المحاضرة الخامسة

2013\11\12

الطرق(الدوال)في لغة جافا Method in java:

سابقا استخدمنا اسم دالة Functionللدلالة على إجراء ما ،اما في لغة الجافا فإن التعبير المستخدم هو طريقة Method .وتأخذ الشكل

```
return-value-type  method-name(  parameter-list  )
{
 declarations and statements
}
```

مثال على ذلك يكمن تعريف الطريقة

```
Int fac(int){
تعليمات
}
```

ويسمى اسم الدالة مع وسطائها بالترويسة(أو محدد الدالة) أي أن fac(int) هي ترويسة(محدد) الطريقة السابقة.

التحميل الزائد للطرق Method Over Loading

يمكن أن نعرف أكثر من طريقة في الصف على أن نغير في محددات هذه الطرق ونسمي تحميل زائد لطريقة ما اذا وجد اكثر من طريقة(دالة)لها ننفس الاسم لكن هناك تغيير في قائمة الوسطاء،وتغيير الوسطاء يكون اما تغيير عدد الوسطاء أو نوعها أو ترتيبها اذا كانت من أنواع مختلفة.

Note:لا يمكن تعريف أكثر من دالة لها نفس المحددات مع تغير فقط في القيمة المرجعة.اي لا يوجد تحميل زائد للطريقة على مستوى القيمة المرجعة أنما التحميل الزائد يكون في تغيير الوسطاء فقط.

```
Int fac(int n);
Double fac(int n);
```

هذه الكتابة خاطئة لأنه لا يوجد تحميل زائد على مستوى القيمة المرجعة

أمثلة على التحميل الزائد للطرق:

```
Int toto(int x,int y);
int toto();//يسمى الباني الخالي ونفسه الباني الإفتراضي
int(toto(int x);//التغيير في عدد الوسطاء
int toto(int x,double y);//التغير في نوع الوسيط الثاني
int toto(double x ,double y);//التغيير في نوع الوسيطين
int toto(int y ,int x);//مطابقة للطريقة في السطر الأول
int toto(int x ,int y ,double z);//التغيير في عدد الوسطاء
```

Note:لا يمكن تعريف دالة ضمن دالة ومن الأخطاء الشائعة تعريف دالة ضمن الدالة الرئيسية main

وإنما يمكن استدعاء(استخدام)دالة ضمن دالة أخرى.

Exa:اكتب برنامج بلغة جافا (باستخدام الطرق)يقوم بحساب المقدار $Z = \frac{(x!+y!)}{(x-y)!}$ حيث x, y عددين صحيحين موجبين.

E-mail: imran.shamia@gmail.com

```
class Factor {
public static  long fact(int n){
       if (n==0)return 1;
       else return(n*fact(n-1));
   }
    public static void main(String[] args) {
       int z,y;
       do{
           System.out.println("type the first number");
           z=Stdin.readInt();
       }while(z<0);
       do{
           System.out.println("type the second number");
           y=Stdin.readInt();
       }while(y<0||y>z);
       double m;
       m=(fact(z)+fact(y))/(fact(z-y));
       System.out.println("m="+m);
       }//end of main
   }end of class
```

:Note

تم إضافة شرط أخر على المتحول y<=zكي لا يحصل خطأ أثناء التنفيذ بتمرير قيمة سالبة لدالة العاملي.

العودية المنحنية(غير المباشرة):

في المثال المجاور نلاحظ أن الدالة g تستدعي الدالة fوكذلك الدالة f تستدعي الدالةg ضمن تعليماتها و هذا م يسمى بالعودية المنحنية.

```
Int g(…){
          (تعليمات ما) int y=f(…);
}//end of method g
int f(…){
          (تعليمات ما) int z=g(…);
}//end of methd f
```


E-mail: imran.shamia@gmail.com

Exa:تعريف دالة حساب البعد بين نقطتين.

```
Double dist(int A[],intB[]){
double z;
z=power((A[0]-B[0])*(A[0]-B[0])+ (A[0]-
B[1])*(A[1]-B[1])+ (A[2]-B[2])*(A[2]-B[2]),0.5);

Return z;

}//end of method.
```

تستدعى الدالة dist بعد تمرير متجهتين لها تمثل كل منها نقطة في الفراغ،

E-mail: imran.shamia@gmail.com