ادامه جزوه طراحی سیستم های شی گرا

- **آرایه های چند بعدی:** تعریف آن بصورت زیر است

نوع آرایه [,…,] نام آرایه = new نوع آرایه [طول بعد ۱ ,طول بعد۲,…..];

- int [,] x=new int [3,5];

- int [,,] x=new int [3,5,7];

- **آرایه های چند بعدی:** مقداردهی اولیه

نوع آرایه [,...,] نام آرایه = [تعداد عناصر بعد ۱ ,تعداد عناصر بعد۲ ,.....];

- int [,] x=new int [2,3]{1,2,0,6,5,3};

- **آرایه های چند بعدی:** دستیابی به عناصر

نام آرایه= [اندیس بعد ۱ ,اندیس بعد۲ ,.....];

- int x=a int [3,4]

textbox1.text=a[1,0].tostring();

پردازش آرایه دو بعدی:

- تصادفی: کاربر می تواند به هر عنصر آرایه دستیابی داشته باشد.
- سطری: عناصر سطرها به ترتیب پردازش می شوند.

```
int i,j;
for (i=0 ; i<m ; i++)
for (j=0 ; i<n ; j++)
a[i,j]=0;
```

- ستونی: عناصر ستون ها به ترتیب پردازش می شوند.

```
int i,j;
for (i=0 ; i<n ; i++)
for (j=0 ; i<m ; j++)
a[i,j]=0
```

❑ تولید اعداد تصادفی:

- Random نام متغیر= new random();

Random w=new random();

❑ آرایه های دندانه ای:

```
int [] [] array =new [3][];
Array[0]=new int[5];
Array[1]=new int[4];
Array[2]=new int[3];
```

پر شدن خانه های آرایه:

```
Array[0][2]=1;
Array[0][4]=2;
Array[0][3]=3;
Array[2][1]=10;
```

رشته ها:

❑ تعریف رشته :

- string نام متغیر;

string S1=“hello”;

❑ دستیابی به رشته:

.

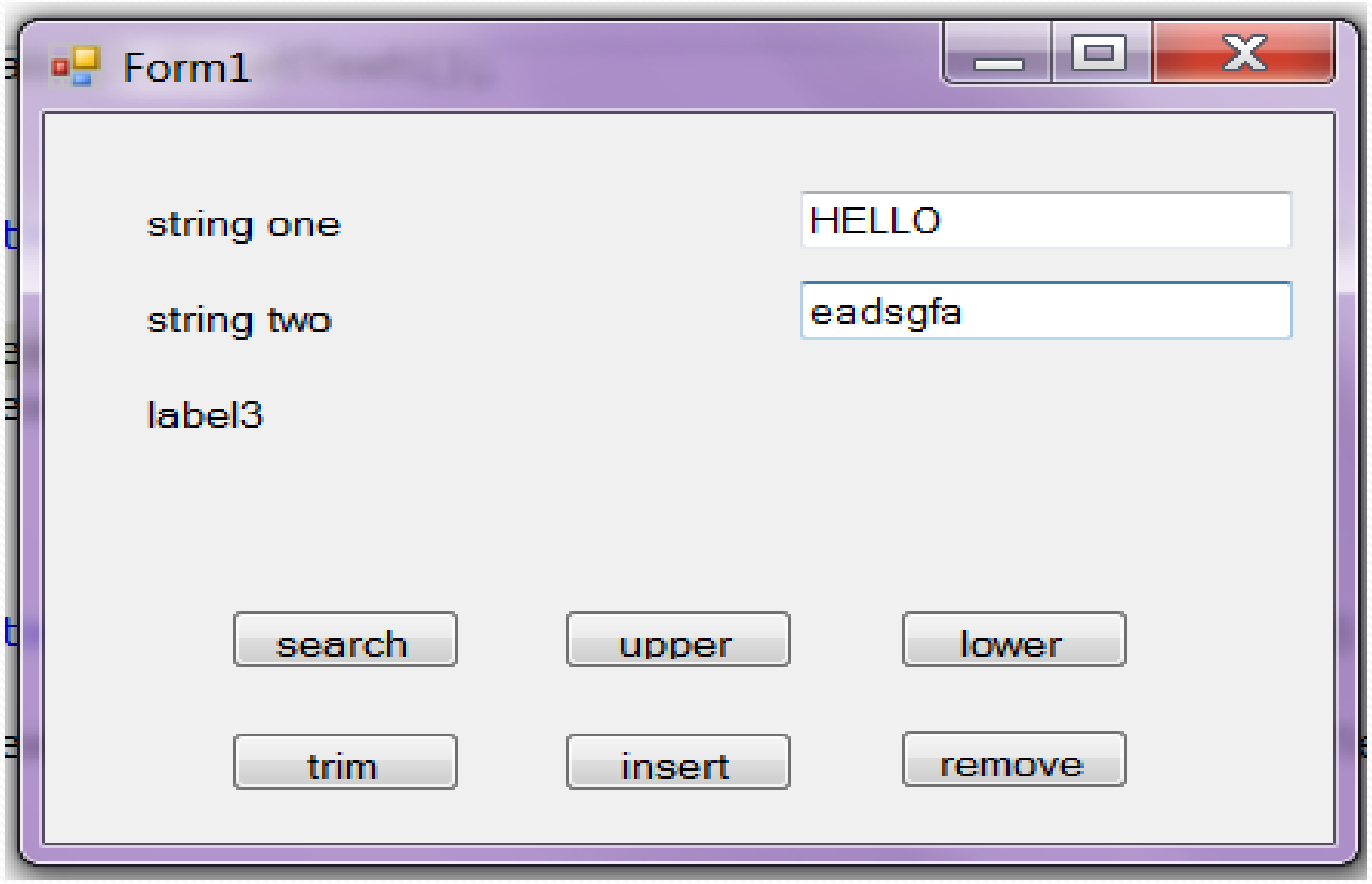
- string S2=“ali is ”;
  char ch =S2[2];

.

.

مثال: برنامه ای بنویسید که دو رشته را از ورودی خوانده و اعمال زیر را انجام دهد:

- **رشته اول را در رشته دوم جست و کرده و اولین مکان وقوع آن را نمایش دهد.**
- **رشته اول را به حروف بزرگ تبدیل کند.**
- **رشته دوم را به حروف کوچک تبدیل کند.**
- **فضای خالی سمت چپ رشته اول و دوم را خذف نماید.**
- **رشته اول را از کاراکتر سوم به بعد رشته دوم اضافه کند.**
- **پنج کاراکتر از کاراکتر چهارم به بعد را حذف نماید.**


Form1
string one
HELLO
string two
eadsgfa
label3
search
upper
lower
trim
insert
remove

```
private void button1_Click(object sender, EventArgs e)
    {
        string st1 = textBox1.Text;
        string st2 = textBox2.Text;
        int i = st2.IndexOf(st1);
        if (i > -1)
            label3.Text = "location is" + i.ToString();
        else
            label3.Text = "not found";
    }

    private void button2_Click(object sender, EventArgs e)
    {
        textBox1.Text = textBox1.Text.ToUpper();
        label3.ResetText();
    }
```

```
private void button3_Click(object sender, EventArgs e)
        {
            textBox1.Text = textBox1.Text.ToLower();
            label3.ResetText();
        }
        private void button4_Click(object sender, EventArgs e)
        {
            textBox1.Text = textBox1.Text.Trim();
            textBox2.Text = textBox2.Text.Trim();


        }
        private void button5_Click(object sender, EventArgs e)
        {
            textBox2.Text = textBox2.Text.Insert(3, textBox1.Text);
        }
        private void button6_Click(object sender, EventArgs e)
        {
            textBox1.Text = textBox1.Text.Remove(4, 5);
        }
    }
}
```

فصل سوم:

شی گرایی

ایجاد کلاس

```
[modifier] class name
{
Classmembers
}
```

```
public class test
{
 classmember
}
```

modifier: سطح دسترسی به کلاس را مشخص می کند. اینکه آیا فقط در داخل این پروژه دستیابی شود یا در پروژه های دیگر قابل دستیابی است.

۱. public: عدم وجود محدودیت در دستیابی، خارج از فضای نام قابل دسترسی است.

۲. internal(پیش فرض): کلاس فقط در همان فضای نامی که تعریف می شود قابل استفاده است.

نمونه سازی کلاس: ایجاد نمونه ای از کلاس را نمونه سازی و نمونه های کلاس را شی می نامند.

Classname obj=new classname(); test objtest= new test();

دسترسی به اعضای شیء: اگر سطح دسترسی عمومی باشد.

objtest.classMember

اعضای کلاس

اعضای کلاس از هر نوعی که باشند دارای یک سطح دسترسی اند:

[modifier] Classmembers

- Public(پیش فرض): معمولا برای متدها و خواص
- Private: معمولا برای فیلد ها و ثوابت

ثوابت

```
public class test
 {
private const int n=10;
Const int m=20 ,p=30;
}
```

اعضای کلاس

- **فیلدها(متغیر نمونه):** برای ذخیره داده ها بکار میروند.

[modifier] type fieldname[=value];

```
public class test
 {
private int x=100;
private float f;
}
```

اعضای کلاس

- **خواص:** از نظر کدهای خارج از کلاس، خاصیت مثل فیلد است ولی از نظر داخل کلاس، خاصیت ها متدهایی هستند که دو کار زیر را انجام می دهند:
  - داده های خود را مقدار می دهند.(set)
  - مقدار داده ها را بازیابی می کنند.(get)
  - برای اینکه خاصیت بتواند مقداری را به کد خارج از کلاس برگرداند از return استفاده می کند.

نمونه ای از کلاس test به همراه خاصیت hore

```
public class test
{
public int hour
{
get
{
return hour;
}
set
{
hour=value;
}
}
}
```

اعضای کلاس

- **متدها:** مجموعه ای از دستورالعمل ها است که عملیاتی را بر روی داده های کلاس انجام می دهد.

```
[modifier] type name[(parameters)]
{
Statements
}
public int sum (int x,int y)
 {
int s;
s=x+y;
return s;
}
int x=5,y=10;
int result;
result=sum(x,y);
```

پارامترهای مجازی

پارامترهای واقعی

تعریف متد

فراخوانی متد

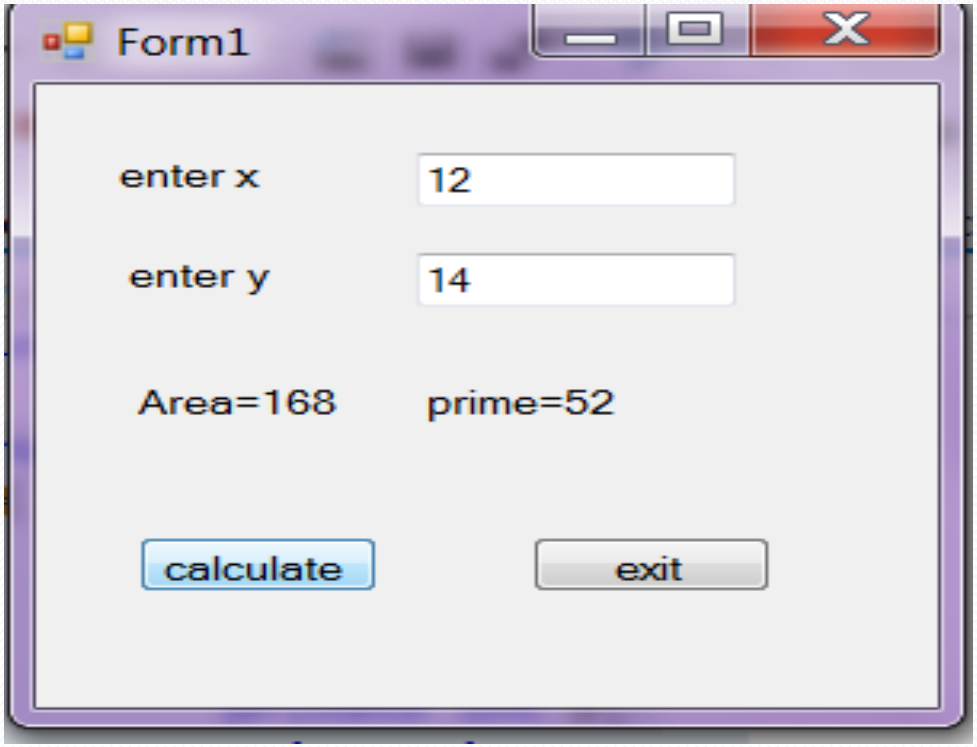
روش های ارسال پارامترها:

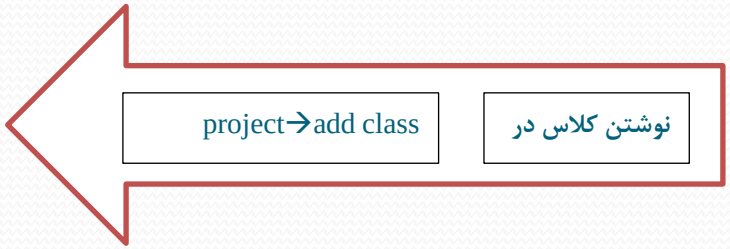
1. ارسال با مقدار: وقتی پارامتری به روش مقدار به متدی ارسال می شود، متد فراخوانی شده یک کپی از مقدار آن پارامتر را دریافت می کند. اگر در این کپی تغییراتی ایجاد شود در متد فراخوان تاثیری ندارد.

2. ارسال با ارجاع: وقتی پارامتری به روش ارجاع به متدی ارسال می شود، متد فراخوان، به متد فراخوانی شده این توانایی را می دهد که مستقیما به داده های اصلی در متد فراخوان دستیابی داشته باشد و آن را اصلاح کند.

.

مثال: برنامه ای بنویسید که طول و عرض مستطیلی را خوانده سپس محیط و مساحت آن را محاسبه کند.(با استفاده از کلاس)






```
namespace program 2
    class rectangle
    {
        private int x;
        private int y;
{

        private int area;
        private int prime;
        public int X
        {
            get
            {
                return x;
            }
            set
            {
                x = value;
            }
        }
        public int Y
        {
            get
            {
                return y;
            }
            set
            {
                y = value;
            }
        }
}
```

```
public void calculateAera()
        {
            area = x * y;
        }
        public void calculateprime()
        {
            prime = (x + y) * 2;
        }
        public int getArea()
        {
            return (area);
        }
        public int getprime()
        {
            return prime;
        }
    }
```

```
    private void button1_Click(object sender, EventArgs e)
    {
        rectangle rect = new rectangle();
        rect.X = System.Convert.ToInt16(textBox1.Text, 10);
        rect.Y = System.Convert.ToInt32(textBox2.Text, 10);
        rect.calculateAera();
        rect.calculateprime();
        label3.Text = "Area=" +
System.Convert.ToString(rect.getArea()) +  "       prime=" +
System.Convert.ToString(rect.getprime());

    }

    private void exit_Click(object sender, EventArgs e)
    {
        Close();
    }
```