1. Passage d'un tableau en paramètre

Un tableau est passé en paramètre à une méthode par **référence**, c'est-à-dire que seule l'adresse du tableau est effectivement transmise à la méthode. La méthode a accès au tableau reçu en paramètre et peut le **modifier**.

Voici un exemple de programme qui affiche le contenu d'un tableau de 6 nombres, double chacun des éléments du tableau puis réaffiche le tableau en utilisant des méthodes.

```
public class ExempleTableauMethodes
{
   public static void main(String args[])
   {
      int tabNombres[] = {2, 8, 1, 9, 12, 5};

      afficher(tabNombres, "avant");
      calculerDouble(tabNombres);
      afficher(tabNombres, "après");
   }

   static void afficher(int tab[], String moment)
   {
      System.out.println("\n\nContenu du tableau " + moment + "\n");
      for (int i = 0; i < tab.length; i++)
         System.out.println("Élément " + (i + 1) + " vaut " + tab[i]);
   }

   static void calculerDouble(int tab[])
   {
      for (int i = 0; i < tab.length; i++)
         tab[i] *= 2;
   }
}
```

Ici on **modifie** le contenu du tableau même si on ne retourne pas de valeur par le nom de la méthode !

Résultats :

```
Contenu du tableau avant

Élément 1 vaut 2
Élément 2 vaut 8
Élément 3 vaut 1
Élément 4 vaut 9
Élément 5 vaut 12
Élément 6 vaut 5


Contenu du tableau après

Élément 1 vaut 4
Élément 2 vaut 16
Élément 3 vaut 2
Élément 4 vaut 18
Élément 5 vaut 24
Élément 6 vaut 10
```

2. Recherche séquentielle dans un tableau

On a très souvent à effectuer la recherche d'un élément dans un tableau afin de connaître sa position. Pour faciliter la réutilisation du code, on fait cette recherche dans une méthode.

La logique consiste à parcourir le tableau jusqu'à la fin pour vérifier si l'élément est là ou pas. On sort de la boucle aussitôt que l'élément est trouvé. Si l'élément n'est pas trouvé, on retourne la valeur **-1**.

Légende :

i	indice dans le tableau
trouve	pour indiquer si l'élément a été trouvé ou pas
nbEl	nombre d'éléments dans le tableau
posi	position de l'élément dans le tableau si trouvé
tablo	tableau d'éléments
no	élément à chercher dans le tableau

Algorithme :

```
i ← 0
posi ← -1
trouve ← faux
tant que (i < nbEl et pas trouve)
    si tablo[i] = no
        trouve ← vrai
        posi ← i
    sinon
        i ← i +1
    fin si
fin tant que
retourner posi
```

Programme en Java :

```
/*
 * recherche séquentielle dans un tableau d'entiers
 * on cherche l'entier no dans le tableau tablo qui contient nbEl éléments
 * on retourne sa position dans le tableau ou -1 si pas trouvé
 */
static int rechercher(int no, int tablo[], int nbEl)
{
    int posi = -1;
    boolean trouve = false;

    for (int i = 0; i < nbEl && !trouve; i++)
        if (tablo[i] == no)
        {
            trouve = true;
            posi = i;
        }
    return posi;
}
```

Exercice 1 : Écrivez une méthode qui reçoit en paramètre un tableau de nombres réels et qui retourne la plus grande valeur trouvée dans le tableau.

```
static          trouverMax(                         )
{
}  // fin de la méthode trouverMax
```

Exercice 2 : Écrivez une méthode qui reçoit en paramètre un tableau de nombres réels et qui retourne **la position** de la plus petite valeur trouvée dans le tableau.

```
static          trouverMin(                         )
{
}  // fin de la méthode trouverMin
```

Exercice 3 : Écrivez une méthode qui reçoit en paramètres un tableau contenant le code sexe de chaque employé d'une compagnie (le code est 'F' pour une femme et 'H' pour un homme) et un tableau contenant leur âge respectif. La méthode doit retourner le nombre de femmes dont l'âge est inférieur à 30.

```
static        compterJeunesFemmes(                                    )
{
} // fin de la méthode compterJeunesFemmes
```

Exercice 4 : Complétez la méthode *main* et la méthode *rechercher* qui reçoit en paramètres un tableau de nombres entiers, le nombre d'éléments que contient le tableau ainsi que le nombre à chercher dans le tableau. La méthode doit retourner la position du nombre cherché dans le tableau ou -1 si le nombre n'est pas trouvé dans le tableau.

```
import javax.swing.*;
import java.text.*;
public class Exercice5 {

   public static void main(String args[]) {
       final int NB_PROD = 8;
       DecimalFormat cash = new DecimalFormat("0.00 $");
       int    tabNoProd[]   = {  234, 125, 657,  987, 213,  934,  678,  776};
       double tabPrixProd[] = {45.99,9.50,5.75,12.35,9.75,87.45,56.99,76.56};

       int noProd,
           qte,
           posiProd;    // position de noProd dans le tableau tabNoProd
       double cout;
       char reponse;
```

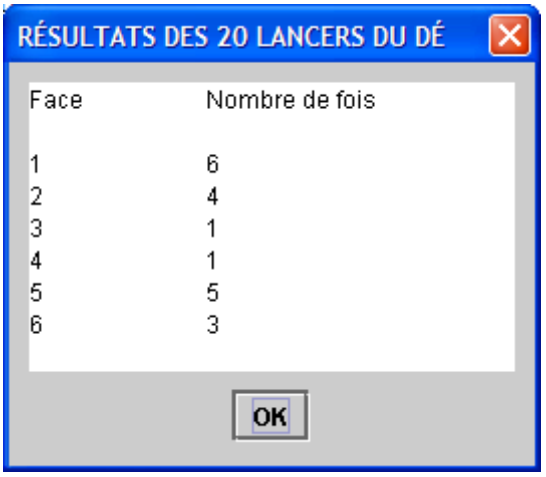
```
        do {
            noProd = Integer.parseInt(JOptionPane.showInputDialog(
                        "Entrez le no du produit à acheter"));
            qte = Integer.parseInt(JOptionPane.showInputDialog(
                        "Entrez la quantité désirée"));

            posiProd =
            if (                          )
            {
                cout =
                JOptionPane.showMessageDialog(null,
                    "Le coût de cet achat est de " + cash.format(cout));
            }
            else
                JOptionPane.showMessageDialog(null, "No de produit erroné");

            reponse = JOptionPane.showInputDialog(
                    "Avez-vous un autre produit à acheter O/N ?").charAt(0);
        } while (Character.toUpperCase(reponse) == 'O');

        System.exit(0);
    } // fin de la méthode main

    static        rechercher(                                              )
    {

    } // fin de la méthode rechercher
} // fin de la classe
```

Exercice 5 : Écrivez une méthode qui reçoit en paramètres un tableau contenant le code sexe de chaque employé d'une compagnie (le code est 'F' pour une femme et 'H' pour un homme) et un tableau contenant leur âge respectif. La méthode doit retourner l'âge de l'homme le plus jeune.

```
static          trouverAgeMinHomme(                                    )
{
}
}  // fin de la méthode trouverAgeMinHomme
```

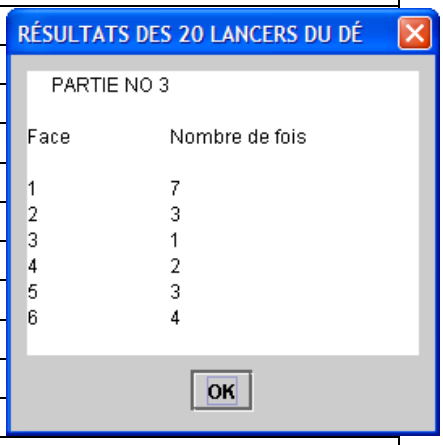
Exercice 6 : Complétez le programme pour résoudre le problème suivant : on lance un dé 20 fois et on désire connaître le nombre de fois que chaque face du dé a été obtenue. Le lancer du dé est simulé à l'aide de la méthode Math.random()

```
import javax.swing.*;
public class Exercice7 {
    public static void main(String args[]) {
        final int NB_LANCERS = 20;
                                                    // tableau de compteurs
        jouerPartie(                                );
        afficherResultats(                              );
        System.exit(0);
    } // fin de la méthode main

    static void jouerPartie(int nbFois[], int nbLancers) {
        int face;
        for (int lancer = 1;                                          )
        {
                    = (int)(Math.random() * 6) + 1;

        }
    }

    static void afficherResultats(int nbFois[], int nbLancers) {
        JTextArea sortie = new JTextArea(5,5);
        sortie.append("Face\tNombre de fois\n\n");
        for (int face = 1; face <= 6; face++)
            sortie.append(                                                    );
        JOptionPane.showMessageDialog(null, sortie,
            "RÉSULTATS DES " + nbLancers +" LANCERS DU DÉ",
            JOptionPane.PLAIN_MESSAGE);
    }
}
```



Exercice 7 : Modifiez le programme de l'exercice 7 en supposant qu'on veut jouer 10 parties de 20 lancers plutôt qu'une seule.

```
import javax.swing.*;
public class Exercice8 {
    public static void main(String args[]) {
        final int NB_LANCERS = 20;
        final int NB_PARTIES = 10;
                                                    // tableau de compteurs

        for (int partie = 1; partie <= NB_PARTIES; partie++) {
            initialiser(           );            // remettre les compteurs à 0
            jouerPartie(                                );
            afficherResultats(                                , partie);
        }
        System.exit(0);
    } // fin de la méthode main

    static void initialiser(int nbFois[]) {
        for (                                          )

    } // fin de la méthode initialiser


    static void jouerPartie(int nbFois[], int nbLancers) {
        int face;
        for (int lancer = 1;                                                ) {
                    = (int)(Math.random() * 6) + 1;

        }
    } // fin de la méthode jouerPartie

    static void afficherResultats(int nbFois[], int nbLancers, int noPartie) {
        JTextArea sortie = new JTextArea();
        sortie.append("PARTIE NO " + noPartie + "\n\n");
        sortie.append("Face\tNombre de fois\n\n");

        for (int face = 1; face <= 6; face++)
            sortie.append(                                                  );
        JOptionPane.showMessageDialog(null, sortie,
            "RÉSULTATS DES " + nbLancers +" LANCERS DU DÉ",
            JOptionPane.PLAIN_MESSAGE);
    } // fin de la méthode afficherResultats
} // fin de la classe
```

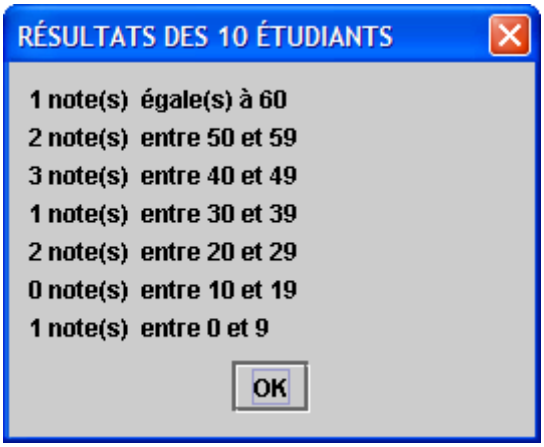


Exercice 8 : Complétez le programme pour résoudre le problème suivant : on lit la note sur 60 (un entier) de 10 étudiants d'une classe en les mémorisant dans un tableau appelé **points**. À partir de ce tableau, on veut établir un tableau appelé **nbNotes** de dimension 7 et qui est composé de la façon suivante :

nbNotes[6] contient le nombre de notes égales à 60
nbNotes[5] contient le nombre de notes de 50 à 59
nbNotes[4] contient le nombre de notes de 40 à 49
....
nbNotes[0] contient le nombre de notes de 0 à 9

```
import javax.swing.*;
public class Exercice9 {
    public static void main(String args[]) {
        final int NB_ETUD = 10;
        int points[] = new int[NB_ETUD];
        int nbNotes[] = new int[7]; // tableau des compteurs des notes
                                    // dans un intervalle

        remplirTableau(                                    );
        compterNbNotes(                                    );
        afficherResultats(                                 );

        System.exit(0);
    } // fin de la méthode main

    static void remplirTableau(int points[], int nbEtud) {
        for (int i = 0;                              )
                        = Integer.parseInt(JOptionPane.showInputDialog(
                          "Entrez la note de l'étudiant " + (i + 1)));
    } // fin de la méthode remplirTableau

    static void compterNbNotes(int points[], int nbNotes[], int nbEtud) {

    } // fin de la méthode compterNbNotes
```

```
    static void afficherResultats(int nbNotes[], int           ) {
        String mots[] = {" entre 0 et 9", " entre 10 et 19",
                         " entre 20 et 29", " entre 30 et 39",
                         " entre 40 et 49", " entre 50 et 59",
                         " égale(s) à 60"};
        String texte = "";

        for (int i = nbNotes.length - 1;                          )
            texte +=               + " note(s)" +            + "\n";

        JOptionPane.showMessageDialog(null, texte,
            "RÉSULTATS DES " +              + " ÉTUDIANTS",
            JOptionPane.PLAIN_MESSAGE);
    } // fin de la méthode afficherResultats
} // fin de la classe
```



Exercice 9 : Complétez le programme suivant qui permet de calculer et d'afficher le coût de l'achat d'un client (il y a plusieurs clients à traiter, mais on suppose qu'un client n'effectue qu'un seul achat) et qui va afficher (après avoir traité tous les clients de la journée) la quantité totale vendue pour chaque produit.

```
import javax.swing.*;
import java.text.*;
public class Exercice10 {

   public static void main(String args[]) {
       final int NB_PROD = 8;
       int    tabNoProd[]   = {  234, 125, 657,  987, 213,  934,  678,  776};
       double tabPrixProd[] = {45.99,9.50,5.75,12.35,9.75,87.45,56.99,76.56};


       traiterLesClients(                                                );
       afficherResultats(                                                );
       System.exit(0);
   } // fin de la méthode main

   static void traiterLesClients(int tabNoProd[], double tabPrix[],
           int tabQteTotale[], int nbProd) {
       DecimalFormat cash = new DecimalFormat("0.00 $");
       int numero,
           qte,
           posiProd;    // position du numéro dans le tableau tabNoProd
       double cout;
       char reponse;

       do {
           numero = Integer.parseInt(JOptionPane.showInputDialog(
                     "Entrez le numéro du produit à acheter :"));
           qte = Integer.parseInt(JOptionPane.showInputDialog(
                     "Entrez la quantité désirée :"));

           if (posiProd != -1)
           {
               cout =
               JOptionPane.showMessageDialog(null,
                   "Le coût de cet achat est de " + cash.format(cout));

           }
           else
               JOptionPane.showMessageDialog(null, "No de produit erroné");

```

```
            reponse = JOptionPane.showInputDialog(
                    "Avez-vous un autre client à traiter O/N ?").charAt(0);
            reponse = Character.toUpperCase(reponse);
        } while (reponse == 'O');
    } // fin de la méthode traiterLesClients

    static int rechercher(int tab[], int nbEl, int valeurCherchee)
    {
        int posi = -1;
        boolean trouve = false;

        for (int i = 0; i < nbEl && !trouve; i++)
            if (tab[i] == valeurCherchee)
            {
                posi = i;
                trouve = true;
            }
        return posi;
    } // fin de la méthode rechercher

    static void afficherResultats(int tabNoProd[], int                    ,
            int nbProd) {
                                                                  RÉSULTATS DE LA JOURNÉE
                                                                  Numéro du produit   Quantité totale
                                                                  234   0
                                                                  125   10
                                                                  657   0
                                                                  987   15
                                                                  213   0
                                                                  934   0
                                                                  678   15
                                                                  776   1
                                                                  OK
    } // fin de la méthode afficherResultats
} // fin de la classe
```