Cours 5

- Tableaux
- Tableaux à 2 dimensions
- Tableaux d'objets

Les **tableaux** ne sont pas ni des objets ni des types simples.

Un tableau se rapproche d'un objet

- Il est manipulé par référence (adresse)
- Il nécessite de *new* pour être défini

Déclaration

char tableau[] *ou* char[] tableau

Création

- En utilisant new

```
int tab[];    tab=new int[5];
char t2[]= new char[4];
```

- implicitement avec l'initialisation

```
int t2[]={1,2,3};
```

•La taille est fixée

•On peut accéder à la taille par le champ
length

•Les indices entre 0 et length-1

•Dépassement d'indice : lancement d'exception
ArrayIndexOutOfBoundsException

•Un objet ou un tableau qui n'est plus référencé
est récupéré par ramasse-miettes (garbage)


```
public class TabAffec
{public static void main(String args[])
       {char t1[]={'b','o','n','j','o','u','r'};
       char t2[]={'h','e','l','l','o'};
       char t3[]={'x','x','x','x'};
       t3=t1;  t1=t2; t2=t3;
       System.out.print("t1 =");
       for (int i=0;i<t1.length;i++) System.out.print (t1[i]);
       System.out.println();
       System.out.print("t2 =");
       for (int i=0;i<t2.length;i++) System.out.print (t2[i]);
       System.out.println();
       System.out.print("t3 =");
       for (int i=0;i<t3.length;i++) System.out.print (t3[i]);
       System.out.println();
       }
}

t1=hello
t2=bonjour
t3=bonjour
```

Les paramètres du programme

```
class Prog23 {
  public static void main (String args[])
    {int i;
    System.out.println("les arguments :");
    for (i=0; i<args.length ; ++i)
      System.out.println(args[i]);}
}
```

Exécution

java Prog23 "Bonjour coucou " a b
les arguments :
Bonjour coucou
a
b


```
class UtilTab
{static double somme(double[] t)
        {double s=0;     int i;
        for (i=0;i <t.length; i++) s+=t[i];    return s;}

static void affiche (double[] t)
{for (int i=0; i <t.length; i++) System.out.print(t[i]+" ");
System.out.println();}

public static double[] somme (double[] t1, double[] t2)
{       int n=t1.length;
        if (n!=t2.length) return null;
        double [] res= new double[n];
        for (int i=0; i <n; i++) res[i]=t1[i]+t2[i];   return res;}
}
public class TstUtil
{ public static void main(String[] args)
        { double tab1[]={1.5, 2.25, 3.5, 4.0};
        System.out.println("tab1 "); UtilTab.affiche(tab1);
        System.out.println("somme" + UtilTab.somme(tab1));
        double tab2[]={2.5, 1.5, 3.5, 4.5};
        System.out.println("tab2 "); UtilTab.affiche(tab2);
        double [] tab3= UtilTab.somme(tab1,tab2);
        System.out.println("tab1+tab2 ");UtilTab.affiche(tab3);}
}
```

Tableaux à deux dimensions

int [][] tab; la référence tab qui pointe sur null
tab=new int [2][3]; un tableau de taille 2 de
tableaux d'entiers de taille 3 initialisés par défaut à 0

tableau[1][1]=5;

tableau[0]=new int[2];


```
import java.io.*;
class Matrice
{   int[][] tab;
    int nb_ligne;
    int nb_col;
    public Matrice()
    {   int i,j;
        nb_ligne=Saisie.lire_entier();
        nb_col=Saisie.lire_entier();
        tab=new int[nb_ligne][nb_col];
        for (i=0;i<nb_ligne;i++)
        {   for(j=0;j<nb_col;j++) tab[i][j]=Saisie.lire_entier();  }
    }

    public void affiche()
    {int i,j;
        for (i=0;i<nb_ligne;i++)
        {for (j=0;j<nb_col;j++)  System.out.print(tab[i][j]+" ");
System.out.println(" ");    }
     }
}
```

```
class TableauB
{public static void main(String[] argv)
  {boolean[] tableau={true,false,true};
System.out.println("Deuxieme element de tableau : "+ tableau[1]); }
}

/*A l'execution, on obtient :
Deuxieme element de tableau : false
*/
```


Tableaux d'objets

```
class Point
{private int x,y;

public Point()
{x=Saisie.lire_entier();   y=Saisie.lire_entier();}

public int getx()  {return x;}

public int gety()  {return y;}

public void affiche()
{System.out.println("x = "+x+" y = "+y);}
}

class Gestion_tab
{ private Point[] tab;
  private int nb_pts_positifs;

public Gestion_tab( int n)  {tab=new Point[n];}

public void ajout_pts()
{for (int i=0; i <tab.length;i++) tab[i]=new Point();}
```

```
public void calcul_pts_positifs()
{int nb=0, i;
for (i=0;i < tab.length;i++)
         if ((tab[i].getx() >0) &&(tab[i].gety()>0)) nb++;
         nb_pts_positifs=nb;
}

public void affiche()
{for (int i=0; i <tab.length;i++) tab[i].affiche();}
}

public class TestGestion
{ public static void main (String[] args)
         {Gestion_tab t=new Gestion_tab(3);
         t.ajout_pts(); t.affiche();
         t.calcul_pts_positifs();}
}
```


Classe Vector

•La taille est dynamique: dès qu'un tableau vectoriel est plein, sa taille est doublée, triplée, etc. automatiquement
•Les cases sont de type Object

•add(object o), add(int i, Object o), addElement(Object o), insertElementAt(Object o, int i) pour ajouter un objet
•remove (Object o), removeElement(Object o) removeElementAt(int i) pour supprimer un objet
•get (int i) et elementAt(int i) retourne l'adresse de l'objet à l'indice i
•indexof(Object o) retourne la position de l'objet
•size()

```
import java.util.*;
public class UseVector1{
public static void main(String args[]){
Vector vect = new Vector(5);
vect.addElement(new Integer(3));
vect.addElement(new String("Bonjour"));
vect.addElement(new Float(3.14));
vect.addElement(new Integer(1));
vect.addElement(new Integer(2));
vect.addElement(new Integer(3));

System.out.println("après add : " + vect + "   et  taille = "
     + vect.size());
System.out.println("capacité du vecteur : " + vect.capacity());
System.out.println(vect);
for(int i=0; i < vect.size(); i++)
   if(vect.elementAt(i) != null)
   System.out.println("vect[" + i + "] : " +
   vect.elementAt(i).getClass().getName() + " --> "
   + vect.elementAt(i));
    else
    System.out.println("vect[" + i + "] est null");
    if(vect.elementAt(1) != null){vect.set(1, new Integer(1000));}
    System.out.println("après set: " + vect);
    vect.remove(0);
    System.out.println("après remove : " + vect + "
    et  taille ="   + vect.size()); }
   }
```

```
apr?s add : [3, Bonjour, 3.14, 1, 2, 3]   et  taille = 6
capacit? du vecteur : 10
[3, Bonjour, 3.14, 1, 2, 3]
vect[0] : java.lang.Integer --> 3
vect[1] : java.lang.String --> Bonjour
vect[2] : java.lang.Float --> 3.14
vect[3] : java.lang.Integer --> 1
vect[4] : java.lang.Integer --> 2
vect[5] : java.lang.Integer --> 3
apr?s set: [3, 1000, 3.14, 1, 2, 3]
apr?s remove : [1000, 3.14, 1, 2, 3]   et  taille = 5
```


Iterator

public boolean hasNext()

public Object next()

public void remove() (enlève le dernier élément retourné)

```
import java.util.*;

class Point
{private int x,y;
public Point(int a, int b)
{x=a;  y=b;}
public int getx()  {return x;}
public int gety()  {return y;}
public void affiche() {System.out.println("x = "+x+" y = "+y);}
}

class Gestion_tab
{ private Vector clt; //private Linkedkist clt;
  private int nb_pts_positifs;
public Gestion_tab( )
{clt= new Vector(); // clt=new linkedList();
nb_pts_positifs=0;}

public void ajoute_pt(int a, int b)
{       Point pp= new Point(a,b);
        clt.add(pp);
}
public void affiche_tous_pts()
{for (int i=0; i <clt.size(); i++)
        {Object obj=clt.get(i);
        Point pp=(Point)obj;
        pp.affiche();}
}
```



```
public void calcul_pts_positifs()
{int nb=0;
Iterator it=clt.iterator();
while (it.hasNext())
        {Object obj=it.next();
        Point pp=(Point) obj;
        if ((pp.getx() >0) &&(pp.gety()>0)) nb++;
        }
        nb_pts_positifs=nb;
        System.out.println("Nombre de points positifs= "+
        nb_pts_positifs);
}
}
public class TestVector
{ public static void main (String[] args)
        {Gestion_tab t=new Gestion_tab();
        int nb_pts=Saisie.lire_entier();
        t.ajoute_pt(2,3);
        t.ajoute_pt(5,5);
         t.affiche_tous_pts();
        t.calcul_pts_positifs();}
}
```