

مقدمة: تعتمد لغات البرمجة غرضية التوجه على ثلاث أسس وهي.

- الصفوف والأغراض Classes and objects
- الوراثة Inheritance
- تعدد الأشكال Polymorphism.

## الصفوف والأغراض في جافا Classes and objects in Java

```
Class Toto{
    دوال او تعليمات
}
```

**الصف Class:** هو عبارة عن بنية معطيات جديدة أي أنها غير متوفرة في اللغة بمعنى آخر تعريف صف هو تعريف نوع جديد من المعطيات قد تكون خطية أو غير خطية. ونعرف الصف بالكلمة المحجوزة class

**الغرض Object:** هو عبارة عن متحول من صف ما معرف.

```
Toto x;\ \ Toto
```

التصريح عن متحول من صف

لبناء غرض (متحول من صف ما) نستخدم الكلمة المحجوزة new

```
Toto x;\ \ التصريح
\ \ بناء الغرض من الصف
x= new Toto();
```

ان كلمة Toto بعد new هي طريقة Method لبناء الأغراض تكون معرفة داخل الصف واذا لم يتم تعريفها من قبل المبرمج فإن لغة جافا تعرف بانها افتراضي لا يأخذ أي وسطاء.

:Notes

- الباني هو عبارة عن طريقة (دالة خاصة) تحمل اسم الصف نفسه ولا يوجد لها نوع ارجاع أي void.
- وجود عدة اشكال لبناء غرض يقتضي وجود أكثر من باني أي أكثر من طريقة تحمل نفس الاسم لكن بتغيير في الوسطاء المدخلة لدالة الباني أي أنه تحميل زائد.
- بمجرد وجود أي باني في الصف لا تقوم لغة جافا بإنشاء باني افتراضي.

### مكونات الصف:

يمكن ان يحوي الصف على

- معطيات أعضاء Data member: وهي عبارة عن متحولات من أنواع مختلفة
- الدوال الأعضاء Method member: وهي عبارة عن دوال (طرق) معرفة داخل الصف التعامل (معالجة)، ووصول، طباعة...

**Note:** في التطبيقات الشبكية تسمى الدوال الأعضاء برسالة لأنها تنقل معطيات الى خارج الصف

Exa: مثال على التحميل الزائد للباني.

```
class My_new_class{
    int x; double y; string s;
    // الي الاغراض صفات أي الأغراض وسطاء هي السابقة المتحولات تعتبر
    هذه ان أي حدى على غرض بكل خاصة وتكون الصف من انشائها يتم سوف
    دالة طريق عن للإراض تمريرها ويتم غرض كل لأجل خاصة قيم تأخذ المتحولات
    \ * الباني
    افتراضية قيم الوسطاء تأخذ خالي باني// My_new_clas(){};
    افتراضية قيم تأخذ الوسطاء باقي// My_new_class(int n){x=n}
```

```

My_new_class)int x,double y){
    this.x=x;
    this.y=y;
}
My_new_class(int n, double m, string z){
    x=n; y=m; s=z;
}
}
// التالية بالاشكال الرئيسية الدالة يحوي الذي الصف ضمن السابق الصف من أغرض تعريف ويمكن
My_new_class MIS=new My_new_class();

My_new_class mis=new My_new_class(5);

My_new_class mis=new My_new_class(3,5.5);

My_new_class mis=new My_new_class(8,3.1,Cham)
القيم المتحولات تأخذ وعندها
// x=8 وy=3.1 و s=Cham

```

**Exe:** اكتب صف بلغة جافا يوم بتعريف عدد عقدي وعمليات الضرب والجمع لعددتين عقديين ثم استخدم هذا الصف في ادخالي عددتين عقديين وحساب وطباعة ناتج كل من جمعهما وضربهما.

```

class Comp{
    double x ,y;
    Comp() { }
    Comp (double x,double y){
        this.x=x;
        this.y =y;
    }//end of constructor
    //الجمع دالة تعريف
    Comp sum(double x,double y){
        Comp Z=new Comp(x1+x,y1+y);
        return Z;
    }//end of the sum method
    //الوسطاء تغيير مع الجمع لدالة اخر تعرف
    Comp sum(comp c){
        Comp Z=new comp(x+c.x,y+c.y);
        return Z;
    }
    // الضرب دالة تعريف
    Comp multiply(comp c){
        Comp Z= new Comp();
        Z.x=x*c.x-y*c.y;
        Z.y=c.x*y+c.y*x;
        return Z;
    }
}

```

```

class Complex {
public static void main(String[] args) {
double x,y,x1,y1;
x=Stdin.readDouble();
y=Stdin.readDouble();
x1=Stdin.readDouble();
y1=Stdin.readDouble();
Comp c1,c2,c3;
c1=new Comp (x,y);
c2 =new Comp (x1,y1);
c3=new Comp(0,0);
c3=c1.sum(c2);
System.out.println("c3="+c3.x+"+i"+c3.y);
c3=c1.multiply(c2);

System.out.println("c3=c1*c2="+c3.x+"+i"+c3.y);
Comp q1=new Comp(3,4);
Comp q2=new Comp(5, 6);
Comp q3=new Comp();
q3.sum(q1, q2);
System.out.println("q3=q1+q2="+q3.x+"+i"+q3.y);

}\\end of main method
}\\end of class

```

**Note:** لتقريب مفهوم الصفوف و الاغراض الصف يكون بنية مجردة تحوي الصفات العامة التي يمكن ان يحويها الغرض مثلا اذا انشأنا صف لتعريف موظف فإن هذا الصف سوف يحوي حقول تمثل صفات من اسم وعمر وراتب وتاريخ التوظيف..... وعند انشاء موظف جديد(غرض من صف موظف) يكون لهذا لموصف الصفات الخاصة به(وسطاء الباني).

[www.syriamath.net](http://www.syriamath.net)

### **Note**

ان استدعاء دالة الجمع يتم عن طريق Object(غرض) من الصف. ففرض  $z$  و  $q$  متحولات من النوع `Comp` أي اغرض من الصف `Comp` فان استدعاء الطريقة الثانية يتم بما يلي:  $z.sum(q)$  وهي تكافئ العبارة  $z = z + q$ .  
اما استدعاء طريقة الجمع الاولى ففرض  $z$  متحول من الصف `Comp` فان استدعاء دالة الجمع يتم  $z.sum(x, y)$  وهي تكافئ العبارة  $z = z + (x + iy)$

يمكن تعرف دالة جمع تأخذ عددين عقدين وتضع ناتج الجمع بعدد ثالث وتكون بالشكل:

```

void sum(Comp z,Comp q){
x=z.x+q.x;
y=z.y+q.y;
}

```

ولها الشكل  $z.sum(x, y)$  وهي تكافئ  $z = x + y$  حيث  $x, y, z$  هي متحولات من الصف `Comp`

