C# SHARP DASTURLASH TILI ORQALI FUNKSIYALAR VA METODLAR MAVZUSIGA OID MASALALARNI KO`RIB CHIQISH

Ubaydullayeva Diloromxon
Namangan davlat universiteti “Amaliy matematika va raqamli texnologiyalar” kafedrasi o‘qituvchisi
diloromubaydullayeva2010@gmail.com

ANNOTATSIYA

Ushbu maqolada C# sharp dasturlash tilida funksiyalar va metodlar mavzusi haqida ma`lumot, mavzular yuzasidan masalalarni C# sharp muhitida ishlanishlari keltirilgan.

Kalit so`zlar: *Funksiya, Metod tushunchasi, Metod yaratish usullari, return operatori, murojaat modifikatorlari, protsedura, void operatoti,public, identifikatorlar.*

ABSTRACT

This article provides information on the topic of functions and methods in the C# sharp programming language, and the processing of issues related to the topic in the C# sharp environment.

Keywords: *Function, Method concept, Method creation methods, return operator, reference modifiers, procedure, void operator, public, identifiers.*

АННОТАЦИЯ

В этой статье представлена информация о функциях и методах языка программирования C# Sharp, а также об обработке проблем, связанных с этой темой, в среде C# Sharp.

Ключевые слова: *Функция, концепция метода, методы создания метода, оператор возврата, модификаторы ссылки, процедура, оператор void, общедоступность, идентификаторы.*

Biz biroz yirikroq dastur yozish mobaynida juda ko’p bir xil bo’lgan vazifani qayta va qayta yozayotgandek bo’lamiz. Bu xuddi kuni bilan paxta terayotgan kishining ishiga o’xshaydi qaniydi bitta robot bo’lsayu unga bir marta paxta terishni o’rgatib qo’ysak keyin har safar aytganingda paxtani terib qo’ysa deb orzu qiladi paxta teruvchi , albatta terimchining bu orzusini amalga oshirish qiyin . Lekin sizning ishingizni (ya’ni kodlashda bir vazifani qayta yozishni) yengillashtirishning yo’li bor , bu yerda bizga Funksiyalar yordamga keladi.

Bir marta kod yozib uni dasturning istalgan joyida istalgancha ishlatish imkonini beruvchi operatorlar guruhi Funksiya deyiladi. Bu huddi yuqoridagi misolimizga

o’xshash bir marta qanday qilishni o’rgat va xoxlagan vaqting xoxlagan joyingda ishlat. Funksiya dasturchining kodini sezilarni darajada qisqartiradi va bu orqali dastur kodini o’qish osonlashadi va qotira hajmidan ham yutiladi.

C# tilida funksiyadan foydalanish. Dasturlashda funksiyalar ikki toifaga bo`linadi.

Qiymat qaytaruvchi.

Qiymat qaytarmaydigan.

Qiymat qaytaruvchi funksiyaning tuzilishi quydagicha:

<qaytaruvchi qiymat turi> <funksiya nomi> (<parametrlar>)

{

//funksiya tanasi

//return <qaytariladigan qiymat>

Bularning barchasiga alohida to’xtalib o’tamiz

1. - bu funksiya qaytaradigan qiymat bo’lib turli xil ma’lumot turlaridan foydalanishimiz mumkin masalan int, float string kabilar.

2. - funksiyaning nomi yani uni chaqiradigan ism deyishimiz ham mumkin unga ism tanlayotganda lotin alifbosidagi katta va kichik harflar , raqamlar va tagchiziq (‘_’) foydalangan holda , raqamdan boshlamasdan va c# tilidagi kalit so’zlardan foydalanmagan holda xoxlagan ismni berishimiz mumkin.

3. (<parametrlar>) – bu funksiya yakuniy qiymatni yaratish uchun foydalanadigan yordamchilar deyishimiz mumkin ushbu yordamchilar sifatida esa turli xil malumot tipidagi o’zgaruvchilar keladi . yordamchilar sifatida 0 tadan xoxlagancha o’zgaruvchilar bo’lishi mumkin ular quyidagicha yoziladi (int a, float b, int c) .

4. // funksiya tanasi - ushbu qism funksiyaning asosiy qismi hisoblaning qaytariladigan yakuniy qiymatgacha bo’lgan barcha ishlar shu yerda qilinadi va figurali qavs (‘{‘) bilan boshlanadi ushbu qismda parametr sifatida kiritilgan o’zgaruvchilar ham ishlatiladi.

5. // return - ushbu amal biz funksiyaning qaytaradigan qiymatini aniqlaydi. ning turi funksiyada elon qilingan dastlabki qiymat bilan bir xil bo’lishi shart.yani funksiya int turida bo’lsa biz return dan keyin ham int turidagi o’zgaruvchi bo’ladi. Ming marta eshitgandan ko’ra bir marta ko’rgan yaxshi deganlaridek biz C# tilida Funksiya ga oddiygana ikki sonda kattasini topuvchi funksiya orqali misol keltiramiz

Qiymat qaytarmaydigan funksiyalar

Agar biz funksiyalarni xizmatchi inson deb bilsak qiymat qaytaradigan va qaytarmaydigan funksiyalarni shunday tariflashimiz mumkin . Deylik xizmatchiga

qanchadir pul(ya'ni parametr) berib do'konga jo'natamiz va bizga aytgan narsamizni (qaytaradigan qiymat) ni olib keladi bu qiymat qaytaradigan funksiyaga misol bo'ladi . boshqa xizmatchiga esa pul(ya'ni parametr) berib unga hovlidagi qandaydir ishlarni aytamiz va u aytgan ishlaringizni qildi lekin sizga hech narsa qaytarib kelmadi bu qiymat qaytarmaydigan funksiyaga misol bo'la oladi. Qisqa qilib aytganda qiymat qaytarmaydigan funksiyaning qaytaruvchi qiymati bo'lmaydi va u qandaydir vazifani bajargan bo'ladi.

```
<void> <funksiya nomi> (<parametrlar>)
{
//funksiya tanasi
}
```

Funksiyaga oid masalalar ishlashga na`munalar.

1- Uchburchak bo`lish sharti.

```
using System;

public class Program
{
 static string uchburchak(double A, double B, double C)
 {
  if(A+B>C && B+C>A && A+C>B)
    return " Uchburchak bo`ladi";
  else return " Uchburchak bo`lmaydi ";
 }

 static void uchburchak2(double A, double B, double C, ref string S)
 {
  if(A+B>C && B+C>A && A+C>B)
    S=" Uchburchak bo`ladi";
  else S= " Uchburchak bo`lmaydi ";
 }
 public static void Main()
 {
  double A,B,C;
  string S="";
  A=double.Parse(Console.ReadLine());
  B=double.Parse(Console.ReadLine());
  C=double.Parse(Console.ReadLine());
```

```
    Console.WriteLine(uchburchak(A,B,C));
    uchburchak2(A,B,C,ref S);
    Console.WriteLine(S);
    }
}
```

Javobi:

```
2
5
4
Uchburchak bo‘ladi
Uchburchak bo‘ladi
```

2. Ihtiyoriy uchta sonning 3-darajasini hisoblovchi funksiya tuzing.

```
using System;

public class Program
{
  static void uchburchak2( ref double A, ref double B,ref  double C )
  {
    A=Math.Pow(A,3);
    B=Math.Pow(B,3);
    C=Math.Pow(C,3);
  }
  public static void Main()
  {
    double A,B,C;

    A=double.Parse(Console.ReadLine());
    B=double.Parse(Console.ReadLine());
    C=double.Parse(Console.ReadLine());
    uchburchak2(ref A, ref B,ref C);
    Console.WriteLine("A={0} B={1} C={2}" , A,B,C);
  }
}
```

Javobi:

5
7
9
A=125 B=343 C=72

3. Ikkita sonning o`rta arifmetigi va o`rta geometrigini hisoblovchi MEAN funksiyasini tuzing. MEAN funksiyasi orqali A,B,C,D sonlaridan (A,B) (A,C) (A,D) juftliklarining o`rta arifmetigi va o`rta geometrigini hisoblang.

```
using System;

public class Program
{
 static void MEAN(double a, double b, out double K, out double P)
      {
         K = (a + b) / 2;
         P = Math.Sqrt(a * b);
 }
 public static void Main()
 {

         double A, B, C, D;
         double a, b;
         A=double.Parse(Console.ReadLine());
         B=double.Parse(Console.ReadLine());
         C=double.Parse(Console.ReadLine());
         D=double.Parse(Console.ReadLine());
         MEAN(A, B,out a,out b);
         Console.WriteLine("{0} {1}",a,b);
         MEAN(A, C, out a, out b);
         Console.WriteLine("{0} {1}", a, b);
         MEAN(A, D, out a, out b);
         Console.WriteLine("{0} {1}", a, b);

 }
}
```

4
5
7

6

4.5 4.47213595499958

5.5 5.29150262212918

5 4.89897948556636

Metod va protsedura tushunchalari

Dasturlashda shunday xolatlarga duch kelinadiki, qandaydir dastur qismini dasturda bir necha marta ishlatishga to‘g‘ri keladi. Bunday xolatlarda dasturning sodda va ixcham ko‘rinishga keltirish uchun ko‘p foydalaniluvchi dasturiy kodni alohida dastur qilib, ya’ni protsedura yoki Metod ko‘rinishidagi qism dastur sifatida ifodalab olinadi. So‘ngra lozim bo‘lganda shu qism dasturga murojaat qilib kerakli natijaga erishiladi.

Protsedura-bu ma’lum vazifani bajarishga mo‘ljallangan qism dasturdir. Protseduraga asosiy dastur tanasidan murojaat qilish mumkin. Agar protsedura unga taqdim qilinuvchi o‘zgaruvchilar asosida qandaydir vazifani bajarishga mo‘ljallangan bo‘lsa bunday protseduralar parametrli protseduralar deb ataladi.

Metod – protseduraga o‘xshash qism dastur. Farqi, Metod uning tanasidagi qism dastur bajarilgandan so‘ng muayyan bir qiymatni qaytarishga mo‘ljallanganligi va Metodni u qiymat qaytarganligi sababli ifodalar tarkibida qo‘llash mumkinligidir.

C# da method, procedure, function kabi kalit so‘zlar yo‘q. C# da har qanday protsedurani Metod ko‘rinishida yozish mumkin, shuning uchun protsedura tushunchasi chiqarib tashlangan. Faqat, agar Metod qiymat qaytarmasligi va xuddi protsedura kabi vazifa bajarishi lozim bo‘lsa, u holda Metodni e’lon qilishda void xizmatchi so‘zidan foydalaniladi.

C# da Metodlar yaratish

C# da Metodni e’lon qilinishi sintaksisi quyidagicha:

[*modifikator][static] [void] <Qaytariluvchi qiymat tipi> <Metod nomi> <([argumentlar])>*

{

// Metod tanasi

}

bu yerda :

modifikator – Metodning foydalanish mumkin bo‘lgan ta’sir doirasini belgilab public, private, protected, internal kalit so‘zlari bo‘lishi mumkin. Bu to‘g‘rida keyinroq to‘xtalamiz;

Modifikator va Metod nomi orasida **static** kalit so‘zi bo‘lishi mumkin. Ushbu kalit so‘z Metodning statik ekanligini bildiradi. Statik Metodlar u joylashgan sinf

yuklanishi bilan avtomatik tarzda xotiraga yuklanadi doimiy tarzda xotirada joylashib o‘tiradi;

void – agar Metod protsedura sifatida qo‘llanilsa, ushbu kalit so‘zi qo‘yiladi;

Qaytariluvchi qiymat tipi – bu Metod qaytaruvchi qiymatning tipidir;

Metod nomi – Metodni dastur va ifodalar ichida qo‘llash uchun unga berilgan nom, ya’ni murojaat qilish nomi;

argumentlar – Metod tanasida hisoblashda kerak bo‘ladigan kattalik(parametr)lar. Agar argument sifatida aniq bir o‘zgarmas qiymat uzatilishi lozim bo‘lsa, shu qiymat tipi va ana shu qiymatni Metod tanasiga olib kiruvchi o‘zgaruvchi nomi yoziladi. Agar bunda argumentlar bir nechta bo‘lsa, ular o‘zaro vergul bilan ajratib yoziladi. Masalan:

```
public static int func1(double a, double b, int r, string s)
{
// Metod tanasi
}
```

Bu yerda func1-Metodga murojaat qilish nomi. a, b, r, s lar esa Metod tanasida hisoblash uchun zaruriy parametrlar. Ushbu parametrlar sifatida avval initsializatsiya qilingan o‘zgaruvchi, o‘zgarmas yoki konkret biror qiymat qo‘yish mumkin. Bu tariqa qo‘llanilgan parametrlardan Metod tanasida biror qiymatni hisoblash uchun foydalanish mumkin, lekin aynan ushbu parametrlarning qiymatini Metod tanasida o‘zgartirish mumkin emas.

Agar Metod o‘zini qaytaruvchi qiymatidan tashqari yana qandaydir boshqa bir qiymatni qaytarishi, ya’ni Metod bir martalik murojaat qilishda bir nechta qiymatni qaytarishi zarur bo‘lsa, buning uchun qiymatni Metod tanasidan olib chiqib ketishi uchun ham parametrlardan foydalaniladi. Faqat ushbu parametrlardan avval ref kalit so‘zini qo‘llash lozim.

Metodlarga oid masalalar ishlashga na`munalar.

1. **.** *s* va *t* haqiqiy sonlar berilgan. Hisoblansin

f(t,-2s, 1.17)+f(2.2, t, s-t)

Bu yerda $f(a,b,c)=\dfrac{2a-b-\sin c}{5+|a-b-c|}$.

```
using System;

public class Program
{
  public static double funk1(double a, double b, double c)
      {
```

```
            return (2 * a - b - Math.Sin(c) / (5 + Math.Abs(a - b - c)));
        }
    public static void Main()
    {
      double t, s, K = 0;
            t = double.Parse(Console.ReadLine());
            s = double.Parse(Console.ReadLine());
            K = funk1(t, -2 * s, 1.17) + funk1(2.2, t, s - t);

            Console.WriteLine(K);

    }
}
```

Javobi: **5**

4

17.4690368443056

2. *s* va *t* haqiqiy sonlar berilgan bo‘lsin. Hisoblansin

[g(1.2, s)+g(t, s) –g(2s-1, st)]/g(2t, 3s)

Bu yerda $g(a,b)=\dfrac{2a^2+3b}{a^2+2ab+3b^2+5a^2b^3+e^a-e^b}$.

```
using System;

public class Program
{
  public static double funk2(double a, double b)
      {
        return      ((2*a*a+3*b)/(a*a+2*a*b+3*b*b+5*a*a*b*b*b+Math.Exp(a)-
Math.Exp(b)));
      }
    public static void Main()
    {
     double a, b, t, s, K = 0;
            t = double.Parse(Console.ReadLine());
            s = double.Parse(Console.ReadLine());
            K=(funk2(1.2,s)+funk2(t,s)-funk2(2*s-1,s*t))/funk2(2*t,3*s);

            Console.WriteLine(K);
```

```
    }
}
```

Javobi: **5**

6

-4654.76253241082

FOYDALANILGAN ADABIYOTLAR (REFERENCES):

1. https://docs.dot-net.uz/
2. https://uzbekdevs.uz/
3. Halimova Dildora Hamidovna " C # dasturlash tilida sinflar ierarxiyasini tashkil etish bo'yicha uslubiy qo'llanma". Buxoro,2015-yil.
4. R.Qabulov, Sh.Nazirov "C va C ++ tili " .Toshkent-2013.
5. O.I.Jalolov, Sh.M.Sharipov "C # dasturlash tilida fayllar bilan ishlash" . Buxoro-2014 .
6. Н.А.Тюкачев , В.Т.Хлебостроев "C # Основы программирования". Москва -2018.
7. "Microsoft Visual Studio dasturi C# dasturlash tili".