جامعة حماه

المعهد التقاني للحاسوب

السنة الأولى

برمجة ٢

محاضرة5

قسم البرمجيات

عملي

إعداد:

م. رفا البنات

م. أريج فياض

٧. الصفوف (الكلاسات):

تقدم لغة #c للمبرمج وسائل تعريف أنماط جديدة، مع إمكانية دمج هذه الأنماط أفضل في مجموعة الأنماط المعرفة سابقاً مثل int,string ,bool,..... وهي أنماط معطيات built-in أي مبنية في اللغة فإذا أردنا تخزين رقم فنستخدم int ولكن لو لدينا زبون معين له اسم وعنوان وإيميل وتاريخ ميلاد وله عمليات خاصة به ، فهذا الزبون لن نقدر أن نتعامل معه إلا باعتباره نمط معطيات جديد ولكن ليس كنمط built-in.

إن بناء أنماط جديدة قابلة للتكامل التام مع ما هو موجود في اللغة، هو خطوة نحو بناء برمجيات أفضل ونحو قابلية إعادة استعمال الكود الموجود، باستغلال أنماط المعطيات التي يبنيها المبرمج أثناء عمله.

إن المفهوم الذي يسمح لنا بتنفيذ أنماط المعطيات المجردة في #C هو مفهوم الصف class.

الصف هو نمط يتضمن وصفاً موحداً ل:

- الواصفات أو الخصائصوهي مجموعة من المعطيات التي ترتبط فيما بينها وتكون غالباً من أنماط مختلفة وتسمى("المعطيات الأعضاء ").
- للسلوك وهو مجموعة من الطرائق (وتسمى "التوابع الأعضاء")التي تسمح بالتعامل مع المعطيات الأعضاء .
- يعرف الصف باستخدام الكلمة المفتاحية class ، ويقع جسم الصف بين العلامتين } و {

اسم الصف Class

{

الخصائص

البواني

الطرق

}

مثال ١:

اكتب class لموظف يحمل البيانات التالية : (اسم وعنوان ورقم).

```
class Employee
    {
        public String name;
        public String address;
        public  int  number;

        public Employee()//باني  افتراضي  من دون وسطاء  له  جسم
        {
            name = "zahra";
            address = "hama";
            number = 1;                    }
```

```
public Employee(String nam, int num, String add)//باني ذو وسطاء
      {
          name = nam;
          number = num;
          address = add;
      }
      public void print()// طريقة او تابع لطباعة الاسم
      { Console.WriteLine("hello +",name); }
```

- **الأغراض (objects):**

يعرف الصفُ مجموعةً ممكنةً من الأغراض ، ينتج كلُ غرض عند تنفيذ البرنامج من استنساخ

instantiation صف موجود.

أما الصفُ فهو تمثيلٌ ساكن لمجموعة ممكنة من الأغراض المستنسخة.

لتوضيح الفرق بين الغرض والصف ممكن أن نقول المصنع هو (الصف) الذي ينتج منتجات

(الأغراض). وبإمكان المبرمج تحديد الطريقة التي يجب اتباعها لإنتاج الغرض (الطريقة هي البواني في #c).

لإنشاء غرض من الصف السابق نكتب:

```
Employee  e= new  Employee();
```

ومع أن جميع المعلومات تُخزن ضمن الغرض، فإن ذلك لا يعني أن بالإمكان الوصول إليها جميعها. إذ تقسم معلومات غرض ما إلى قسمين:

- قسم عام **Public**، مرئي ويمكن الوصول إليه من خارج الصف.
- قسم خاص **Private**، غير مرئي ولا يمكن الوصول إليه من خارج الصف.

نستطيع الوصول إلى كل خصائص الصف (class) من خلال اسم الغرض (object)متبوعاً بنقطة (.)

- **البواني (constructors):**

الباني :هوعبارة عن تابع له نفس اسم الصف لكنه لايرجع أي بيانات أي لا يكون له قيمة عائدة return value ولا نضع له void ، ويجب أن يكون نوعه public

تعمل هذه الدالة تلقائياً عندما نأخذ نسخة من ال class يُكتب في هذه الدالة قيم ابتدائية نريد ان نمررها لمتغير ما ومن الممكن أن نمرر له مدخلات أثناء الانشاء مثل الدالة

ويوجد أكثر من نوع للبواني:

(١)الباني الافتراضي (default):

هو باني يُنشأ تلقائياً عند إنشاء الصف ولا داعي لكتابته ويكون بلا وسطاء دوماً و يكون جسمه فارغ .

طريقة تعريف الباني التلقائي:

```
Puplic + ClassName()
{
}
```

(٢)باني بدون وسطاء(non pararmeterized counstructors):

ويكون جسمه يحتوي على قيم ابتدائية للخصائص الموجودة في الكلاس وهي قيم مؤقتة يمكننا تغييرها بمجرد ادخال بيانات جديدة تُخزن بمكانها.

طريقة تعريف الباني الذي لا يأخذ وسطاء :

```
Puplic + ClassName()
{
قيم ابتدائية
}
```

ويكون شكل الباني عند انشاء غرض(object) جديد في ال main.

```
ClassName object=new ClassName ();
```

(٣)باني ذو وسطاء(pararmeterized counstructors):

وهو يأخذ مدخلات مثل الدالة تماماً وعند أخذ نسخة من ال class فيجب تمرير مدخلات له ليتم تنفيذه.

طريقة تعريف الباني الذي يأخذ وسطاء:

```
Puplic + ClassName(parameters)
{
ربط البارمترات مع الخصائص
}
```

ويكون شكل الباني عند انشاء غرض(object) جديد في ال main

```
ClassName object=new ClassName (arguments);
```

الفائدة من الباني بشكل عام: إعطاء قيم للخصائص الموجودة في الصف فلا نحتاج لإعطائها قيم بال main وكما قلنا سابقاً هو المسؤول عن إنتاج غرض أي استنساخ نسخة من الصف.

بالعودة للمثال ١:

```
static public void Main(String[] args)
        {
            employee e = new employee();
            e.name = "sami";
            e.number = 2;
            e.address = "alepo";
            e.print();

        }
    }
}//p
```

```
C:\Windows\system32\cmd.exe
hello sami
Press any key to continue
```

هنا لم نمرر بين قوسين أي وسيط أي هنا قمنا بإنشاء غرض باستخدام الباني الافتراضي بالطريقة الأولى ثم من خلال الغرض e المنشأ من نوع employee قمنا بتعبئة قيم الخصائص للصف واستدعاء دالة Print.

```
static public void Main(String[] args)
        {
            String name = "sami";
            int number = 5;
          String address ="Alepo";
            employee e = new employee(name,number,address);
            e.print();

        }
```

```
C:\Windows\system32\cmd.exe
hello sami
Press any key to continue
```

هنا قمنا بإنشاء الغرض من خلال الباني ذو الوسطاء ،فعند إنشاء الغرض نقوم بتمرير قيم الوسطاء للباني.

```
static public void Main(String[] args)
        {
            String name = Console.ReadLine();
            int number = int.Parse(Console.ReadLine());
          String address =Console.ReadLine();
            employee e = new employee(name,number,address);
            e.print();
```

هنا قمنا بجعل المستخدم يدخل القيم التي يرغب بها .

- **الهادم (destructor):**

هو عكس الباني ، وهو أيضاً عبارة عن دالة تعمل تلقائياً بعد انتهاء البرنامج ، ووظيفته انه يهدم أي كائن لم يُستخدم او غرض تم استخدامه لكن لم يعد له حاجة ، ولا نحتاج إلى استدعائه فهو يعمل فور انتهاء البرنامج دون استدعاء ولا يحتاج الى محددات الوصول ويكون لكل صف هادم واحد فقط.

طريقة التعريف: ~ClassName() { }

مثال ٢:

- **ليكن لدينا طالب له اسم ورقم متسلسل وعلامات لثلاث مواد ، قم بإنشاء صف يحتوي المعطيات السابقة بالإضافة إلى دوال تقوم بحساب المتوسط لتلك العلامات وطباعة النتائج.**

```
class Student {

        public string name;
        public int number;
        public int mark1;
        public int mark2;
        public int mark3;

        public Student()// باني بدون وسطاء فيه قيم ابتدائية
        {
            name = "khaled";
            number = 177;
            mark1 = 65;
            mark2 = 71;
            mark3 = 63;
        }//end of constructor
         public Student(string nam, int num, int m1, int m2, int m3)//باني بوسطاء
         {
              name = nam;
             number= num ;
              mark1= m1;
              mark2= m2;
              mark3= m3;
         }//end of constructor
         public float average()//تابع حساب المتوسط
         {
             float avg = 0;
             avg = (mark1 + mark2 + mark3) / 3;
             return avg;
         }//end of  average method
         public void print()// تابع طباعة الاسم والرقم
         {
             Console.WriteLine("name:" + name);
             Console.WriteLine("number:" + number);
         }//end of print method
    }//end of class
    class Program
    {
        static void Main(string[] args)
        {
           Student s1 = new Student();
            s1.name = "muhammed";
            s1.number = 178;
            s1.mark1 = 78;
            s1.mark2 = 91;
```

```
        s1.mark3 = 75;
        float avg1 = s1.average();
        s1.print();
        Console.WriteLine("average for " + s1.name + " is:" + avg1);
        Console.WriteLine("-------------------------");
        Student s2 = new Student("sara", 179, 89, 75, 63);
        float avg2 = s2.average();
        s2.print();
        Console.WriteLine("average for " + s2.name + " is:" + avg2);
        Console.WriteLine("-------------------------");

        Console.ReadKey();

    }
```

```
name:muhammed
number:178
average for muhammed is:81
-------------------------
name:sara
number:179
average for sara is:75
-------------------------
```

مثال ٣:

ليكن لدينا صف يمثل الوقت : بالساعات والدقائق والثواني والمطلوب كتابة الصف وتمثيل الوقت واستخدام باني وهادم.

```
class Time
  {
      public int hour;
      public int minute;
      public int second;

      public Time(int h,int m,int s)
      {
          hour = h;
          minute = m;
          second = s;
      }
      public Time()//الباني
      {
          hour = 00;
          minute = 00;
          second = 00;
      }
      ~Time()//الهادم
      { Console.WriteLine("time up"); }
  }
  class Program
  {
      static void Main(string[] args)
      {
          Time time1 = new Time();
          time1.hour = 12;
          time1.minute = 23;
          time1.second = 45;
```

```
            Console.WriteLine(time1.hour+ ":" +time1.minute + ":" +time1.second );
            Time time2 = new Time(07, 13, 55);
            Console.WriteLine(time2.hour +":" +time2.minute + ":" +time2.second);
            Console.ReadKey();
        }
```

```
12:23:45
7:13:55
```

مثال ٤:

ليكن لدينا مستطيل له طول وعرض ،والمطلوب كتابة صف يمثل هذا المستطيل وتابع لحساب محيطه وتابع لحساب مساحته.

```
class Recangle  {
        float length;
        float width;
        public Recangle(int l, int w)
        {
            length = l;
            width = w;
        }
        public void area()
        { Console.WriteLine("area=" + length * width); }

        public void periphery()
        { Console.WriteLine("periphery=" + (length + width) / 2); }

    }
    class Program
    {
        static void Main(string[] args)
        {    Console.WriteLine("enter the length: ");
            int length = int.Parse(Console.ReadLine());
            Console.WriteLine("enter the width: ");
            int width = int.Parse(Console.ReadLine());
            Recangle r1 = new Recangle(length,width);
            r1.area();
            r1.periphery();

            Console.ReadKey();
        }
```

```
enter the length:
12
enter the width:
5
area=60
periphery=8.5
```