**Programmation Orientée Objets avec Java**

**Chapitre 1 : les bases de Java**

**Stéphane Malandain / Yassin Rekik**

**Semestre d’automne 2024**

* 1. **Exercice**
* Écrire un code qui simule le comportement d’une boucle while à l’aide de la boucle do while
* Écrire un code qui simule le comportement d’une boucle do while à l’aide de la boucle while

import java.util.Arrays;

public class Loop {

public static void whileWithDoWhile(int i) {

// essayez par exemple de reproduire l'équivalent suivant:

while(i < 5) {

System.out.println(i);

i++;

}

// devient:

if (i < 5) {

do {

System.out.println(i);

i++;

} while (i < 5);

}

}

public static void doWhileWithWhile(int i) {

// essayez par exemple de reproduire l'équivalent suivant:

do {

System.out.println(i);

i++;

} while (i < 5);

// devient:

System.out.println(i);

i++;

while(i < 5) {

System.out.println(i);

i++;

}

}

public static void main(String[] args) {

//whileWithDoWhile(10);

//doWhileWithWhile(10);

for (String a: args) {

System.out.println(a);

}

for (int i = 0; i < args.length; i += 1) {

System.out.println( args[i] );

}

int i = 0;

while (i < args.length) {

System.out.println( args[i] );

i += 1;

}

Arrays.asList(args).forEach( a -> System.out.println(a) );

}

}

**1.2 Exercice**

Indiquez quelles lignes ne compilent pas :

**Une image contenant texte

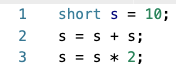
Description générée automatiquement**

**1  : ok**

1. **: non**
2. **: ok**
3. **: ok**

**1.3 Exercice**

Indiquez quelles lignes ne compilent pas :

****

**2 : non**

**3 : non**

**Une image contenant texte

Description générée automatiquement**

**1.4 Exercice**

****

Que vaut i à la fin de l’exécution ?

**Une image contenant texte

Description générée automatiquement**

**1.5 Exercice**

On dispose des méthodes suivantes :

****

Et des déclarations suivantes :

Une image contenant texte

Description générée automatiquement

Spécifiez si les appels suivants sont corrects ou non tout en justifiant vos réponses

**Une image contenant table

Description générée automatiquement Une image contenant table

Description générée automatiquement**

**Pratique**

**1.6 Exercice (Triangle)**

A l’aide d’un paramètre n indiquant la hauteur, écrivez une fonction printTriangle(int n) qui affiche un triangle sous cette forme (ici, n = 4) :

****

public class serie1\_16 {

static void printTriangle(int n) {

for (int i = 0; i<n; i++) {

for (int j=0;j<=i; j++) System.out.print("\*");

System.out.println();

}

}

public static void main(String[] args) {

printTriangle(4);

}

}

**1.7 Exercice (Sapin)**

A l’aide d’un paramètre n indiquant la hauteur, écrivez une fonction

printSapin(int n) qui affiche un sapin sous cette forme (ici, n = 4) :

**Une image contenant texte

Description générée automatiquement**

public class serie1\_17 {

static void printSapin(int n) {

int up = 1;

for (int i = 0; i<n; i++) {

for (int j=1;j<n-i; j++) System.out.print(" ");

for (int k=0;k<up; k++) System.out.print("\*");

System.out.println();

up += 2;

}

}

public static void main(String[] args) {

printSapin(4);

}

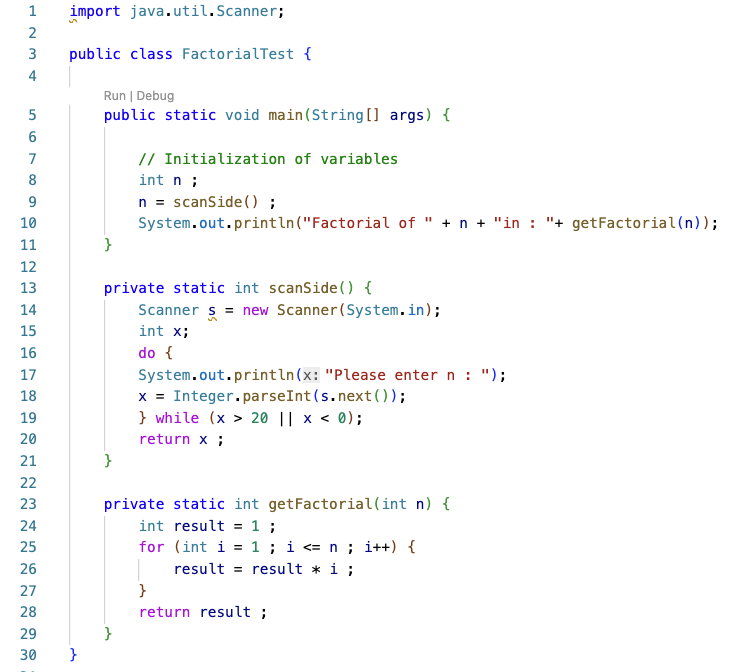
}

**1.8 Exercice (Factorielle)**

Réalisez une fonction permettant de calculer la factorielle de n.

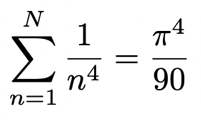
**Une image contenant texte

Description générée automatiquement**

****

**1.9 Exercice (Pi)**

Réalisez une fonction permettant de calculer une approximation de ∏ :



Cette fonction prend n en paramètre et retourne ∏

**1.10 Exercice (Jeu du serpent)**

Écrivez un programme qui affiche un serpent. A chaque itération, le serpent est affiché à l’aide d’une suite d’étoiles. Le programme demande ensuite d’augmenter ou de diminuer la taille du serpent. Le programme s’arrête lorsque la taille du serpent est nulle.

Exemple :

Une image contenant texte

Description générée automatiquement

import java.util.Scanner;

import java.util.\*;

public class serpent {

public static void printSerpent(int n) {

for (int i=1; i<n; i++) System.out.print("\*");

System.out.print("(^^)");

System.out.println();

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

boolean onContinue = true;

int taille = 1;

char direction = '+';

String val;

while (onContinue) {

printSerpent(taille);

System.out.print("Direction:");

val = scanner.next();

direction = val.charAt(0);

if (direction == '+') taille ++;

if (direction == '-') taille --;

onContinue = (taille > 0);

}

scanner.close();

System.out.println(" Bye !");

}

}

**1.11 Exercice (Vecteurs)**

Réalisez un ensemble de fonctionnalités permettant le calcul vectoriel sur des vecteurs. Utilisez un tableau de doubles pour représenter un vecteur.

Écrivez les méthodes statiques suivantes :

* La méthode add. Elle prend deux vecteurs et retourne leur addition.

Ex. add(new double[]{1.0, 2.0, 3.0}, new double[]{2.0, 1.0, -5.0}) retourne double[3]{3.0, 3.0, -3.0}.

* La méthode mul qui multiplie un vecteur par une valeur numérique.
* A l’aide des 2 premières fonctions, écrivez la méthode statique sub qui soustrait le deuxième vecteur du premier.
* La méthode len qui retourne le nombre de composante d’un vecteur
* norm qui retourne la norme (ou la longueur) d’un vecteur.

Par exemple, norm(new double[]{1.0, 2.0, 2.0}) retourne 3.0.

* Le calcul de la norme d’un vecteur  Une image contenant texte, horloge, jauge

  Description générée automatiquementest 
* sum qui prend une liste de vecteurs en paramètre et les additionne
* norms qui prend une liste de vecteurs en paramètre et retourne la norme de leur addition
* concat qui concatène 2 vecteurs
* sliceFrom qui retourne un sous-ensemble du vecteur jusqu’ä un indice (non compris)
* slice qui est la combinaison des deux précédentes (avec un index de début et un de fin)

Remarques :

* Respectez le nommage
* Respectez les conventions de nommage du langage :

<https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html>

* Écrivez des méthodes courtes et concises
* Retournez des nouvelles copies de tableaux. Ne modifiez pas vos arguments
* Utilisez les fonctionnalités offertes par la classe Arrays :

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Arrays.html>

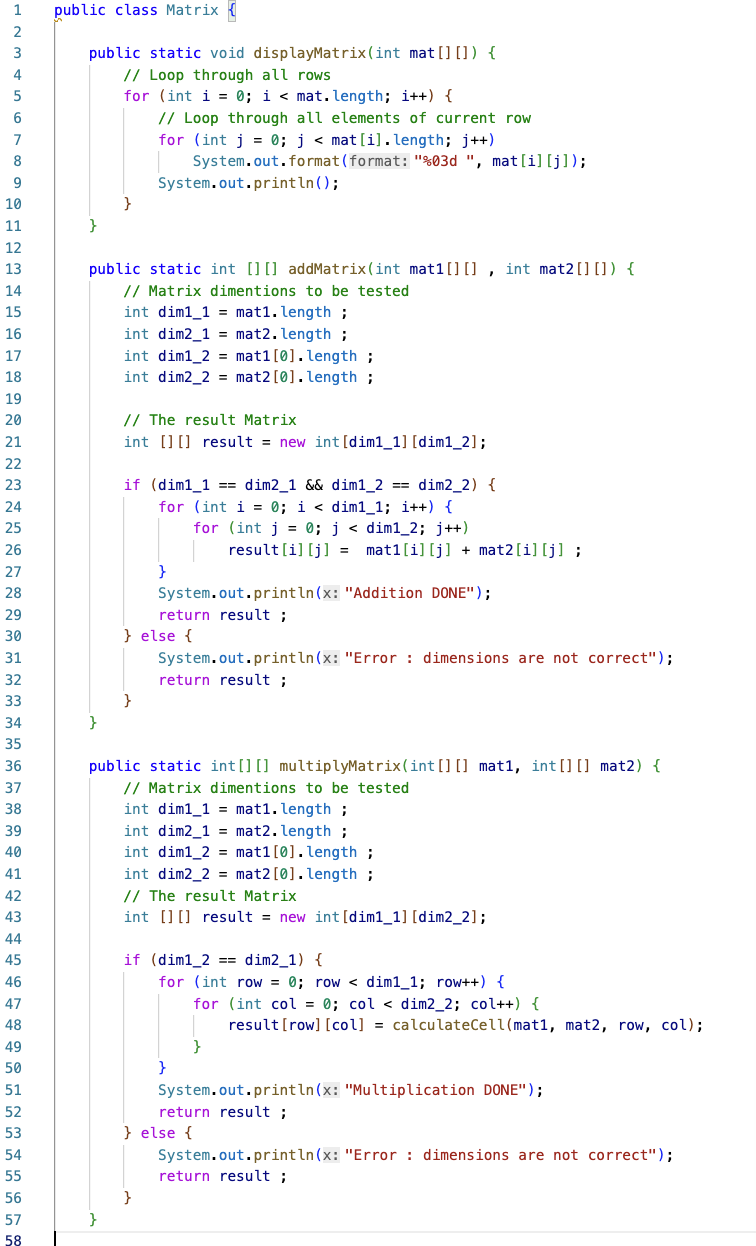
* Écrivez vos méthodes statiques dans un fichier Vector.java (sans main)
* La méthode principale avec vos tests doit se trouver dans un fichier App.java
* Les fonctions qui retournent des vecteurs doivent retourner un vecteur vide si les vecteurs entrés en arguments ne sont pas conformes !

**1.12 Exercice (Matrices)**

Réalisez des fonctionnalités sur les matrices :

* L’affichage d’une matrice
* La multiplication de deux matrices
* L’addition de deux matrices

Retournez une matrice vide en cas d’erreur





**1.13 Exercice (Tableaux)**

Écrivez une méthode qui compte le nombre d’éléments positifs dans un tableau de tableaux. Un exemple d’utilisation est fourni dans la méthode principale main suivante :

**Une image contenant texte

Description générée automatiquement**

**1.14 Exercice (Strings)**

Écrivez les fonctionnalités ci-dessous :

import java.util.List;

public class App {

/\* Détermine si un nombre qui se trouve dans une chaîne de caractères contient

\* un entier

\* Note: prend en compte le signe, mais ne prend pas en compte les espaces

\* Exemples:

\* isNumeric("42") -> true

\* isNumeric(" 22 ") -> true

\* isNumeric(" -33 ") -> true

\* isNumeric(" 22.0") -> false

\* isNumeric("2f3") -> false \*/

public static boolean isNumeric(String term) {

char[] chars = term.trim().toCharArray();

if (chars[0] == '-') {

chars[0] = '0';

}

for (char c: chars) {

if (!Character.isDigit(c)) {

return false;

}

}

return true;

}

/\* Retourne un tableau d'indices où chaque valeur indique la position d'un

\* caractère dans une chaîne

\* Exemples:

\* indexes("maison", 'i') -> {2}

\* indexes("tralala", 'a') -> {2,4,6}

\* indexes("coucou", 'x') -> {} \*/

public static int[] indexes(String term, char c) {

List<Integer> indexes = new ArrayList<>();

int i = 0;

for (char ch: term.toCharArray()){

if (ch == c) {

indexes.add( i );

}

i += 1;

}

return transform( indexes );

}

private static int[] transform(List<Integer> is) {

int[] res = new int[is.size()];

for (int i = 0; i < is.size(); i += 1) {

res[i] = is.get(i);

}

return res;

}

}