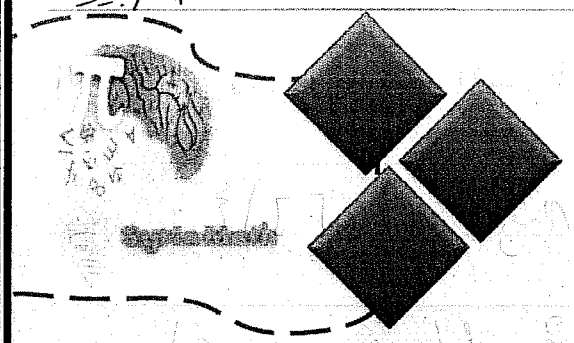

المحاضرة 8

نظري ☐
عملي ☑

◄ دكتور المادة: سمير جعفر

◄ عنوان المحاضرة: برامج بلغة جافا
+ حل أسئلة المذاكرة الثانية

برنامج جلسة العملي / فئة يوم الأحد :

اكتب صنفاً يعرف متجهة من النوع الخاص يحوي دالة طباعة مجموع عناصرها بالشكل العمودي، ثم استخدمه في برنامج رئيسي يدخل متجهة أعداد حقيقية مدخلة البعد ويحسب المتوسط الحسابي لعناصرها ويطبعه.

```
class Array{
  private double A[];
  Array(){ }
  Array(int n){ A= new double[n]; }

  public void input(){                    دالة إدخال عناصر المتجهة
    for(int i=0 ; i< A.length ; i++)
      A[i] = Stdin.readDouble(); }

  public double sum(int x){
    if(x==0) return(A[0]);
    else{ double m;                       // ترجع عنصر المتجهة ذو الدليل x
      m = A[x] + sum(x-1);                مضافاً إلى استدعاء الدالة
      return m; } }                       بالنسبة للعنصر الذي يسبقه //
}
```

دالة حساب المجموع تأخذ وسيطاً يمثل دليل عنصر في المتجهة، وشرط توقف هذه الدالة هو أن يكون الدليل = 0، عندها ترجع أول عنصر في المتجهة (لأنه لا يوجد عناصر قبله)

البرنامج الذي سنستخدم فيه الصف السابق :

```
class UseArray{
public static void main(String args[]){
  int n;
  do{ n=Stdin.readInt(); } while(n<=2);
  Array A = new Array(n);
  A.input();

  double s = (A.sum(n-1))/n;        ← المتوسط الحسابي
  System.out.print(s); }
}
```

في تعليمة حساب المتوسط الحسابي استعملنا دالة المجموع بدءاً من (n-1) لأنه دليل آخر عنصر في المتجهة ، ومستناً على n وهو عدد عناصر المتجهة.

برنامج جلسة العمل / فئة يوم الأربعاء :

أكتب صفاً يمثل مصفوفة من النوع الحاصل بالإضافة إلى بواني ودوال إخراج وإدخال مناسبة، ويحوي دالة طباعة جداء عناصر كل من المصفوفة التي مررناها ، وأخرى لإيجاد مجموع جداءات أعمدة المصفوفة ، واستخدمه في برنامج يدخل مصفوفة حيث يكون عدد أسطرها ضعف عدد أعمدتها ويطبعها ويطبع مجموع جداءات أعمدتها.

```
class Mat{
  private int[][] A;
  Mat(){}                          ← باني خالي
  Mat(int n, int m){ A = new int[n][m]; }
                                   ↑ باني يعطي المتجهة أبعادها
```

```
public void input(){
  for(int i=0; i<A.length; i++)
  for(int j=0; j<A[0].length; j++)
    A[i][j] = Stdin.readInt(); }

public void print(){
  for(int i=0; i<A.length; i++){
  for(int j=0; j<A[0].length; j++)
    System.out.print(A[i][j] + " ");
    System.out.println(); }
}

int mult(int n){ //الوسيط المرر للدالة يمثل العمود الذي ستحسب جداء عناصره
  int s=1;
  for(int i=0; i<A.length; i++)   //حلقة for تمر على الأسطر
    s*=A[i][n];
  return s; }

int sum(){ //دالة حساب مجموع جداءات عناصر الأعمدة.
  int s=0;
  for(int i=0; i<A[0].length; i++)   //حلقة for تمر على الأعمدة
    s+=mult(i); (*)
  return s; }
}
```

(*) يتم استدعاء الدالة السابقة لكل عمود من الأعمدة، فتحسب جداء عناصره ويضاف الناتج إلى s.

```
class UseMat{
public static void main(String args[]){
  int n;
  do{n= Stdin.readInt();} while (n<1);

  Mat obj = new Mat(2*n , n); <-((الأسطر ضعف الأعمدة))
  obj.input();
  obj.print();
  System.out.print(obj.sum());
                                  }}
```

برامج المذاكرة الثانية:

النموذج الأول:

اكتب صفاً يعرف صفوفة أعداد حقيقية من النوع الخاص بالإضافة إلى بواني ودوال إدخال وإخراج مناسبة ودالة تحسب جداء عناصر كل سطر على حدى ثم تجمع ناتج جداء عناصر كل سطر وترجع الناتج، ثم استخدم الصف السابق في برنامج يدخل صفوفة مربعة ويطبعها على شكل صفوفة بالإضافة لطباعة مجموع جداءات أسطرها.

مثال توضيحي لعمل الآلة:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \rightarrow 1\times2\times3 + 4\times5\times6 + 7\times8\times9$$

```
public class A{
  private double[][] M;
A(int n, int m){ M = new double[n][m];}
```

```
void input(){                    //دالة لإدخال عناصر المصفوفة
 for(int i=0 ;i< M.length ;i++)
  for(int j=0 ;j< M[0].length ;j++)
    M[i][j]= Stdin.readDouble(); }

void print(){                    //دالة لطباعة المصفوفة بالشكل المصفوفي
  for(int i=0 ;i<M.length ;i++){
  for(int j=0 ;j< M[0].length ;j++)
    System.out.print(M[i][j] + " ");
    System.out.println(); }
                }
double sum(){
  double r=0;                    //حيادي الجمع
  for(int i=0 ;i< M.length ;i++){
   double t=1;                   //حيادي الضرب
  for(int j=0 ;j< M[0].length ;j++)
   t*= M[i][j];                  //هداء عناصر كل سطر
  r+= t; }                       //ثم نجمع ناتج الضرب
  return r; }                    //وهكذا مع آخر سطر، ثم نعيد ناتج الجمع
        }
```

استخدام الصف السابق في برنامج:

```
class UseA{
public static void main(String args[]){
 int n;
 do{ n=Stdin.readInt();} while(n<1);
```

A a = new A(n,n); //بناء غرض من الصف السابق يحجز حافظة مربعة
a.input(); //استدعاء دالة الإدخال
a.print(); //استدعاء دالة الطباعة
* System.out.println(a.sum()); }
}

* طباعة الناتج عن استدعاء دالة الجمع من أجل الغرض a .

النموذج الثاني :

اكتب صفاً يعرّف حافظة أعداد حقيقية من النوع الخاص بالإضافة إلى البواني ودوال الإدخال والإخراج المناسبة ، ودالة ترجع متجهة صيغ عناصر i من المتجهة هو مجموع عناصر السطر i من الحافظة ، ثم استخدم الصف السابق في برنامج يدخل حافظة مربعة ويطبعها على شكل حافظة بالإضافة لطباعة متجهة مجموع كل سطر .

مثال توضيحي لعمل الدالة :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \Rightarrow \begin{pmatrix} 1+2+3 \\ 4+5+6 \\ 7+8+9 \end{pmatrix}$$

public class B{
private double[][]M ;
B(int n, int m){ M = new double[n][m] ; }

void input(){
for(int i=0 ; i<M.length ; i++)
for(int j=0 ; j<M[0].length ; j++)
M[i][j] = Stdin.readDouble();
}

```
void print(){
 for(int i=0 ; i<M.length ;i++){
 for(int j=0 ; j<M[0].length ;j++)
  System.out.print(M[i][j] + " ");
  System.out.println(); }
                        }
double[] sum(){                    (الدالة المطلوبة نوع إعادتها متجه)
  double[] r = new double[M.length]; (1)
   for(int i=0 ; i<M.length ; i++)
   for(int j=0 ; j<M[0].length ; j++)
      r[i] += M[i][j] ;  (2)
  return r ; }  (3)
                        }
```

(1) متجه أعداد حقيقية r بعدها هو بعد المصفوفة الأول (عدد أسطر المصفوفة).

(2) تحسب نواتج جمع عناصر كل سطر وتضع الناتج في العنصر i من المتجه على الترتيب.

(3) ترجع المتجه الناتجة.

استخدام الصف السابق في برنامج:

```
class UseB{
public static void main(String args[]){
  int n;
do{n=Stdin.readInt();} while(n<1);
  B b = new B(n,n);
  b.input();
  b.print();
```

```
double[] arr = b.sum();          (1)
for(int i=0; i<arr.length; i++)  (2)
    System.out.print(arr[i] + " ");
}
```

(1) تعريف متجهة وإسناد الناتج عن استدعاء دالة الجمع إليها.
(2) طباعة عناصر المتجهة.

النموذج الثالث:

تأتي الشعاع من إحداثيات لنقطة بدء، و شدة وهي مقدار حقيقي تكون قيمتها موجبة وقد تساوي الصفر، بالإضافة إلى زاوية وهي مقدار صحيح محصور بين -179 و 180.

اكتب صفاً يعرف شعاعاً في الفراغ يحوي فقط البادئ الخاص، حيث يتم إدخال القيم عن طريق دالة إدخال تحوي الشروط السابقة، كما يطلب كتابة دالة طباعة البعد بين نقطة بدء هذا الشعاع ونقطة بدء شعاع آخر ممرر كوسيط للدالة، بالإضافة لدالة أخرى طباعة الفرق بين زاوية هذا الشعاع وزاوية شعاع آخر ممرر كوسيط للدالة، ثم استخدم الصف السابق في برنامج يعرف ويدخل ثلاثة أشعة، ويحسب ويطبع البعد بين نقطة بدء الأول ونقطة بدء الثاني، وحساب الفرق بين زاوية الثاني وزاوية الثالث، ويطبع كلمة (same) في حال كان الفرق مساوياً للصفر وكلمة (before) إذا كان الفرق أصغر من الصفر وكلمة (after) إذا كان الفرق أكبر من الصفر.

```
public class Vector{
    double x,y,z;      // إحداثيات نقطة البدء
    double l;          // الشدة
    int a;             // الزاوية
```

ملاحظة: نقطة البدء هي نقطة في الفراغ ← لها ثلاث إحداثيات.

```
Vector( ){ }                                   ((باني خالي))
void input( ){                                 ((دالة الإدخال))
    x = Stdin.readDouble( );
    y = Stdin.readDouble( );                   ((لمن عليها شروط))
    z = Stdin.readDouble( );
do{ l = Stdin.readDouble( );} while (l < 0);   (1)
do{ a = Stdin.readInt( );} while(a<-179 || a>180);   (2)
}
double distance(Vector v){                     (3)
return (Math.sqrt(Math.pow(x - v.x, 2) +
    Math.pow(y - v.y, 2) + Math.pow(z - v.z, 2)));
}
int diff(Vector v){
    return (a - v.a); }
}
```

(1) شرط إدخال الشدة أن تكون أكبر من الصفر أو تساوي الصفر، لذلك وضعنا حلقة تدخل فيها l طالما أنها أصغر تماماً من الصفر.

(2) a محصورة بين القيمتين 179- و 180، لذلك وضعنا حلقة تدخل فيها طالما أنها > 180 أو < -179.

(3) اعتماداً على قانون حساب البعد بين نقطتين في الفراغ:

$$d = \sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2}$$

- استخدام الصف السابق في برنامج:

```
class UseVector{
public static void main(String args[]){
```

```
Vector V1 = new Vector();        // تعريف وبناء ثلاثة
Vector V2 = new Vector();        // أغراض من نوع الصف
Vector V3 = new Vector();        // السابق.
V1.input();                      // إدخال قيم الأعضاء في كل من الأغراض
V2.input();                      // الثلاثة من خلال استدعاء دالة الإدخال
V3.input();
System.out.print(V1.distance(V2));   // طباعة المسافة بين نقطتين البدء من خلال استدعاء الدالة بالنسبة لأحد الأغراض وتمرير الآخر كوسيط
int n = V2.diff(V3);             // أسند ناتج الفرق بين الزاويتين لمتحول n
if (n==0) System.out.print("same");
else if (n<0) System.out.print("before");
else System.out.print("after");
}}
```

- النموذج الأخير مماثل للنموذج السابق تقريباً مع اختلاف أنه يطلب تعريف نقطة البدء في المستوي، أي أننا نحتاج لتعريف إحداثيين إضافيين.
بالإضافة إلى استدعاء دالة حساب الفرق بين الزاويتين بدالة حساب الفرق بين الشرين.

انتهت المحاضرة