

# برنامه نویسی پیشرفته C#

۷ مهر ۹۸  
ملکی مجد

# یادآوری

- عضویت در گروه تلگرام

- کانال تلگرام

- منابع، تمرین ها و اطلاع رسانی

- تمرین شماره صفر

- قسمتی از نمره پایان ترم را دارد

- مهلت تا ساعت ۱۰ شب دوشنبه

## مباحث

- یادآوری متد (از جلسه پیش)
- مثال از متد
- Bodied method
- آرایه
- متغیر و حوزه
- مفهوم کلاس و شی

# متد - Method

- پارامترهای ورودی
- مقدار پیش فرض default value
- named arguments

Return values •

Parameter passing •

- ref
- out

متد خواندن رشته ای غیر خالی از ورودی کنسول و برگرداندن آن

```
static string readString ( string prompt )
{
    string result ;
    do
    {
        Console.Write ( prompt ) ;
        result = Console.ReadLine ();
    } while ( result == "" ) ;
    return result ;
}
```

متد خواندن رشته ای غیر خالی از ورودی کنسول و برگرداندن آن

```
static string readString ( string prompt )
```

```
string name;
```

```
name = readString( "Enter your name : " );
```

```
    Console.Write ( prompt ) ;
```

```
    result = Console.ReadLine ();
```

```
    } while ( result == "" ) ;
```

```
    return result ;
```

```
}
```

متد خواندن عدد صحیحی در یک بازه خاص از ورودی کنسول  
و برگرداندن آن به صورت یک عدد صحیح  
(از متد خواندن رشته استفاده شده)

```
static int readInt ( string prompt, int low, int high )  
{  
    int result ;  
  
    do  
    {        string intString = readString (prompt) ;  
        result = int.Parse(intString);  
    } while ( ( result < low ) || ( result > high ) );  
  
    return result;  
}
```

متد خواندن عدد صحیحی در یک بازه خاص از ورودی کنسول  
و برگرداندن آن به صورت یک عدد صحیح  
(از متد خواندن رشته استفاده شده)

```
static int readInt ( string prompt, int low, int high )  
{
```

```
    int age;  
    age = readInt ( "Enter your age : ", 0, 100);
```

```
        result = int.Parse(intString);  
    } while ( ( result < low ) || ( result > high ) );
```

```
    return result;
```

```
}
```



```

using System;

class MethodLibraries
{
    static string readString(string prompt)
    {
        string result;
        do
        {
            Console.Write(prompt);
            result = Console.ReadLine();
        } while (result == "");
        return result;
    }

    static int readInt(string prompt, int low, int high)
    {
        int result;

        do
        {
            string intString = readString(prompt);
            result = int.Parse(intString);
        } while ((result < low) || (result > high));

        return result;
    }

    public static void Main()
    {
        string name;
        name = readString("Enter your name : ");
        Console.WriteLine("Name: " + name);

        int age;
        age = readInt("Enter your age : ", 0, 100);
        Console.WriteLine("Age: " + age);
    }
}

```

کد شماره ۱۳

```

using System;

class MethodLibraries
{
    static string readString(string prompt)
    {
        string result;
        do
        {
            Console.Write(prompt);
            result = Console.ReadLine();
        } while (result == "");
        return result;
    }

    static int readInt(string prompt, int low, int high)
    {
        int result;

        do
        {
            string intString = readString(prompt);
            result = int.Parse(intString);
        } while ((result < low) || (result > high));

        return result;
    }

    public static void Main()
    {
        string name;
        name = readString("Enter your name : ");
        Console.WriteLine("Name: " + name);

        int age;
        age = readInt("Enter your age : ", 0, 100);
        Console.WriteLine("Age: " + age);
    }
}

```

## کد شماره ۱۳

در نوشتن متدها  
همواره باید وضعیت های ایجاد خطا رو در نظر بگیرید،  
مدیریت خطاها را یاد خواهید گرفت

# Expression bodied method

- برخی متدها خیلی ساده هستند، نتیجه یک محاسبه (بدون منطق های اضافه) را بر می گردانند یا اینکه یک تک کار را انجام می دهند.
- سی شارپ یک صورت ساده شده برای چنین متدهایی دارد
- چنین متدهای می توانند پارامترهای ورودی و مقدار خروجی داشته باشند

## مثال

```
int addValues(int leftHandSide, int rightHandSide)
{
    // ...
    return leftHandSide + rightHandSide;
}
```

```
void showResult(int answer)
{
    // display the answer
    Console.WriteLine($"The answer is {answer}");
    return;
}
```

```
int addValues(int leftHandSide, int rightHandSide) => leftHandSide + rightHandSide;
```

```
void showResult(int answer) => Console.WriteLine($"The answer is {answer}");
```

## مثال

```
int addValues(int leftHandSide, int rightHandSide)
{
    // ...
    return leftHandSide + rightHandSide;
}

void showResult(int answer)
{
    // display the answer
    Console.WriteLine($"The answer is {answer}");
    return;
}
```

```
int addValues(int leftHandSide, int rightHandSide) => leftHandSide + rightHandSide;
```

```
void showResult(int answer) => Console.WriteLine($"The answer is {answer}");
```

# آرایه - Array


- Definition – تعریف
  - `int[] iArray = new int[100];`
  - `int[,] i2Array = new int[50,100];`
  - `int[, ,] i3Array = new int[50,200,100];`
- ایندکس آرایه از صفر شروع می‌شود

## حوزه یک متغیر Scope of variable


```
{  
    int i ;  
    {  
        int j ;  
    }  
}
```

- قسمتی از برنامه که متغیر وجود دارد
- از مکانی که متغیر تعریف شده است
- تا تمام شدن بلاکی که در آن تعریف شده است

```
{  
    int i ;  
    {  
        int j ;  
    }  
    j = 99 ;  
}
```



```
{  
    int i ;  
    {  
        int i ;  
    }  
}
```



## نمونه ای از بررسی حوزه متغیر

```
{  
    int i ;  
}  
{  
    int i ;  
    {  
        int j ;  
    }  
}
```



## متغیر محلی حلقه فور - For loop local variables

- متغیر تعریف شده در ابتدای تعریف حلقه
- تفاوت تعریف متغیر در ابتدای حلقه و در بدنه حلقه؟

```
for ( int i = 0 ; i < 10 ; i = i + 1 )  
{  
    Console.WriteLine ( "Hello" ) ;  
}
```

## متغیرهای یک متد محلی هستند

```
class LocalExample
{
    static void OtherMethod ()
    {
        local = 99; // this will not compile
    }

    static void Main ()
    {
        int local = 0;
        Console.WriteLine ("local is :" + local);
    }
}
```

# Data member in classes

- برای این که دو متد، متغیر مشترکی داشته باشند:
- متغیر را مربوط به کلاس آن ها باید تعریف کنیم.

# Data member in classes

```
class MemberExample
{
    // the variable member is part of the class
    static int member = 0 ;

    static void OtherMethod ()
    {
        member = 99;
    }

    static void Main ()
    {
        Console.WriteLine ("member is : " + member);
        OtherMethod();
        Console.WriteLine ("member is now : " + member);
    }
}
```

# مفهوم شی گرایی (مستقل از زبان برنامه نویسی C#)

- توضیح مفهوم به طور کلی
  - کلاس class
  - شی object
  - ساخت شی
- استفاده از داده و متدهای شی و کلاس
  - کلاس یکی داره
  - مربوط به یک شی خاص
- مخفی سازی (encapsulation lead to data hiding)
- وراثت inheritance
- چند ریختی polymorphism
- انتزاع interface