

Les structures de contrôles

L'idée est de s'entraîner avec les structures de contrôles

- [New Page](#)

New Page

Exercice sur les IfThenElse

Exercice IfThen

Considérez la séquence d'instructions suivante:

```
if (A>B) System.out.println ("premier choix \n"); else
if (A>10) System.out.println ("deuxième choix \n");
if (B<10) System.out.println ("troisième choix \n");
else System.out.println ("quatrième choix \n");
```

a) Copiez la séquence d'instructions en utilisant des tabulateurs pour marquer les blocs if - else appartenant ensemble.

b) Déterminez les réponses du programme pour chacun des couples de nombres suivants et vérifiez à l'aide de l'ordinateur.

- A=10 et B=5
- A=5 et B=5
- A=5 et B=10
- A=10 et B=10
- A=20 et B=10
- A=20 et B=20

Exercice IfThenElse

Considérez la séquence d'instructions suivante:

```
if ( A>B)
if ( A>10)
System.out.println ("premier choix \n"); else if ( B<10)
System.out.println ("deuxième choix \n"); else
if ( A==B) System.out.println ("troisième choix \n");
else System.out.println ("quatrième choix \n");
```

a) Copiez la séquence d'instructions en utilisant des tabulateurs pour marquer les blocs if - else appartenant ensemble.

b) Pour quelles valeurs de A et B obtient-on les résultats:
premier choix, deuxième choix, ... sur l'écran?

c) Pour quelles valeurs de A et B n'obtient-on pas de réponse sur l'écran?

d) Notez vos réponses et choisissez vous-mêmes des valeurs pour A et B pour les vérifier l'aide de l'ordinateur.

Exercice maximum

Ecrivez un programme qui lit trois valeurs entières (A, B et C) au clavier et qui affiche la plus grande des trois valeurs, en utilisant:

- if - else et une variable d'aide MAX
- les opérateurs conditionnels et une variable d'aide MAX

Exercice Tri

Ecrivez un programme qui lit trois valeurs entières (A, B et C) au clavier. Triez les valeurs A, B et C par échanges successifs de manière à obtenir :

val(A) val(B) val(C)

Affichez les trois valeurs.

Exercice signe

Ecrivez un programme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe du produit de A et B sans faire la multiplication.

Exercice abs

Ecrivez un programme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe de la somme de A et B sans faire l'addition. Utilisez la fonction abs de la classe Math

== Exercice second degré ==

Ecrivez un programme qui calcule les solutions réelles d'une équation du second degré $ax^2+bx+c = 0$ en discutant la formule:

$X1, X2 = (-b \pm \text{Math.sqrt}(b*b-4ac)) / 2a$

Utilisez une variable d'aide D pour la valeur du discriminant b^2-4ac et décidez à l'aide de D, si l'équation a une, deux ou aucune solution réelle. Utilisez des variables du type int pour A, B et C.

Considérez aussi les cas où l'utilisateur entre des valeurs nulles pour A; pour A et B; pour A, B et C. Affichez les résultats et les messages nécessaires sur l'écran.

Exercice sur les boucles et les tableaux

Boucle et tableau 1

Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Calculer et afficher ensuite la somme des éléments du tableau.

Boucle et tableau 2

Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Boucle et tableau 3

Effacer ensuite toutes les occurrences de la valeur 0 dans le tableau T et compter les éléments restants. Afficher le tableau résultant.

Boucle et tableau 4

Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Ranger ensuite les éléments du tableau T dans l'ordre inverse sans utiliser de tableau d'aide. Afficher le tableau résultant.

Idée: Echanger les éléments du tableau à l'aide de deux indices qui parcourent le tableau en commençant respectivement au début et à la fin du tableau et qui se rencontrent en son milieu.

Boucle et tableau 5

Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau TPOS et toutes les valeurs strictement négatives dans un troisième tableau TNEG. Afficher les tableaux TPOS et TNEG.

Boucle et tableau 6

Ecrire un programme qui lit les dimensions L et C d'un tableau T à deux dimensions du type int (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de tous ses éléments.

Boucle et tableau 7

Ecrire un programme qui lit les dimensions L et C d'un tableau T à deux dimensions du type int (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de chaque ligne et de chaque colonne en n'utilisant qu'une variable d'aide pour la somme.

Boucle et tableau 8

Ecrire un programme qui transfère un tableau M à deux dimensions L et C (dimensions maximales: 10 lignes et 10 colonnes) dans un tableau V à une dimension L*C.

Exemple:

a	b	c
---	---	---

d	e	f
---	---	---

en

a	d
b	e
c	f

Maximum et minimum des valeurs d'un tableau

Ecrire un programme qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers A. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

Insérer une valeur dans un tableau trié

Un tableau A de dimension N+1 contient N valeurs entières triées par ordre croissant; la (N+1)ième valeur est indéfinie. Insérer une valeur VAL donnée au clavier dans le tableau A de manière à obtenir un tableau de N+1 valeurs triées.

```
boolean hasBeenInserted = false;
int[] resultat = new int[arr.length + 1];
int index = 0;
int toBeInserted = 6;
for (int i = 0; i < arr.length; i++) {
    int valeur = arr[i];
    if (valeur >= toBeInserted && !hasBeenInserted) {

        resultat[index] = toBeInserted;
        index++;
        hasBeenInserted = true;
    }
}
resultat[index] = valeur;
```

```
    i++;
```

```
}
```

Exercice sur les boucles

Calcul de Factorielle

Essayer d'écrire le code permettant de calculer l'opération factorielle.

Cette opération prend un entier en paramètre et :

* renvoie 1 si le paramètre est vaut 0

* renvoie $n*(n-1)*(n-2)*\dots*1$. Par exemple $5!=5*4*3*2*1$

```
public class Fact {

    public static void main(String [] args)
    {
        String param=Utils.readLine();
        int nb=Utils.convertStringToInt(param);
        int factorielle=1;
        for (int i = 2; i <= nb; i++)
        {
            factorielle *= i;
        }
        System.out.println(factorielle);
    }
}
```

Dessin

Ecrire une méthode static prenant en paramètre :

* nbTiret un nombre

* nbEspace un nombre

* nbMotif un nombre

L'idée étant d'afficher au plus nb caractère composé de nbTiret de fois le caractère - et nbEspace de fois le caractère espace.

Par exemple pour nbMotif=3, nbTiret=3 et nbEspace=2 on aura la chaîne suivante <--- --- --- >

```
public static void drawLine (int nm, int nt, int nb)
{
    for (int i = 0; i < nm; i++)
    {
        // Tirets
        for (int j = 0; j < nt; j++)
        {
            System.out.print ("-");
        }
        // Espaces
        for (int j = 0; j < nb; j++)
        {
            System.out.print (" ");
        }
    }
    System.out.println();
}
```

Palindrome

Énoncée

Écrire une méthode statique prenant en paramètre :

* str une chaîne de caractère

L'idée étant d'afficher vrai si la chaîne de caractère est un palindrome. Un palindrome étant une chaîne de caractère se lisant de la même façon du début à la fin que de la fin vers le début.

Par exemple radar est un palindrome.

Dans l'exercice suivant :

- les espaces sont ignorées 'rad ar' est un palindrome
- la fonction replace, length et charAt de la classe String peuvent être utilisés

La méthode conseillé est d'utilisé la méthode donnée ci joint qui élimine les espaces d'une chaine de caractère

Solution

```
public static boolean isPalindrome (String s)
{
    s = s.replace(" ", "");
    for (int i = 0; i < s.length() / 2; i++)
    {
        if (s.charAt (i) != s.charAt (s.length() - 1 - i))
        {
            return false;
        }
    }
    return true;
}
```

Enoncée

Maintenant l'idée est d'écrire une methode renvoyant le miroir d'une chaine de caractère. Par exemple le mot 'caractère' a travers cette méthode doit renvoyer 'erètcarac'

```
public static String reverse (String str)
{
    String s = "";
    for (int i = str.length() - 1; i >=0; i--)
    {
        s += str.charAt (i);
    }
    return s;
}
```

Enoncée

Écrivez une méthode de signature

```
public static String merge (String s, String t)
```

qui permet de fusionner les deux chaînes de caractères s et t. Fusionner deux chaînes de caractères s1s2s3... et t1t2t3... consiste à construire une nouvelle chaîne de caractères qui sera construite en prenant une lettre à la fois dans chacune des chaînes. La fusion sera la chaîne s1t1s2t2s3t3...

Faites bien attention que les deux chaînes n'ont pas forcément les mêmes longueurs. Par exemple, l'appel

```
merge ("Hello", "Bonjour");
```

renvoie la chaîne de caractères HBeolnljoour.

On utilisera la méthode static max de la classe Math qui renvoie le maximum de deux entiers.

Solution

```
public static String merge (String s, String t)
{
    String ret = "";
    int max = Math.max (s.length(), t.length());
    for (int i = 0; i < max; i++)
    {
        if (i < s.length()) ret += s.charAt (i);
        if (i < t.length()) ret += t.charAt (i);
    }
    return ret;
}
```

PairImpair

Énoncée

Ecrire une methode static prennant en paramètre un tableau d'entier et imprimant le nombre d'entier impair du tableau.

On utiliseras la propriété que $x \% 2$ vaut zero si x est pair

Solution

```
public static int nbOfOddValues (int[] tab)
{
    int cnt = 0;
    for (int i = 0; i < tab.length; i++)
    {
        if (tab[i] % 2 != 0)
        {
            cnt++;
        }
    }
    return cnt;
}
```

Livret A

Enoncée

Le but est d'écrire une méthode prennant en paramètre :

- une somme d'argent
- un taux de livret A
- un nombre indiquant le nombre d'année de placement

Cette méthode doit calculer la somme d'argent placé au taux sus cité l'argent placé en début d'année:

Solution

```
public static double computeLivretA (double money, double interet, int nbyear)
{
```

```
for (int i=0;i<nbyear;i++) money+=money*interet/100;
return money;
}
```

Enoncée

Le but est d'écrire une méthode renvoyant le nombre d'année qu'il faut pour que de l'argent placé atteigne montant demandé.

Par exemple pour passer de 100€ à 1000€ avec un taux à 10% la méthode doit renvoyer 7.

Solution

```
public static int livretA(double money, double target, double interest)
{
    double balance = 0;
    int years = 0;

    while (balance < target)
    {
        balance += money; // début année, j'ajoute argent sur le compte
        balance += balance * interest / 100; // fin année, j'ajoute intérêts
        years++; // une année écoulée
    }

    return years;
}
```

Pierre Feuille Ciseaux

Enoncée

Dans ce jeux, l'utilisateur choisie un objet parmi une feuille, une pierre et un ciseaux.

L'ordinateur fait deux meme.

Le résultat est que la feuille gagne sur le cailloux, qui gagne sur le ciseaux, qui gagne sur la feuille.

Pour reussir, il faut prendre la methode Math.random pour avoir un nombre aléatoire.

Solution

```

public class PierreFeuilleCiseaux{
public static void main (String args[]) {
String arme[] = {"la pierre", "les ciseaux", "la feuille"};
String message[] = { arme[0] + " casse " + arme[1], arme[2] + " recouvre " + arme[0], arme[1]
+ " coupent "+ arme[2] };
int resultat[] = {0, 2, 1};
// affichage menu
for (int i=0;i<3;i++){
System.out.println(i + ". " + arme[i]);
}
// choix du joueur 1 : l'humain
int j1 = -1;
do{
j1 = Utils.readInt(); // ascii

}while(j1 < 0 || j1 > 2);
int j2 = (int)(Math.random()*3); // choix du joueur 2 : la machine
System.out.println("J' ai choisi " + arme[j2]);
if ( j1 == j2 ) { System.out.println("On a choisi pareil ! " );System.exit(0); }
int i = j1 + j2 - 1;
System.out.println(message[i]);
if (j1 == resultat[i]) {
System.out.println("Tu gagnes" );
} else {
System.out.println("Je gagne" );
}
}
}
}

```

Les images

L'idée est de coder des filtres sur les images (vecteur composé de pixel rouge, vert, bleue et transparence).

Nous allons commencer par une superposition d'image (addition de deux images):

La spécification de la fonction est `static Image add(Image original,Image original2, float alpha1,float alpha2);`

et l'image de destination est `rougeDestination=alpha1*rougeOriginal1+alpha2*rougeOriginal2` (et ceci pour toutes les couleurs)

En sus, sont prévue 3 filtres:

- un passage en niveau de gris d'une image (moyenne des 3 couleurs)
- un passage en niveau de gris de meilleurs qualité ($0.21 * \text{rouge} + 0.71 * \text{vert} + 0.07 * \text{bleu}$)

- un filtrage gaussien (application de la matrice 121).

Pour faire cela, nous allons utiliser 2 classes:

```
public class Pixel {
    public int alpha, red, green, blue;
    public static Pixel createPixel(int alpha,int red,int green,int blue)
    {
        Pixel p=new Pixel();
        p.alpha=alpha;
        p.red=red;
        p.green=green;
        p.blue=blue;
        return p;
    }
}
```

et

```
import java.awt. Color;
import java.awt. image. BufferedImage;
import java. io. File;
import java. io. IOException;

import javax. imageio. ImageIO;

public class Image {

    public int width;
    public int height;
    private BufferedImage data;

    public static Image createImage(Image myimage)
    {
        Image result= new Image();
        result.height=myimage.height;
        result.width=myimage.width;
        result.data=new BufferedImage(myimage.data.getWidth(), myimage.data.getHeight(),
        myimage.data.getType());
        return result;
    }
}
```

```

public static Pixel getPixel(Image myimage,int i,int j)
{
Pixel pixel=new Pixel();
pixel.alpha = new Color(myimage.data.getRGB(i, j)).getAlpha();
pixel.red = new Color(myimage.data.getRGB(i, j)).getRed();
pixel.green = new Color(myimage.data.getRGB(i, j)).getGreen();
pixel.blue = new Color(myimage.data.getRGB(i, j)).getBlue();
return pixel;
}
private static int colorToRGB(Pixel p) {

int newPixel = 0;
newPixel += p.alpha;
newPixel = newPixel << 8;
newPixel += p.red; newPixel = newPixel << 8;
newPixel += p.green; newPixel = newPixel << 8;
newPixel += p.blue;

return newPixel;

}
public static void setPixel(Image myImage, int i, int j, Pixel p)
{
int newPixel = colorToRGB(p);

// Write pixels into image
myImage.data.setRGB(i, j, newPixel);

}
public static void saveImage(Image myimage,String filename)
{
try {

File imageFile = new File(filename);
ImageIO.write(myimage.data, "jpg", imageFile);
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

//"jpg" is the format of the image

```

```


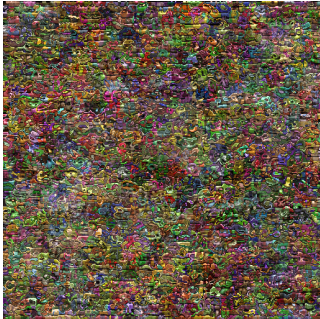



//imageFile is the file to be written to.

}
public static Image readImage(String fileName)
{
Image result=new Image();
try {
result.data = ImageIO.read(new File(fileName));
} catch (IOException e) {
throw new RuntimeException(e);
}
result.width = result.data.getWidth();
result.height = result.data.getHeight();

return result;
}
}

```

Résultats des exercices:

	Image de base	Image de composition	Résultat
Addition d'image			
Floue			

Solution

```

public class Main {
private static Image avg(Image original) {

int alpha, red, green, blue;
Pixel newPixel;

Image avg_gray =Image.createImage(original);

for(int i=0; i<original.width; i++) {
for(int j=0; j<original.height; j++) {

Pixel p=Image.getPixel(original, i, j);

// Get pixels by R, G, B
alpha = p.alpha;
red = p.red;
green = p.green;
blue = p.blue;

int newColor = (red + green + blue)/3;

// Return back to original format
newPixel = Pixel.createPixel(newColor, newColor, newColor, newColor);

// Write pixels into image
Image.setPixel(avg_gray,i, j, newPixel);

}
}

return avg_gray;

}
private static Image avglum(Image original) {

int alpha, red, green, blue;
Pixel newPixel;

Image avg_gray =Image.createImage(original);

```

```

for(int i=0; i<original.width; i++) {
for(int j=0; j<original.height; j++) {

Pixel p=Image.getPixel(original, i, j);

// Get pixels by R, G, B
alpha = p.alpha;
red = p.red;
green = p.green;
blue = p.blue;

int newColor = (int) (0.21 * red + 0.71 * green + 0.07 * blue);

// Return back to original format
newPixel = Pixel.createPixel(newColor, newColor, newColor, newColor);

// Write pixels into image
Image.setPixel(avg_gray,i, j, newPixel);

}
}

return avg_gray;

}
private static Image add(Image original,Image original2, float alpha1,float alpha2) {

int alpha, red, green, blue;
Pixel newPixel;

Image avg_gray =Image.createImage(original);

for(int i=0; i<original.width; i++) {
for(int j=0; j<original.height; j++) {

Pixel p1=Image.getPixel(original, i, j);
Pixel p2=Image.getPixel(original2, i, j);

// Get pixels by R, G, B
alpha = 1;
red = (int) (p1.red*alpha1+p2.red*alpha2);

```

```

green = (int) (p1.green*alpha1+p2.green*alpha2);
blue = (int) (p1.blue*alpha1+p2.blue*alpha2);

// Return back to original format
newPixel = Pixel.createPixel(alpha, red, green, blue);
// Write pixels into image
Image.setPixel(avg_gray,i, j, newPixel);

}
}

return avg_gray;

}
private static Image gauss(Image original) {

int alpha, red, green, blue;
Pixel newPixel;

Image avg_gray =Image.createImage(original);

for(int i=1; i<original.width-1; i++) {
for(int j=1; j<original.height-1; j++) {

Pixel p1=Image.getPixel(original, i-1, j-1);
Pixel p2=Image.getPixel(original, i-1, j);
Pixel p3=Image.getPixel(original, i-1, j+1);
Pixel p4=Image.getPixel(original, i, j-1);
Pixel p5=Image.getPixel(original, i, j);
Pixel p6=Image.getPixel(original, i, j+1);
Pixel p7=Image.getPixel(original, i+1, j-1);
Pixel p8=Image.getPixel(original, i+1, j);
Pixel p9=Image.getPixel(original, i+1, j+1);

// Get pixels by R, G, B
alpha =
(p1.alpha+p2.alpha+p3.alpha+p4.alpha+2*p5.alpha+p6.alpha+p7.alpha+p8.alpha+p9.alpha)/10;
red = (p1.red+p2.red+p3.red+p4.red+2*p5.red+p6.red+p7.red+p8.red+p9.red)/10;
green =
(p1.green+p2.green+p3.green+p4.green+2*p5.green+p6.green+p7.green+p8.green+p9.green)/10;
blue = (p1.blue+p2.blue+p3.blue+p4.blue+2*p5.blue+p6.blue+p7.blue+p8.blue+p9.blue)/10;

```

```
// Return back to original format
newPixel = Pixel.createPixel(alpha, red, green, blue);

// Write pixels into image
Image.setPixel(avg_gray, i, j, newPixel);

}
}

return avg_gray;

}

public static void main(String [] args)
{
Image i=Image.readImage("C:\\PGM\\Scic\\project\\testimage\\lena30.jpg");
Image avgimage=avglum(i);
Image.saveImage(avgimage, "C:\\PGM\\Scic\\project\\testimage\\lena30avglum.jpg");
Image avgimage2=avg(i);
Image.saveImage(avgimage2, "C:\\PGM\\Scic\\project\\testimage\\lena30avg.jpg");
Image gauss=gauss(i);
Image.saveImage(gauss, "C:\\PGM\\Scic\\project\\testimage\\lena30gauss.jpg");
Image texture=Image.readImage("C:\\PGM\\Scic\\project\\testimage\\texture.png");
Image addImage=add(i, texture, 0.5f, 0.2f);
Image.saveImage(addImage, "C:\\PGM\\Scic\\project\\testimage\\addImage.jpg");

}
}
```