kichik va tez ishlaydigan ma’lumotlar tuzilmasini yaratadi;

```
struct Nuqta
{
    public int X;
    public int Y;
}

Nuqta p;
p.X = 10;
p.Y = 20;
```

Yuqoridagi qiymat turlari bilan ishlaganda uning qiymati to‘g’ridan- to‘g’ri o‘zgaruvchida saqlanadi va uni boshqa o‘zgaruvchiga nusxalash orqali yangi mustaqil qiymat olinadi.Ushbu qiymat turlari C# dasturlash tilida tez-tez ishlatiladi.

Buning sababi esa ular kichik va samarali ekanligidadir.

2.Havola turlari (Reference Types)- tasavvur qilaylik, havola turi bu bizdagi katta bir kutubxona, lekin biz kitobni bevosita emas, balki uning joylashgan shkafi va raqamini olamiz. Shunday qilib, har safar kitobni o‘qishga kirganimizda, o‘sha shkafga borib, kerakli sahifalarni topamiz. Havola turlari hamxuddi shunday: ular haqiqiy ma’lumotni emas, balki uning joylashgan manzilini ko‘rsatadi. Havola turlari C# dasturida murakkab va ko‘p elementli ma’lumotlarni ifodalash uchun yaratilgan. Ular orasida eng keng tarqalganlari:

sinflar (class), **matnlar (string)**, **massivlar (array)** va **delegatlar (delegate)**.

Sinflar — dasturchi o‘zi yaratgan, o‘ziga xos xususiyatlar va funksiyalar to‘plami;

Matnlar — so‘z va gaplarni saqlash uchun maxsus havola turi;

Massivlar — bir turdagi ko‘p elementlardan tashkil topgan ketma-ketlik;

Delegatlar — funksiyalarni qamrab oluvchi maxsus ko‘rsatkichlar;

Havola turlarining asosiy xususiyatlaridan biri shundaki, ular haqiqiy ma’lumotni emas, balki unga yo‘l ko‘rsatuvchi manzilni saqlaydi. Bu esa bir nechta o‘zgaruvchilar bir xil obyektga havola qilishi mumkin degani. Demak, ulardan birini o‘zgartirsak, boshqasi ham o‘zgarishni sezadi, chunki aslida ular bitta ma’lumotni baham ko‘rmoqda.

Faraz qilaylik, bizda O‘quvchi nomli sinf bor. Unda talabaning ismi va yoshi haqida ma’lumotlar saqlanadi. Siz O‘quvchi1 nomli obyekt yaratdingiz. Endi O‘quvchi2 o‘zgaruvchisini O‘quvchi1 ga tenglasangiz, ular bir xil talaba haqidagi ma’lumotlarni ifodalaydi. Shunday qilib, O‘quvchi2 da ismini o‘zgartirsangiz, O‘quvchi1 ham o‘zgarganini ko‘rasiz, chunki aslida ular bitta talaba obyektiga havola qiladi.

Misol:

```
class Talaba
{
public string Ism;
public int Yosh;
}
Talaba talaba1 = new Talaba();
talaba1.Ism = "Ali";
talaba1.Yosh = 20;

Talaba talaba2 = talaba1;
talaba2.Ism = "Vali";

Console.WriteLine(talaba1.Ism); // Natija: Vali
```

Havola turlari dasturga qulaylik, samaradorlik va kuchli imkoniyatlar olib keladi. Ular yordamida murakkab ma’lumotlar tuzilmasini yaratish, bir nechta joyda bitta obyekt bilan ishlash va xotirani samarali boshqarish mumkin. C# ning o‘zida esa **Garbage Collector** (axlat yig‘uvchi) mavjud bo‘lib, foydalanilmay qolgan obyektlarni avtomatik tozalaydi, shu bilan dasturchini xotira boshqaruvidan ozod qiladi.

Havola turlari bu dasturdagi obyektlar dunyosining yo‘l xaritasidir. Ular qiymatni emas, balki qiymatga olib boruvchi manzilni eslab qoladi. Bu esa dasturchiga bir nechta

joydan bitta ma’lumot bilan ishlash, uni boshqarish, o‘zgartirish va nazorat qilishda katta erkinlik beradi. Dastur kodida sinflar, massivlar, matnlar, delegatlar kabi murakkab obyektlar aynan havola turlari orqali yaratiladi va boshqariladi.Havola turlarining yana bir afzalligi - ular orqali obyektga bog‘liq funksiyalar, metodlar va holatlarni birlashtirish, shuningdek, xotirani avtomatik boshqarish (Garbage Collector) orqali resurslarni samarali ishlatish imkoniyatidir. Bu esa dasturiy mahsulotlar ishlab chiqishda ishonchlilik, barqarorlik va tezlikni ta’minlaydi.Dasturlashda har bir tushuncha o‘zining amaliy ahamiyatiga ega bo‘lsa-da, havola turlarini chuqur anglash dasturchining fikrlash doirasini kengaytiradi. Bu orqali u kodga shunchaki buyruqlar to‘plami sifatida emas, balki bir butun tizim sifatida qaray boshlaydi. Har bir obyekt o‘zaro bog‘liq, har bir havola bir-birini to‘ldiruvchi zanjirga aylanadi.Xulosa qilib aytganda, havola turlari — bu C# tilining yuragidir. Ular orqali dastur nafaqat ishlaydi, balki yashaydi. Dasturchi esa bu yurak urishini eshitgan holda, undan ilhomlanib, yanada samarali, toza va kuchli kodlar yozishga intiladi.

Xulosa:

C# dasturlash tili – bu mantiq va tartibni o‘zida mujassam etgan zamonaviy til bo‘lib, unda dastur tuzilmasining poydevorini aynan ma’lumot turlari tashkil etadi. Har bir ma’lumot turi o‘z xususiyatiga ega bo‘lib, ular orqali kompyuter bilan aniq va lo‘nda muloqot qurish mumkin bo‘ladi.Ma’lumotlar ustida bajariladigan arifmetik, mantiqiy va taqqoslash amallari dasturga hayot bag‘ishlaydi — ular orqali dastur qaror qabul qiladi, yo‘nalishni tanlaydi va foydalanuvchiga kerakli natijalarni taqdim etadi. Bu jarayonlar dasturchiga tizimli fikrlashni, xatolardan saboq olishni va kodga ijodiy yondashishni o‘rgatadi.Xulosa qilib aytganda, C# tilidagi turli ma’lumotlar turlari va amallar dasturlashning asosiy tayanchi bo‘lib, ular dasturchining qo‘lidagi qudratli vositaga

aylanadi. Ushbu vositalarni to‘g‘ri tushunish va samarali qo‘llash - puxta, barqaror va zamonaviy dasturlar yaratish sari qo‘yilgan mustahkam qadamdir.

Foydalanilgan adabiyotlar:

1. Albahari, J., & Albahari, B. (2021). C# 10 in a Nutshell: The Definitive Reference. O'Reilly Media

2. Troelsen, A., & Japikse, P. (2022). Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. Apress.

3. Skeet, J. (2019). C# in Depth (4th ed.). Manning Publications

4. Richter, J. (2012). CLR via C# (4th ed.). Microsoft Press.

5. Microsoft Documentation: https://learn.microsoft.com/en-us/dotnet/csharp/

6. Lippert, E. (n.d.). Eric Lippert's Blog – Professional insights on C# internals and type system. https://ericlippert.com/