سئوال

برنامه نویسی شی گرا یعنی چه و چه تفاوتی با برنامه نویسی ساخت یافته دارد؟

منظور از کلاس در برنامه نویسی شی گرا چیست؟

وقتی گفته می شود که یک زبان شی گرا هست منظور این است که این زبان سه ویژگی زیر را پشتیبانی می کند:
۱.کپسوله سازی:encapsulation

۲.وراثت:Inheritance

۳.چند ریختی:polymorphism

کپسوله سازی:

برنامه نویسی شی گرا (Oriented Programming Object یا OOP) داده ها (خصوصیات) و توابع(رفتار) را در بسته هائی به نام کلاس محصور می کند.و از این طریق جزئیات پیاده سازی در داخل خود کلاس ها پنهان هستند.(فقط اشیاء کلا سهای دیگر می دانند که فلان شی از فلان کلاس ،فلان رفتار را انجام میدهد ولی نمی دانند که این شی چگونه این رفتار را انجام می دهد)

وراثت:

یعنی یک کلاس از یک کلاس دیگر ارث می برد.ارث بری صورتی از قابلیت استفاده مجدد است.فرض کنید یک کلاس به نام دانشجو دارید که رفتار و خصوصیاتی را برایش تعریف کرده اید حالا می خواهید یک کلاس دانشجوی کارشناسی تعریف کنید . فکر می کنید کدام روش مناسبتر است : یک کلاس کاملا جدید تعریف کنید یا اینکه کلاس دانشجوی کارشناسی را نوعی از دانشجو قرار بدهید.

اگر روش دوم را انتخاب کنید یک برنامه نویس حرفه ای هستید. با این کار کلاس دانشجوی کارشناسی از کلاس دانشجو ارث می برد یعنی کلاس دانشجوی کارشناسی تمام خصوصیات یک دانشجو را به ارث می برد و علاوه بر آن می توانید خصوصیات و رفتار های دیگری را علاوه بر خصوصیات دانشجو ،که مختص یک دانشجوی کارشناسی است به آن اضافه کنید.

در این حالت به کلاس دانشجو <کلاس والدclass parent یا پایه >و به کلاس دانشجوی کارشناسی <کلاس مشتق شده drived class > گفته می شود .دو نوع ارث بری داریم:یگانه و چند گانه

چند ریختی:

فرض کنید مجموعه ای از کلاس های هندسی مثل دایره و مثلث و مستطیل دارید که همه از کلاس پایهshape مشتق شده ، هر کدام از این کلاسها فرمول ریاضی خاص خود را برای محاسبه مساحت دارند. حال فرض کنید در کلاس والد، رفتار(تابع) area تعریف شده باشد در نتیجه هر کدام از کلاسهای مشتق شده تابعarea مخصوص به خودشان را دارند ولی نام تمام آنها همان area مربوط به کلاس والد است.این امکان با استفاده از پشتیبانی یک زبان از polymorphism به وجود می آید.

تعریف شی:

یک شی شامل مجموعه ای از متدها و در برخی موارد شامل ویژگی ها و رخدادهاست

تعریف کنترل:

کنترل به اشیایی گفته می شود که بر روی فرم نمایان می شود و در زمان اجرا کاربر می تواند با آنها تعامل داشته باشد

تعریف کامپوننت:

کامپوننت ها اشیای خاصی هستند که در زمان طراحی می توان ویژگی های آنها را دستکاری و تغییر داد

ویژگی :

مجموعه ای از داده هاست که یک شی در طول دوره حیات خود نگهداری می کند

متد :

عبارت است از مجموعه ای از اعمال که یک شی قادر به انجام آن می باشد

رخداد:

عبارت است از مجموعه ای از وقایع که یک شی قادر به تشخیص آن می باشد

تعریف کلاس

کلاس یک ساختار داده ای پیچیده است که شامل مجموعه ای از داده ها و توابع مورد نیاز برای کار روی داده ها مـی باشـد . کلاس ها به ما این امکان را می دهند تا بتوانیم مجموعه داده و توابع و تکنولوژی کار با آن داده را به صورت کپسوله درآوریم . به داده هـا و توابع یک کلاس اعضای کلاس می گویند. این اعضا می توانند به صورت خصوصی ، عمومی و یا مجازی باشند.

یک عضو خصوصی مختص همان کلاسی است که متعلق به آن است و دسترسی به آن در خارج از محدوده کلاس امکان پـذیر نیست (فقط توابع عضو کلاس می توانند به عضوهای خصوصی کلاس دسترسی داشته باشند) بخش تعاریف خصوصی یک کلاس با کلیـد واژه `private` شروع می شود . در صورتی که عضو عمومی می تواند به ارث داده شود و دسترسی به آن هم در کلیه مشتقات کلاس امکان پذیر است بخش تعاریف عمومی یک کلاس با کلید واژه `public` شروع می شود.

یک عضو مجازی می تواند توسط وارث رونویسی شود و عملکرد آن مطابق میل و خواست وارث تغییر داده شود. تعاریف مجازی یک کلاس با کلید واژه `virtual` شروع می شود

عضو های سازنده و مخرب در یک کلاس :

عضو سازنده در یک کلاس وظیفه مقدار دهی اولیه مقادیر مربوط به داده های کلاس را بر عهده دارد نام سازنده (`constructor`) با نام کلاس یکسان است . یک کلاس می تواند بیش از یک سازنده داشته باشد

عضوب مخرب (destructor) در یک کلاس وظیفه آزاد سازی حافظه در اختیار گرفته شده توسط داده های کلاس را برعهده دارد. نام عضو مخرب همان نام کلاس است که علامت ~ قبل از آن قرار گرفته است

هر دوی این عضو ها در زیر مجموعه اعضای عمومی کلاس قرار دارند.

پرسش های پایان درس

۱ – کدام گزینه در مورد یک زبان شی گرا صحیح می باشد

الف) پشتیبانی از وراثت ب) پشتیبانی از چند ریختی ج) پشتیبانی از کپسوله سازی د) همه موارد

۲ – کدام گزینه در مورد تعریف شی صحیح نمی باشد

الف) دسترسی به اعضای خصوصی اشیا توسط سایر اشیاء دیگر امکان پذیر نمی باشد

ب) یک شی شامل مجموعه ای از متدها و در برخی موارد شامل ویژگی ها و رخدادهاست

ج) یک شی همانند یک متغیر عمومی است

د) دسترسی به اعضای عمومی اشیا توسط سایر اشیاء دیگر امکان پذیر می باشد

۳ – توابع عضو در کلاس .....

الف) به همه اعضای کلاس دسترسی دارند ب) فقط به اعضای عمومی دسترسی دارند

ج) فقط به اعضای عمومی و مجازی دسترسی دارند د) فقط به اعضای خصوصی دسترسی دارند

٤ – عضو عمومی در کلاس .....

الف) در تمام مشتقات کلاس در دسترس می باشد ب) در مشتقات کلاس می توان آن را رونویسی کرد

ج) فقط در داخل کلاس در دسترس می باشد د) فقط در داخل فضای نام فعلی در دسترس می باشد

٥– عضو مجازی در کلاس .....

الف) در کلاس های مشتق شده قابل استفاده است

ب) می تواند پارامترها و شکل کلی آن توسط وارث تغییر داده شود

ج) وارث نمی تواند ساختار داخلی عضو مجازی را مطابق سلیقه خودش تغییر دهد

د) الف و ج

٦– عضو عمومی در کلاس با کلید واژه ..... مشخص می شود

۷– عضو خصوصی در کلاس با کلید واژه ..... مشخص می شود

۸– عضو مجازی در کلاس با کلید واژه ..... مشخص می شود

۹– مجموعه ای از داده هاست که یک شی در طول دوره حیات خود نگهداری می کند.

۱۰ –.... عبارت است از مجموعه ای از اعمال که یک شی قادر به انجام آن می باشد.

۱۱ – .... مجموعه ای از رخدادها / وقایع است که یک شی قادر به تشخیص آن می باشد.

ایجاد کلاس

برای ایجاد کلاس در سی شارپ از کلید واژه class به شکل زیر استفاده می کنیم

```
[modifier]  class  classname
{
    class members
}
```

که در آن modifier سطح تعیین کننده سطح دستیابی به کلاس می باشد و یکی از کلید واژه های public و یا internal می باشد. وقتی سطح دستیابی public باشد دسترسی به کلاس از خارج از فضای نامی (NameSpace) که کلاس در آن تعریف شده باشد امکان پذیر است .سطح دستیابی internal مشخص می کند که کلاس فقط در همان فضای نام قابل دسترسی باشد.
Classname نامی است که توسط برنامه نویس برای کلاس انتخاب می شود و از قانون نامگذاری متغیرها تبعیت می کند

Class members اعضای کلاس را مشخص می کند. اعضای کلاس شامل داده ها و توابع عضو می باشد.

مثال : تعریف یک کلاس به نام Test

```
public  class  Test
{
…
}
```

نمونه سازی از کلاس

برای استفاده از کلاس باید نمونه ای از کلاس را ایجاد کنید. ایجاد نمونه ای از کلاس را نمونه سازی و نمونه ایجاد شده از کلاس را شی می گویند.

شکل کلی دستور ایجاد نمونه از یک کلاس به صورت زیر می باشد

```
ClassName objectName=new ClassName();
```

که در آن className نام کلاس و objectName نام نمونه / شی می باشد.

مثال : ایجاد شی ی به نام objTest از کلاس Test

```
Test objTest=new Test();
```

دسترسی به اعضای اشیاء

پس از اینکه یک نمونه از یک کلاس ایجاد شود اعضای عمومی (public) آن کلاس از طریق شی قابل دسترسی می باشد. برای دسترسی به این اعضا از نام شی / نمونه به همراه یک نقطه و سپس نام عضو عمومی مورد نظر عمل می کنیم

مثال : فرض کنید کلاس Test دارای یک عضو عمومی به نام x باشد و شی objTest از نوع کلاس Test تعریف شده باشد. در این صورت برای دسترسی به عضو x به شکل زیر عمل می کنیم

```
objTest.x
```

اعضای کلاس

اعضای کلاس می تواند یکی از موارد ثابت ها ، فیلد ها ، خواص ، متد ها ، سازنده ها ، مخرب ها و ... باشد

فرم کلی تعریف اعضای کلاس به شکل زیر می باشد

```
[modifier] classmembers
```

Modifier می تواند public و یا private باشد.

تعریف اعضای ثابت

اعضای ثابت با کلید واژه Const تعریف می شوند

مثال :

```
public class Test
{
  private const int x=10;
  public const double PI=3.14;
}
```

نکته : امکان تغییر مقادیر ثابت در ادامه برنامه وجود ندارد.

تعریف فیلد ها

فیلد ها همانند متغیر های معمولی تعریف می شوند و برای ذخیره داده ها به کار می روند. نام دیگر فیلد ها متغیر نمونه است.

فرم کلی تعریف فیلد ها به شکل زیر می باشد

```
[سطح دستیابی] نوع داده نام فیلد [=مقدار];
```

مثال :

```
public class Test
{
  private int x=10;
  public int y;
}
```

تعریف ویژگی ها / خواص

از نظر کد خارج از کلاس ویژگی ها همانند فیلد ها هستند و لی از نظر کلاس متفاوت می باشند. خاصیت ها دو کار را انجام می دهند ۱- داده های خود را مقدار می دهند ۲- مقدار داده ها را بازیابی می کنند.

مقدار دادن به خاصیت در قسمت set و بازیابی مقدار در قسمت get انجام می شود.

مثال :

```
public class Test
{
    private int _y;
    public int y
    {
      get
      {
          return _y;
      }
```

```
        set
        {
             _y=value;
        }
    }
}
```

نکته قابل توجه در مورد ویژگی ها این است که در موقع مقدار دهی در بخش set می توان کنترل نمود که داده های غیر مجاز توسط کاربر وارد نشود. نکته دیگر اینکه وقتی کد خارج از کلاس مقداری را به یک خاصیت نسبت دهد این مقدار توسط واژه ی کلیدی value قابل دسترسی می باشد.

نکته : چنانچه در تعریف یک ویژگی / خاصیت بخش get حذف شود آن ویژگی فقط نوشتنی و در صورتی که بخش set حذف شود آن ویژگی فقط خواندنی می باشد.

تعریف متد ها

متد ها مجموعه ای از دستورالعمل ها هستند که عملیاتی را بر روی داده های کلاس انجام می دهند. فرم کلی تعریف متد به صورت زیر می باشد

```
[سطح دستیابی] نوع داده نام متد ([پارامترها]);
```

مثال :

```
public class Test
{
   public int Sum(int x,int y)
  {
   int s;
   s=x+y;
   return s;
  }
}
```

متد های همنام

دو متد در صورتی می توانند همنام باشند که از نظر تعداد ،ترتیب و یا نوع پارامترها با هم متفاوت باشند

متد های بازگشتی

اگر یک متد مجددا خودش را فراخوانی کند به آن متد بازگشتی می گوییم.

عضو سازنده کلاس

عضو سازنده کلاس همنام خود کلاس می باشد و می تواند دارای پارامتر باشد. یک کلاس می تواند بیش از یک سازنده داشته باشد

مثال :

```
public class Test
{
   Private int a,b;
   public Test (){a=0;b=0;}
```

```
   public Test (int x,int y){a=x;b=y;}
}
```

عضو مخرب کلاس

عضو مخرب کلاس همنام خود کلاس می باشد که قبل از آن علامت ~ قرار دارد. یک کلاس نمی تواند بیش از یک عضو مخرب داشته باشد.عضو مخرب در زمانی که استفاده از شی ایجاد شده از کلاس تمام می شود به صورت خودکار اجرا می شود . بنابراین تمام مواردی که قرار است در پایان عمر شی انجام شود در این قسمت قرار می گیرد. مثلا فرض کنید یک کلاس برای کار با فایل ها ایجاد کرده باشید. عضو سازنده در ابتدای کار می تواند فایل را برای عملیات مورد نظر باز کند و در انتهای کار عضو مخرب باید عمل بستن فایل را انجام دهد

مثال :

```
public class Test
{
   Private int a,b;
   ~Test ()
    {
       …
    }

}
```

پرسش های پایان درس

۱- تفاوت کلاس public و کلاس internal در چیست؟

۲- به یک متغیر Public که در یک کلاس تعریف شده باشد..... می گویند

۳- از نظر ..... بین فیلد و ویژگی تفاوتی وجود ندارد

الف) طراح کلاس ب)استفاده کننده از کلاس ج) کلاس وارث د) ب و ج

٤- ایجاد نمونه ای از کلاس را ..... و نمونه ایجاد شده از کلاس را ..... می گویند.

الف) تعریف شی ، کلاس ب)امتغیر نمونه ، نمونه سازی ج) **نمونه سازی** ، شی د) نمونه سازی ، کنترل

٥-..... به یک متغیر Public اطلاق می شود که در یک کلاس تعریف شده باشد

الف) فیلد ب) ویژگی ج) متد د)ایستا

٦- وقتی کد خارج از کلاس مقداری را به یک خاصیت نسبت دهد این مقدار توسط واژه ی کلیدی ..... قایل دسترسی می باشد.

الف) set ب) get ج) value د)properties

۷ - کدام گزینه صحیح نمی باشد

الف) عضو سازنده در یک کلاس وظیفه مقدار دهی اولیه مقادیر مربوط به داده های کلاس را بر عهده دارد

ب) اعضای ثابت با کلید واژه Const تعریف می شوند

ج) از نظر کد خارج از کلاس ویژگی ها همانند فیلد ها هستند

د) اگر یک متد مجددا خودش را فراخوانی کند به آن متد همنام می گوییم

تمرین ۱ :

یک کلاس بنویسید که بتواند عمل جمع ، ضرب ، تفریق ، تقسیم و باقیمانده صحیح را بر روی دو عدد انجام دهد. کلاس دارای دو خاصیت به نام های x , y برای مشخص نمودن دو عدد و ۵ متد به نام های add و sub و mul و div و mod برای اعمال جمع ، ضرب ، تفریق ، تقسیم و باقیمانده صحیح باشد.

تمرین ۲:

یک کلاس برای انجام عملیات جمع ، تفریق و ضرب دو عدد n رقمی و محاسبه فاکتوریل یک عدد n رقمی بر پایه کار روی رشته های عددی ایجاد کنید

ایجاد کلاس ، استفاده از یک کلاس در کلاس دیگر ، وراثت از یک کلاس

ایجاد کلاس :

ایجاد یک کلاس ساده به نام Date با چند سازنده و دو متد به نام های GetDate و SetDate برای برگرداندن و مقداردهی متغیرهای داخلی کلاس

```
//-----------------------------------------------------------------------
public class Date
{
    private int M, D, Y;
    public Date() { M = 0; D = 0; Y = 0; }
    public Date(int year) { M = 0; D = 0; Y = year; }
    public Date(int year,int month) { M = month; D = 0; Y = year; }
    public Date(int year, int month,int day)
    { M = month; D = day; Y = year; }
    public string GetDate()
    {
       return Y.ToString() + "-" + M.ToString() + "-" + D.ToString();
    }
    public string SetDate(int year, int month,int day)
    {
       M = month; D = day; Y = year;
       return Y.ToString() + "-" + M.ToString() + "-" + D.ToString();
    }
}
//-----------------------------------------------------------------------
```

- استفاده از کلاس Date در کلاس Person
- نوشتن متد () GetPersonInfo برای برگرداندن مقادیر فیلد های کلاس Person

```
//-----------------------------------------------------------------------
public class Person
{
    string fname, lname;
    public Date birthDate;
    public Person()
    {
        Name = Family = "";
        birthDate = new Date();
    }
    public string Name{
      get{return fname;}
      set{fname=value;}
    }
    public string Family
    {
        get { return lname; }
        set { lname = value; }
    }

    public Person(string name,string family,Date BirthDate)
    {
        Name=name;
        Family = family;
        birthDate = BirthDate;
    }
    public string GetPersonInfo()
```

```
    {
        return Name+" ,"+Family+","+birthDate.GetDate();
    }
}
//-----------------------------------------------------------------------
```

- وراثت
- ایجاد کلاس Employee بر روی کلاس پایه Person
- کلاس Employee تمامی ویژگی های عمومی کلاس پایه را به ارث می برد
- ایجاد متد () GetEmployeeInfo برای برگرداندن مقادیر فیلد های کلاس Person

```
//-----------------------------------------------------------------------
public class Employee : Person
{
    public Employee():base() {  }
    public Date Start;
    public string GetEmployeeInfo()
    {
      return base. GetPersonInfo()+","+Start.GetDate();
    }

}
//-----------------------------------------------------------------------
```

نحوه استفاده از کلاس Employee

```
//-----------------------------------------------------------------------
        Employee emp=new Employee();
        emp.Name = "ali";
        emp.Family = "ahmadi";
        emp.birthDate = new Date(1360, 01, 01);
        emp.Start = new Date(1388, 01, 01);

       MessageBox.Show(emp.GetEmployeeInfo(),"Emp");
//-----------------------------------------------------------------------
```

تعریف و رونویسی متد های مجازی

تعریف کلاس Shape با یک متد مجازی به نام Area() برای برگرداندن مساحت

```
//-----------------------------------------------------------------------
class Shape
{
    public virtual double Area() { return 0; }
}
//-----------------------------------------------------------------------
```

تعریف کلاس Circle با رونویسی متد مجازی محیط برای محاسبه محیط دایره

```
public class Circle : Shape
{
    double r;
    public double Radius
    {
        get { return r; }
        set { r = value; }
    }
    public override double Area() { return r * r * 3.14159; }
}
//-----------------------------------------------------------------------
```

مثال :

کلاس های تعریف شده قبلی را طوری باز نویسی کنید که متد ToString () مربوط به هر کلاس اعمال خواسته شده در زیر را انجام دهد

- رونویسی متد ToString () در کلاس Date برای بازگرداندن تاریخ
- رونویسی متد ToString () در کلاس Person برای بازگرداندن نام ، نام خانوادگی و تاریخ تولد شخص با جداکننده کاما
- رونویسی متد ToString () در کلاس Employee برای بازگرداندن نام ، نام خانوادگی و تاریخ تولد و تاریخ استخدام کارمند با جداکننده کاما

ایجاد یک کلاس ساده به نام Date با چند سازنده و دو متد به نام های GetDate و SetDate برای برگرداندن و مقداردهی متغیرهای داخلی کلاس

```
//-----------------------------------------------------------------------
public class Date
{
    private int M, D, Y;
    public Date() { M = 0; D = 0; Y = 0; }
    public Date(int year) { M = 0; D = 0; Y = year; }
    public Date(int year,int month) { M = month; D = 0; Y = year; }
    public Date(int year, int month,int day)
    { M = month; D = day; Y = year; }
    public string GetDate()
    {
       return Y.ToString() + "-" + M.ToString() + "-" + D.ToString();
    }
    public string SetDate(int year, int month,int day)
    {
       M = month; D = day; Y = year;
       return Y.ToString() + "-" + M.ToString() + "-" + D.ToString();
```

```
    }
    public override string ToString()
    {
        return GetDate();
    }
}
//-----------------------------------------------------------------------
```

- استفاده از کلاس Date در کلاس Person
- رونویسی متد مجازی ()ToString برای برگرداندن مقادیر فیلد های کلاس Person

```
//-----------------------------------------------------------------------
public class Person
{
    string fname, lname;
    public Date birthDate;
    public Person()
    {
        Name = Family = "";
        birthDate = new Date();
    }
    public string Name{
      get{return fname;}
      set{fname=value;}
    }
    public string Family
    {
        get { return lname; }
        set { lname = value; }
    }

    public Person(string name,string family,Date BirthDate)
    {
        Name=name;
        Family = family;
        birthDate = BirthDate;
    }
    public override string ToString()
    {
        return Name+" ,"+Family+","+birthDate.ToString ();
    }
}
//-----------------------------------------------------------------------
```

- وراثت
- ایجاد کلاس Employee بر روی کلاس پایه Person
- کلاس Employee تمامی ویژگی های عمومی کلاس پایه را به ارث می برد
- رونویسی متد مجازی ()ToString برای برگرداندن مقادیر فیلد های کلاس Employee

```
//-----------------------------------------------------------------------
public class Employee : Person
{
    public Employee():base() {  }
    public Date Start;
    public override string ToString()
    {
```

```
        return base.ToString()+","+Start.GetDate();
    }

}
//-----------------------------------------------------------------------
```

نحوه استفاده از کلاس Employee

```
//-----------------------------------------------------------------------
        Employee emp=new Employee();
        emp.Name = "ali";
        emp.Family = "ahmadi";
        emp.birthDate = new Date(1360, 01, 01);
        emp.Start = new Date(1388, 01, 01);

       MessageBox.Show( emp.ToString(),"Emp");
//-----------------------------------------------------------------------
```

تمرین:

متدی به نام DiffDate به کلاس Date اضافه کنید که تفاضل دو تاریخ شمسی را از هم کم کند و حاصل را به صورت روز و ماه و سال برگرداند.

جواب) با فرض اینکه تاریخ d2 بزرگتر از تاریخ d1 باشد می توان جواب را به شکل زیر نوشت

```
//-----------------------------------------------------------------------
    public Date Diff(Date d2, Date d1)
    {
        if (d1.D > d2.D)
        {
            d2.D += (d2.M <= 7) ? 31 : 30;
            d2.M--;
            if (d2.M == 0)
             {
                d2.Y--;
                d2.M = 12;
             }
        }
        if (d1.M > d2.M)
        {
            d2.Y--;
            d2.M += 12;
        }
        return new Date(d2.Y - d1.Y, d2.M - d1.M, d2.D - d1.D);
    }
//-----------------------------------------------------------------------
```

راه حل دوم

```
//-----------------------------------------------------------------------
    public Date Diff(Date d2, Date d1)
    {
        if (d1.D > d2.D)
        {
            d2.D += (d2.M <= 7) ? 31 : 30;
            d2.M--;
            d2.Y = d2.Y - ((d2.M == 0) ? 1 : 0);
            d2.M = (d2.M == 0) ? 12 : d2.M;
        }
        if (d1.M > d2.M)
        {
            d2.Y--;
            d2.M += 12;
        }
        return new Date(d2.Y - d1.Y, d2.M - d1.M, d2.D - d1.D);
    }
//-----------------------------------------------------------------------
```

برنامه ای بنویسید که با استفاده از کلاس Employee نام، نام خانوادگی،تاریخ تولد و تاریخ شروع به کار ۱۰ نفر از پرسنل را گرفته و سپس نمایش دهد.(راهنمایی : آرایه ای از نوع Employee تعریف کنید)

پرسش های پایان درس

۱- متد مجازی پیش فرض در همه کلاس ها ..... می باشد

الف) ()ToString ب) متد سازنده ج) متد مخرب د) متد مجازی

۲- بعد از اجرای کامل دستورات زیر مقدار متغیر k چیست

```
int k=5;
for(int i=0;i<10;i++);
        for(int j=5;j<20;j++) j++;
```

الف) ۱۵۵ ب) ۲۰ ج) ٦ د) ٥

۳- بعد از اجرای کامل دستورات زیر مقدار متغیر k چیست

```
int k=5;
for(int i=0;i<10;i++);
        for(int j=5;j<20;j++) k++;
```

الف) ۱۵۵ ب) ۲۰ ج) ٦ د) ٥

٤- خروجی متد abc چیست

```
private int abc(int x, int y)
{
  int t;
  while(x%y) {   t=y;  y =x%y;   x=t;   }
  return y;
}
```

الف) محاسبه ک.م .م بین دو عدد x , y ب) محاسبه ب.م .م بین دو عدد x , y

ج) محاسبه باقیمانده تقسیم x بر y د) محاسبه خارج قسمت x بر y

٥- نتیجه اجرای قطعه کد زیر چیست

```
Class MyClass{
 public int a(int x){return x%2==0;}
 public int b(int x){if(a(x)) return x; else return 0; }
}
MyClass  x=new MyClass();
bool y=x.a(10);
```

الف) متغیر y دارای مقدار true می شود

ب) تعریف کلاس ایراد دارد

ج) دسترسی به متد a با توجه به تعریف کلاس امکان پذیر نیست و باید از متد b به جای آن استفاده شود

د) برنامه با خطا متوقف می شود

٦- یک کلاس طراحی کنید که :

- **دارای دو ویژگی برای تعیین دو عدد صحیح باشد**
- **متدی به نام BMM برای تعیین بزرگترین مقسوم علیه مشترک بین دو عدد موجود در ویژگی های فوق باشد**
- **متدی به نام KMM برای تعیین کوچکترین مضرب مشترک بین دو عدد موجود در ویژگی های فوق باشد**

۷ – متدی بنویسید که بتواند یک رشته عددی را دریافت و متتم ۹ آن را برگرداند.

۸- کد ملی یک شماره ۱۰ رقمی است که رقم سمت راست آن از روی سایر ارقام بدست می آید.بدین ترتیب که از سمت راست ارقام موجود در موقعیت۲ تا ۱۰ آن در موقعیت خودشان ضرب می شوند و سپس نتیجه ها با هم جمع شده و در نهایت عدد حاصل بر ۱۱ تقسیم می شود. اگر باقیمانده عددی کمتر از ۲ باشد در این صورت رقم سمت راست باید برابر باقیمانده باشد در غیر اینصورت رقم سمت راست باید برابر یازده منهای باقیمانده باشد. یک کلاس طراحی کنید که دارای متدی عمومی برای چک کردن اعتبار کد ملی باشد. (کد ملی را بگیرد و در صورت اعتبار مقدار `true` و در غیر اینصورت مقدار `false` را به عنوان نتیجه برگرداند)

مثال : فرض کنید می خواهیم اعتبار کد ملی ۰۷٤۸۸۲۰۲۱۳ را بررسی کنیم . در این صورت طبق جدول زیر ارقام ۲ تا ۱۰ کد را در موقعیت شان ضرب کرده و حاصل را با هم جمع می کنیم (۲+٦+۰+۱۰+٤۸+٥٦+٦۳+۰ برابر است با ۲۱۷، سپس ۲۱۷ را بر ۱۱ تقسیم می کنیم . باقیمانده این تقسیم برابر ۸ است که کمتر از ۲ نیست پس رقم سمت راست کد ملی باید ۸-۱۱ باشد که در مورد این کد رقم سمت راست ۳ می باشد. با این حساب این کد ملی به عنوان یک کد ملی معتبر مورد قبول است)

ارقام کد ملی	۰	۷	٤	۸	۸	۲	۰	۲	۱	۳
موقعیت رقم	۱۰	۹	۸	۷	٦	٥	٤	۳	۲	۱
حاصل ضرب	۰	٦۳	۳۲	٥٦	٤۸	۱۰	۰	٦	۲	

ایجاد یک کلاس برای انجام عملیات جمع ، تفریق و ضرب دو عدد n رقمی و محاسبه فاکتوریل یک عدد n رقمی بر پایه کار روی رشته ها

```
class MyMath
{
    //تبدیل رشته به طول خاص با افزودن صفر به سمت چپ رشته
    private string FixlenTo(string x, int len)
    {
        while (x.Length < len) x = "0" + x;
        return x;
    }
    //حذف صفرهای سمت چپ رشته
    private string RemoveLeftZero(string x)
    {
        int i=0;
        int L=x. Length;
        while (i<L && x.Substring(i, 1).Equals("0")) i++;
        if(i<L)
            return x.Substring(i);
        else
            return "0";
    }
    //جمع دو عدد یک رقمی و برگرداندن نتیجه به صورت یک رقمی
    // و در صورت لزوم یک رقم ده بر یک
    private int FullAdder(int a, int b, int c, out int d)
    {
        int r = a + b + c;
        d = r / 10;
        r = r % 10;
        return r;
    }
    // جمع دو عدد
    public string Add(string x, string y)
    {
        int L = x.Length;
        if (L < y.Length) L = y.Length;
        x = FixlenTo(x, L);
        y = FixlenTo(y, L);
        int d = 0;
        int i = L - 1;
        String r = "";
        int a, b,c;
        while (i >= 0)
        {
            c = d;
            a = Convert.ToInt16(x.Substring(i, 1));
            b = Convert.ToInt16(y.Substring(i, 1));
            r = FullAdder(a, b, c,out d) + r;
            i--;
        }
        if (d > 0)
            r = d.ToString() + r;
        return r;
```

```
}
 //بدست آوردن متمم عدد
private string NComplete(string n)
{
    int a,L=n.Length;
        string r="";
    for(int i=0;i<L;i++)
    {
      a =9- Convert.ToInt16(n.Substring(i, 1));
      r=r+a.ToString();
    }
    return r;
}
//تفریق دو عدد
public string Sub(string x, string y)
{
    int L = x.Length;
    if (L < y.Length) L = y.Length;
    string r;
    x = FixlenTo(x, L);
    y = FixlenTo(y, L);
    y= NComplete(y);
    r=Add(x,y);
        if(r.Length>L)
        r = RemoveLeftZero(Add(r.Substring(1, L), "1"));
    else
        r = "-" + RemoveLeftZero(NComplete(r));
    return r;
}
 //ضرب دو عدد یک رقمی
private int FullMul(int a, int b, int c, out int d)
{
    int r = a * b + c;
    d = r / 10;
    r = r % 10;
    return r;
}
// ضرب یک رقمی در چند رقمی
private string Mul1ToN(string x, int b)
{
    if(b==0) return "0";
    int L = x.Length;
    int d = 0;
    int i = L - 1;
    String r = "";
    int a, c;
    while (i >= 0)
    {
        c = d;
        a = Convert.ToInt16(x.Substring(i, 1));
        r = FullMul(a, b, c,out d) + r;
        i--;
    }
```

```
        if (d > 0)
            r = d.ToString() + r;
        return r;
    }
     ضرب دو عدد چند رقمی//
    public string Mul (string x, string y)
    {
        int L = y.Length;
        int i = L - 1;
        String r1="",r = "0";
        int b, j=0;

        while (i >= 0)
        {
            b = Convert.ToInt16(y.Substring(i, 1));
            r1 = Mul1ToN(x, b)+FixlenTo("",j);
            r=Add(r,r1);
            i--;
            j++;
        }
        return r;
    }

    محاسبه فاکتوریل یک عدد//
    public string Factorial(string x)
        {
            int L = x.Length;
            //int i = L - 1;
            String  Stop,r = x;
            Stop = "1";//FixlenTo("", L - 1) + "1";
            while (x!= Stop)
            {
                x=Sub(x,"1");
                r = Mul(r, x);
            }
            return r;
    }

}
```

مثال :نحوه ی استفاده از کلاس

```
MyMath m=new MyMath();
textBox1.Text = m.Add("999", "1");
textBox2.Text = m.Sub ("123", "200");
textBox3.Text = m.Mul("1000", "700");
textBox4.Text = m.Factorial("10");
```

تمرین :

با توجه به مطالبی که تا کنون آموخته اید و استفاده از کلاس MyMath برنامه ای بنویسید که اعمال جمع ، تفریق و ضرب را بر روی دو عدد n و m رقمی انجام و نتیجه را در یک textbox نمایش دهد . برنامه همچنین قادر باشد تا یک عدد n رقمی را دریافت و فاکتوریل آن را محاسبه و در یک textbox نمایش دهد

نحوه کار با ListView ها

نحوه افزودن عنوان ستون ها

```
ColumnHeader c1 = new ColumnHeader();
c1.Text = "111";
ColumnHeader c2 = new ColumnHeader();
c2.Text = "222";
ColumnHeader c3 = new ColumnHeader();
c3.Text = "333";
listView1.Columns.Clear();
listView1.Columns .Add(c1);
listView1.Columns .Add(c2);
listView1.Columns.Add(c3);
listView1.View = View.Details;
```

نحوه افزودن داده به لیست

```
ListViewItem lst = new ListViewItem();
lst.Text = "123";
lst.SubItems.Add("456");
lst.SubItems.Add("789");
listView1.Items.Add(lst);
```

نوشتن متد برای یافتن یک آیتم در لیست

```
//-------------------------------------------------------------
private bool FindItem(string str)
{
    foreach (ListViewItem item in this.listView1.Items)
    {
        //جستجو روی آیتم موجود در موقعیت ستون چهارم
        if (item.SubItems[3].Text.Equals(str))
        {
            return true;
        }
    }
    return false;
}
//-------------------------------------------------------------
```

نحوه استفاده از متد فوق

```
if (this.FindItem(this.txtBarcode.Text))
{
     MessageBox.Show("سعي مجددا لطفا . است تكراري توليد شماره
كنيد");
     this.txtBarcode.Text = string.Empty;
     this.txtBarcode.Focus();
}
else
                ……
```

نوشتن متد برای حذف آیتم از لیست

```
private void btnDeleteRecord_Click(object sender, EventArgs e)
{
    int i = 0;
    foreach (int p in this.listView1.SelectedIndices)
    {
        this.listView1.Items.RemoveAt(p - i);
        i++;
    }

}
```

فرض کنید یک فرم با کنترل های txtTarikh برای ورود تاریخ فاکتور ، txtShomFactor برای شماره فاکتور ، txtCodeMosht برای ورود کد خریدار ، txtBarcode برای ورود شماره تولید ، txtArzesh برای ورود ارزش هر متر مربع فرش ، کامبوباکس cmbVahedArz برای تعیین واحد ارز مورد استفاده (ریال ، دلارآمریکا ، یورو ، فرانک سوئیس ، پوندانگلیس ، درهم امارات ، ین ژاپن ، دلارکانادا ، لیرترکیه ، ریال سعودی ، دینارکویت) و یک listView و دکمه هایی برای خواندن از فایل(btnLoadFromFile) ، ثبت اطلاعات در فایل (btnSabt) ، حذف رکورد(btnDeleteRecord) و خروج (btnClose) و دیالوگ های بازکردن فایل(OpenFileDialog) و ذخیره فایل (SaveFileDialog) را داشته باشیم و ستون های لیست به ترتیب تاریخ ، شماره فاکتور ، کد خریدار ، کد واحد ارز ، ارزش هر متر و شماره تولید باشد. مطلوبست نوشتن برنامه ای که بتواند اطلاعات را دریافت و به صورت غیر تکراری در لیست ذخیره نماید و در صورت کلیک کاربر بر روی هر یک از دکمه های اشاره شده در فوق اعمال مورد نظر شامل ذخیره داده ها در فایل ، لود کردن داده ها از فایل ، حذف رکوردهای انتخابی و ... را انجام دهد.

برخی از متد های مورد نیاز در زیر آورده شده است

ذخیره اطلاعات listView در فایل متنی

```
 private void btnSabt_Click(object sender, EventArgs e)
 {
     btnSabt.Enabled = false;
     if (saveDialog1.ShowDialog() == DialogResult.Cancel) return;
     saveDataToFile();
     btnSabt.Enabled = true;
}
```

روال ذخیره اطلاعات در فایل

```
private void saveDataToFile()
{
   using (StreamWriter writer = new StreamWriter(saveDialog1.FileName,
   false, Encoding.UTF8))
   {
        string s;
        //افزودن نام و تعداد ستون ها و سطرهای لیست به سطر اول فایل
        s="Factor\t" + listView1.Columns.Count.ToString() + "\t";
        s+= listView1.Items.Count.ToString();
        writer.WriteLine(s);
```

```
        int i, n;
        n = listView1.Columns.Count;
        //افزودن عنوان ستون ها به سطر دوم فایل
        s = listView1.Columns[0].Text;
        for (i = 1; i < n; i++)
            s += "\t" + listView1.Columns[i].Text;
        writer.WriteLine(s);
        //افزودن داده های مربوط به لیست به سطرهای سوم به بعد
        foreach (ListViewItem item in this.listView1.Items)
        {
           s = item.SubItems[0].Text + "\t" +
               item.SubItems[1].Text+ "\t" +
               item.SubItems[2].Text + "\t" +
               item.SubItems[3].Text + "\t" +
               item.SubItems[4].Text +"\t" +
               item.SubItems[5].Text + "\t" +
               item.SubItems[6].Text;
           writer.WriteLine(s);
        }
        writer.Close();
    }
    MessageBox.Show("داده ها در فایل ذخیره شد");
}
```

لود کردن داده های لیست از داده های موجود در فایل

```
private void btnLoadFromFile_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.Cancel) return;
    if (openFileDialog1.FileName.IndexOf(".txt") < 1)
       openFileDialog1.FileName += ".txt";
    if (openFileDialog1.FileName.Length > 4)
       OpenDataFile();//فراخوانی روال خواندن داده ها از فایل
}


private void OpenDataFile()
{
    try
    {
        using (StreamReader reader = new
             StreamReader(openFileDialog1.FileName,Encoding.UTF8))
    {
        string str = "";
        str=reader.ReadLine();
        if (!str.Split(new char[] { '\t' })[0].Equals("Factor"))
        {
              reader.Close();
              MessageBox.Show("فایل داده ای درست انتخاب نشده است");
              return;
        }
        reader.ReadLine();//skip header line;
        while ((str = reader.ReadLine()) != null)
        {
           string[] strArray = str.Split(new char[] { '\t' });
           ListViewItem item = new ListViewItem();
```

```
            item.Text = strArray[0];
            item.SubItems.Add(strArray[1]);
            item.SubItems.Add(strArray[2]);
            item.SubItems.Add(strArray[3]);
            item.SubItems.Add(strArray[4]);
            item.SubItems.Add(strArray[5]);
            item.SubItems.Add(strArray[6]);
            this.listView1.Items.Add(item);
        }
        reader.Close();
    }
  }
  catch
  {
      MessageBox.Show("خطا در خواندن از فایل داده ای");
  }
}
```

متد کنترل غیر تکراری بودن شماره تولید

```
private bool FindItem(string str)
{
    foreach (ListViewItem item in this.listView1.Items)
    {
       if (item.SubItems[3].Text.Equals(str))
       {
          return true;
       }
    }
    return false;
}
```

افزودن داده ها به لیست در صورت فشردن کلید ENTER در کنترل شماره تولید(بارکد محصول)

```
private void txtBarcode_KeyDown(object sender, KeyEventArgs e)
{
   if (e.KeyCode == Keys.Return)
   {
       AddDataToList(); // فراخوانی متد افزودن داده ها به لیست
   }
}
```

متد افزودن داده به لیست

```
private void AddDataToList()
{
     try
     {
        //کنترل غیر تکراری بودن شماره تولید
        if (this.FindItem(this.txtBarcode.Text))
        {
           MessageBox.Show("شماره تولید تکراری است . لطفا مجددا سعی کنید");
           this.txtBarcode.Text = string.Empty;
```

```
            this.txtBarcode.Focus();
        }
        else
        {
          //افزودن داده ها به لیست
          ListViewItem item = new ListViewItem();
          item.Text = this.txtTarikh.Text;
          item.SubItems.Add(this.txtShomFactor.Text);
          item.SubItems.Add(this.txtCodeMosht.Text);
          item.SubItems.Add(this.txtBarcode.Text);
          item.SubItems.Add(this.txtArzesh.Text);

          item.SubItems.Add(this.cmbVahedArz.SelectedIndex.ToString());
          item.SubItems.Add(this.txtCodeJ.Text);
          this.listView1.Items.Insert(0, item);
          this.txtBarcode.Text = string.Empty;
          this.txtBarcode.Focus();
        }
     }
     catch
     {
        this.txtBarcode.Text = string.Empty;
        this.txtBarcode.Focus();
     }
}
```

رویداد مربوط به حذف داده ها در صورت فشردن کلید حذف رکورد

```
private void btnDeleteRecord_Click(object sender, EventArgs e)
{
   int i = 0;
   foreach (int p in this.listView1.SelectedIndices)
   {
       this.listView1.Items.RemoveAt(p - i);
       i++;
   }
}
```

رویداد مربوط به فشردن کلید خروج

```
private void btnClose_Click(object sender, EventArgs e)
{
   this.Close();
}
```

تمرین :

با توجه به مطالبی که تا کنون آموخته اید

برنامه ای بنویسید که اطلاعاتی شامل :

کد ملی ، نام ، نام خانوادگی ، نام پدر ، شماره شناسنامه ،آدرس ، شماره تلفن ، شماره موبایل و آدرس ایمیل افراد را گرفته و در یک listView نمایش دهد.

در صورتی که کاربر بر روی دکمه ثبت در فایل کلیک کند داده ها در فایل مورد نظر ذخیره شود.

در صورتی که کاربر بر روی دکمه خواندن از فایل کلیک کند داده ها از فایل مورد نظر بازیابی و در لیست نمایش داده شود.

امکان جستجو ، ویرایش و حذف رکورد از لیست وجود داشته باشد

داده ها نباید به صورت تکراری در لیست درج شود. ملاک تکراری نبودن کنترل کد ملی افراد است

در موقع ورود اطلاعات کنترل اعتبار کد ملی افراد باید انجام شود و از پذیرش کد های نامعتبر خود داری گردد

در #C دو دسته کلاس برای ارتباط با بانک اطلاعاتی در نظر گرفته شده است ، یک دسته برای ارتباط با بانک اطلاعاتی SQL و دسته دیگر برای ارتباط با OLEDB

هدف مایکروسافت از این جداسازی بازنویسی کد های ارتباط با SQL برای بهبود سرعت کار با این بانک اطلاعاتی بوده است

برای ارتباط با بانک اطلاعاتی احتیاج به اشیاء زیر داریم

یک شی ارتباطی

یک شی فراهم کننده داده

یک شی برای اجرای دستورات بر روی بانک اطلاعاتی

در جدول زیر نام این اشیاء با توجه به نوع بانک اطلاعاتی و کاربرد آنها لیست شده است

شی	SQL Server	OLEDB	کاربرد
ارتباط	SqlConnection	OleDbConnection	ارتباط با بانک اطلاعاتی
اجرای دستور	SqlCommand	OleDbCommand	اجرای دستور /دستورات روی بانک اطلاعاتی
فراهم کننده داده	SqlDataAdapter	OleDbDataAdapter	انجام عملیات روی بانک اطلاعاتی
فراهم کننده داده خواندنی	SqlDataReader	OleDbDataReader	دسترسی خواندنی به داده بانک اطلاعاتی

کلاس های کار روی SQLServer و OLEDB در فضای نام حداگانه ای قرار دارند و برای استفاده از هر کد ام از آنها باید با استفاده از using فضای نام را به برنامه معرفی کنیم

برای معرفی کلاس های کار روی SQLServer از سرآیند

```
using System.Data.SqlClient;
```

و برای معرفی کلاس های کار با OLEDB از سرآیند

```
using System.Data.OleDb;
```

در ابتدای برنامه استفاده می کنیم

بعد از برقراری ارتباط با بانک اطلاعاتی و دسترسی به دیتا سایر اشیاء دیگر نیازی به درگیر شدن با پیچیدگی های ارتباط و نوع ذخیره سازی و ... در بانک اطلاعاتی ندارند و صرفا روی دیتای آماده شده به شکل استاندارد (جدول) می کنند.

سایر اشیاء کار روی بانک اطلاعاتی عبارتند از

شی	کاربرد
DataSet	حاوی یک یا چند DataTable می باشد
DataTable	دسترسی به داده ها را به صورت جدول (سطر و ستون) فراهم می کند
DataRelation	بین دو DataTable رابطه برقرار می کند
DataView	نمایی از محتویات فیلتر شده یک DataTable را ارائه می کند
DataGridView	**نمایش اطلاعات یک جدول بر روی فرم برنامه**

برای استفاده از این اشیاء باید سرآیند زیر را به ابتدای برنامه تان اضافه کنید

```
using System.Data;
```

بانک اطلاعاتی در سی شارپ

یک پروژه ویندوزی جدید به نام SqlWinApplication ایجاد کنید.

حالا ابزارهای زیر را به فرم اضافه کنید :

۴ تا Button ، ۲ تا Lable ، ۲ تا Textbox

از قسمت Date در Toolbox یک DataGridView به فرم اضافه کنید بعد خاصیت های Text, Name انها را طبق جدول زیر(جدول ۱-۷) تنظیم کنید و بعد چیدمان انها را مثل شکل ۱-۷ درست کنید.

Name	Text	نام ابزار
btnInsert	درج	**Button1**
btnUpdate	ویرایش	**Button2**
btnDel	حذف	**Button3**
btnSearch	جستجو بر حسب نام	**Button4**
txtID		**Textbox1**
txtName		**Textbox2**
	ID:	**Lable1**
	Name :	**Lable2**
grdTable1		**DataGridView**

جدول ۱-۷

شکل ۱-۷

کد نویسی

روی فرم کلیک راست کنید و از منو گزینه View Code را انتخاب کنید .

حالا در قسمت کد نویسی.قبل از هر کاری Using مربوط به SQL را در ابتدای کدتان قرار دهید :

```
using System.Data.SqlClient;
```

حالا یک تابع به نام Fill() تعریف کنید و کدهای زیر را درآن قرار دهید

```
public void Fill()
{
  SqlConnection ObjConnection =
  new SqlConnection("Data Source=Localhost;Initial Catalog=Sample;
  Integrated Security=True");
  SqlDataAdapter ObjDataAdapter=
  new SqlDataAdapter("Select * From Table1", ObjConnection);
  DataSet ObjDataSet = new DataSet();

  ObjConnection.Open();
  ObjDataAdapter.Fill(ObjDataSet, "Table1");
  ObjConnection.Close();

  grdTable1.AutoGenerateColumns = true;
  grdTable1.DataSource = ObjDataSet;
  grdTable1.DataMember = "Table1";

  ObjDataAdapter = null;
  ObjConnection = null;
}
```

دوباره به قسمت Design برگردید.سپس روی قسمت خالی یا روی نوار آبی رنگ بالای فرم در حال طراحی دوبار کلیک کنید و در رخداد Load فرم کد زیر را بنویسید :

```
private void Form1_Load(object sender, EventArgs e)
{
   Fill();
}
```

برنامه را اجرا کنید، خواهید دید که در DataGridView فیلدهای جدول Table1 نشان داده می شود ولی هیچ داده ای در این جدول وجود ندارد.

اکنون باید کدهای مربوط به درج رکورد جدید را بنویسید.برای اینکار روی دکمه درج دوبار کلیک کنید ودر رخداد کلیک آن کد های زیر را بنویسید :

```
private void btnInsert_Click(object sender, EventArgs e)
{
  SqlConnection ObjConnection = new SqlConnection ("DataSource=Localhost;
  InitialCatalog=Sample;IntegratedSecurity=True");
  SqlCommand ObjCommand=
  new SqlCommand("Insert into Table1(ID,Name) Values(@ID,@Name)",ObjConnection);
  ObjCommand.Parameters.AddWithValue("@ID",txtID.Text);
  ObjCommand.Parameters.AddWithValue("@Name",txtName.Text);
  ObjConnection.Open();
  ObjCommand.ExecuteNonQuery();
  ObjConnection.Close();
  Fill();
}
```

بعد از این کار نوبت به دکمه ویرایش می رسد .کد زیر را در رخداد کلیک مربوط به ان دکمه بنویسید:

```
private void btnUpdate_Click(object sender, EventArgs e)
{
  SqlConnection ObjConnection = new SqlConnection ("DataSource=Localhost;
  InitialCatalog=Sample;IntegratedSecurity=True");
  SqlCommand ObjCommand =
  new SqlCommand("Update Table1 Set Name=@Name Where ID=@ID;", ObjConnection);
  ObjCommand.Parameters.AddWithValue("@ID", txtID.Text);
  ObjCommand.Parameters.AddWithValue("@Name", txtName.Text);
  ObjConnection.Open();
  ObjCommand.ExecuteNonQuery();
  ObjConnection.Close();
  Fill();
}
```

برای رخداد کلیک مربوط به دکمه حذف کد زیر را بنویسید

```
private void btnDel_Click(object sender, EventArgs e)
{
  SqlConnection ObjConnection = new SqlConnection ("DataSource=Localhost;
  InitialCatalog=Sample;IntegratedSecurity=True");
  SqlCommand ObjCommand =
  new SqlCommand("Delete From Table1 Where ID=@ID;", ObjConnection);
  ObjCommand.Parameters.AddWithValue("@ID", txtID.Text);
  ObjConnection.Open();
  ObjCommand.ExecuteNonQuery();
  ObjConnection.Close();
  Fill();
}
```

برای رخداد کلیک مربوط به دکمه جستجو کد زیر را بنویسید

```
private void btnSearch_Click(object sender, EventArgs e)
{
  SqlConnection ObjConnection = new SqlConnection ("DataSource=Localhost;
  InitialCatalog=Sample;IntegratedSecurity=True");
  SqlDataAdapter ObjDataAdapter =
  new SqlDataAdapter("Select * From Table1 where Name=@Name", ObjConnection);
  ObjDataAdapter.SelectCommand.Parameters.AddWithValue("@Name", txtName.Text);
  DataSet ObjDataSet = new DataSet();
  ObjConnection.Open();
  ObjDataAdapter.Fill(ObjDataSet, "Table1");
  ObjConnection.Close();

  grdTable1.AutoGenerateColumns = true;
  grdTable1.DataSource = ObjDataSet;
  grdTable1.DataMember = "Table1";

  ObjDataAdapter = null;
  ObjConnection = null;
}
```

توضیح کد هایی هست که نوشتیم :

قبل از هر توضیحی لازم است تا کمی در مورد **DataGridView** بدانیم .این کامپوننت همانطور که از نامش پیداست برای نمایش داده های یک جدول بکار می رود.جدولی که این کامپوننت داده های ان را نمایش میدهد در این برنامه از شی **Dateset** بدست می اید.حال چنانچه توضیح کدهای مربوط به تابع **Fill** را بخوانید متوجه خواهید شد که چطور از این کامپوننت استفاده می کنیم.

```
SqlConnection ObjConnection = new SqlConnection ("DataSource=Localhost;
InitialCatalog=Sample;IntegratedSecurity=True");

SqlDataAdapter ObjDataAdapter=
new SqlDataAdapter("Select * From Table1", ObjConnection);
DataSet ObjDataSet = new DataSet();

ObjConnection.Open();
ObjDataAdapter.Fill(ObjDataSet, "Table1");
ObjConnection.Close();
```

اولین کدی که نوشته شد کد تابع **Fill** بود.در این کد ابتدا یک شی از کلاس **SqlConnection** و سپس یک شی از کلاس **SqlDataAdapter** وپس از آن یک شی از کلاس **Dateset** ایجاد کردیم . در نهایت با استفاده از متد **Fill**،شی **Dataset** را با جدولی به نام **Table1** پر کردیم.

```
grdTable1.AutoGenerateColumns = true;
grdTable1.DataSource = ObjDataSet;
grdTable1.DataMember = "Table1";
```

بعد از آن خاصیت **AutoGenerateColumns** مربوط به شی **DataGridView** را **True** کردیم تا سطرها و ستون ها در این کامپوننت به صورت خودکار ایجاد شود و سپس منبع داده ای را هم برای این شی مشخص کردیم.

```
ObjDataAdapter = null;
ObjConnection = null;
```

بعد هم منابع را آزاد کردیم.

کد نوشته شده برای دکمه های درج ، ویرایش وحذف همه یکی هستند وفقط قسمت دستورات **Sql** آن با هم فرق دارد.

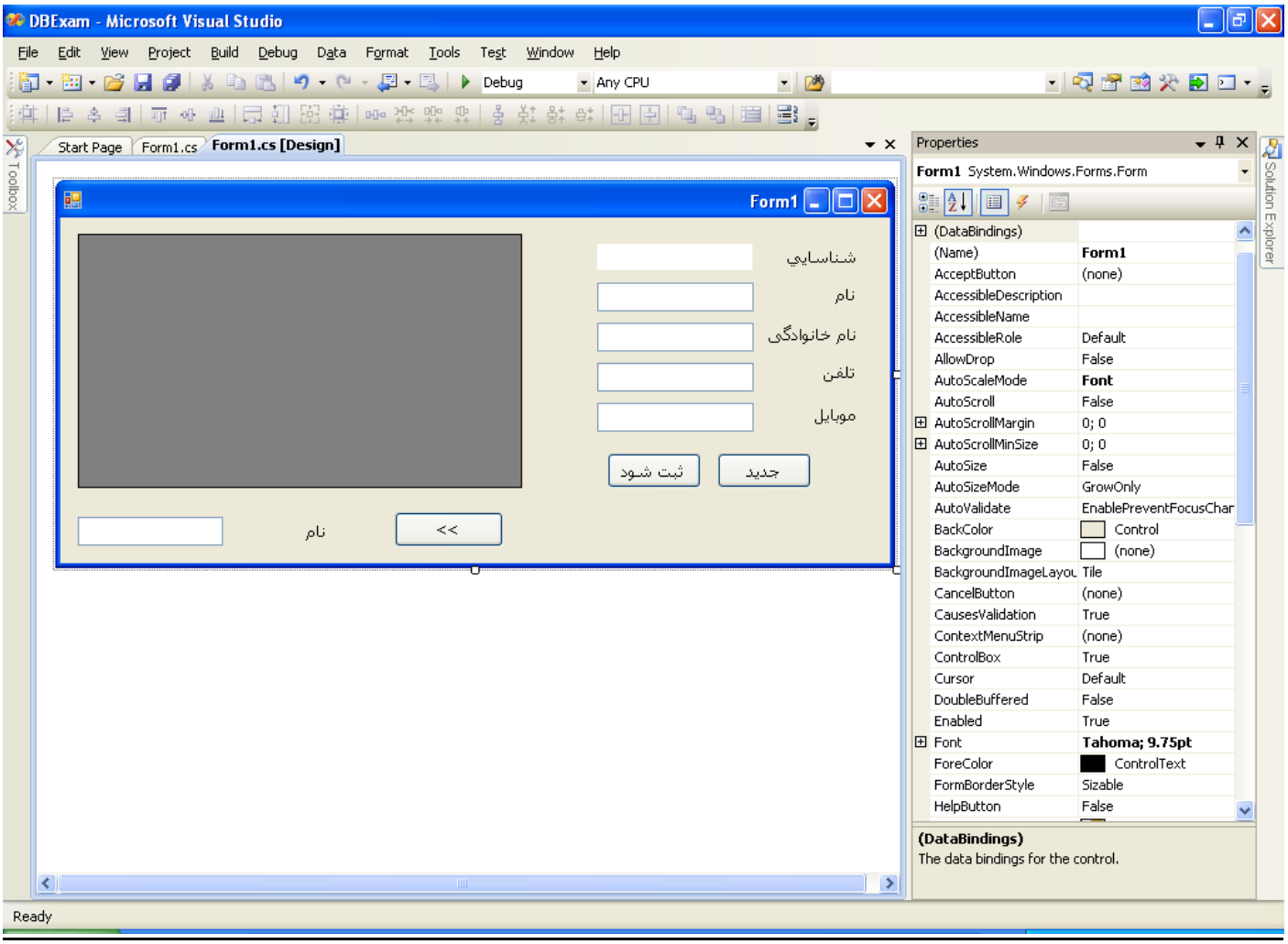
```
SqlConnection ObjConnection = new SqlConnection ("DataSource=Localhost;
InitialCatalog=Sample;IntegratedSecurity=True");
SqlCommand ObjCommand=
new SqlCommand("Insert into Table1(ID,Name)Values(@ID,@Name)",ObjConnection);
ObjCommand.Parameters.AddWithValue("@ID",txtID.Text);
ObjCommand.Parameters.AddWithValue("@Name",txtName.Text);
ObjConnection.Open();
ObjCommand.ExecuteNonQuery();
ObjConnection.Close();
Fill();
```

باز هم در این کد ابتدا یک شی از کلاس **SqlConnection** بعد یک شی از کلاس **SqlCommand** ایجاد کردیم.سپس بامتد **ExecuteNonQuery()** دستورات را اجرا کرده و با فراخوانی متد **Fill()** که تعریف کرده بودیم **DataGridView** را به بروزرسانی کردیم.

کد جستجو بر حسب نام عینا همان کد مربوط به تابع **Fill()** هست تنها تفاوت در قسمت دستورات **Sql** می باشد که یک شرط اضافه شده که یک پارامتر دارد و ان پارامتر با استفاده از تابع **AddWithValue** از طریق **Textbox** ، **txtName.Text** به ان داده شده است.

```
SqlDataAdapter ObjDataAdapter =
new SqlDataAdapter("Select * From Table1 where Name=@Name", ObjConnection);
ObjDataAdapter.SelectCommand.Parameters.AddWithValue("@Name", txtName.Text);
```

نمونه دوم کار با بانک اطلاعاتی



یک پروژه ویندوزی جدید به نام **SqlWinApplication** ایجاد کنید.

حال ابزارهای زیر را به فرم اضافه کنید :

۳ تا **Button** ، ۷ تا **Lable** ، ۴ تا **Textbox**

از قسمت **Date** در **Toolbox** یک **DataGridView** به فرم اضافه کنید بعد خاصیت های **Text**, **Name** انها را طبق جدول زیر تنظیم کنید و بعد چیدمان انها را مثل شکل فوق درست کنید.

نام ابزار	Text	Name			
Button1	جدید	btnNew	**Lable**	شماره شناسایی	
Button2	ثبت شود	btnSabt	**Lable**	نام	
Button3	<<	btnNext	**Lable**	نام خانوادگی	
textbox		txtName	**Lable**	تلفن	
Textbox		txtFamily	**Lable**	موبایل	
Textbox		txtTel	**Lable**		lblID
Textbox		txtMobile	**Lable**	نام	
textbox		db_txtName	**DataGridView**		dataGridView1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace DBExam
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
//خالی کردن فیلد های ورودی
        private void ClearData()
        {
            lblID.Text = "";
            txtName.Text="";
            txtFamily.Text = "";
            txtTel.Text = "";
            txtMobile.Text = "";

        }
//نمایش داده ها در فیلد های روی صفحه نمایش
     private void GetData(int row )
        {
            lblID.Text = dataGridView1[0, row].Value.ToString()  ;
            txtName.Text = dataGridView1[1, row].Value.ToString();
            txtFamily.Text = dataGridView1[2, row].Value.ToString();
            txtTel.Text = dataGridView1[3, row].Value.ToString();
            txtMobile.Text = dataGridView1[4, row].Value.ToString();

        }
//انتقال داده ها از ورودی به بانک اطلاعاتی
        private void SetData()
        {
            String SqlStr = "Insert Into PhoneBook(Name,Famil,Tel,Mobile) ";
            SqlStr+=" Values ('"+ txtName.Text+"','"+txtFamily.Text+"'";
            SqlStr+=",'"+ txtTel.Text+ "','"+txtMobile.Text +"')";
            cnn.Open();
            SqlCommand cmd=new SqlCommand(SqlStr,cnn);
            cmd.ExecuteNonQuery();
            //dataGridView1.
            ds.Clear();
            da.Fill(ds,"PhoneBook");
            cnn.Close();
        }
//تعریف متغیر های عمومی برنامه
        SqlConnection cnn;
        SqlDataAdapter da;
        DataSet ds;

//مقدار دهی متغیر ها در زمان لود فرم
        private void Form1_Load(object sender, EventArgs e)
        {
```

```
            string cnnStr = "Persist Security Info=True;User ID=sa; Password=1008; Initial
Catalog=Test;Data Source=.";
            cnn = new SqlConnection(cnnStr);
            cnn.Open();
            da = new SqlDataAdapter("select * from PhoneBook", cnn);
            ds=new DataSet();
            da.Fill(ds, "PhoneBook");
            dataGridView1.DataSource = ds;
            dataGridView1.DataMember = "PhoneBook";
            db_txtName.DataBindings.Add(new Binding("Text", ds, "PhoneBook.Name"));
            dataGridView1.DataBindings.Add(new Binding("DataSource", ds, "PhoneBook"));
            MessageBox.Show(cnn.ToString());
            cnn.Close();

        }

        private void btnNext_Click(object sender, EventArgs e)
        {
            this.BindingContext[ds, "PhoneBook"].Position++;

        }
//ثبت داده ها در صورت فشردن دکمه ثبت
        private void btnSabt_Click(object sender, EventArgs e)
        {
            SetData();
        }
//خالی کردن مقادیر فیلد ها در صورت فشردن دکمه جدید
        private void btnNew_Click(object sender, EventArgs e)
        {
            ClearData();
        }


        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
        {

            //int i=e.RowIndex;
            //GetData(i);
        }

// نمایش داده ها در صورت حرکت روی جدول نمایش داده ها
      private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
        {
            int i = e.RowIndex;
            GetData(i);

        }


    }
}
```