



Résumily | Université de Boumerdes

Document

Solution Série n°1 2022/2023

Module

Programmation Orientée Objet

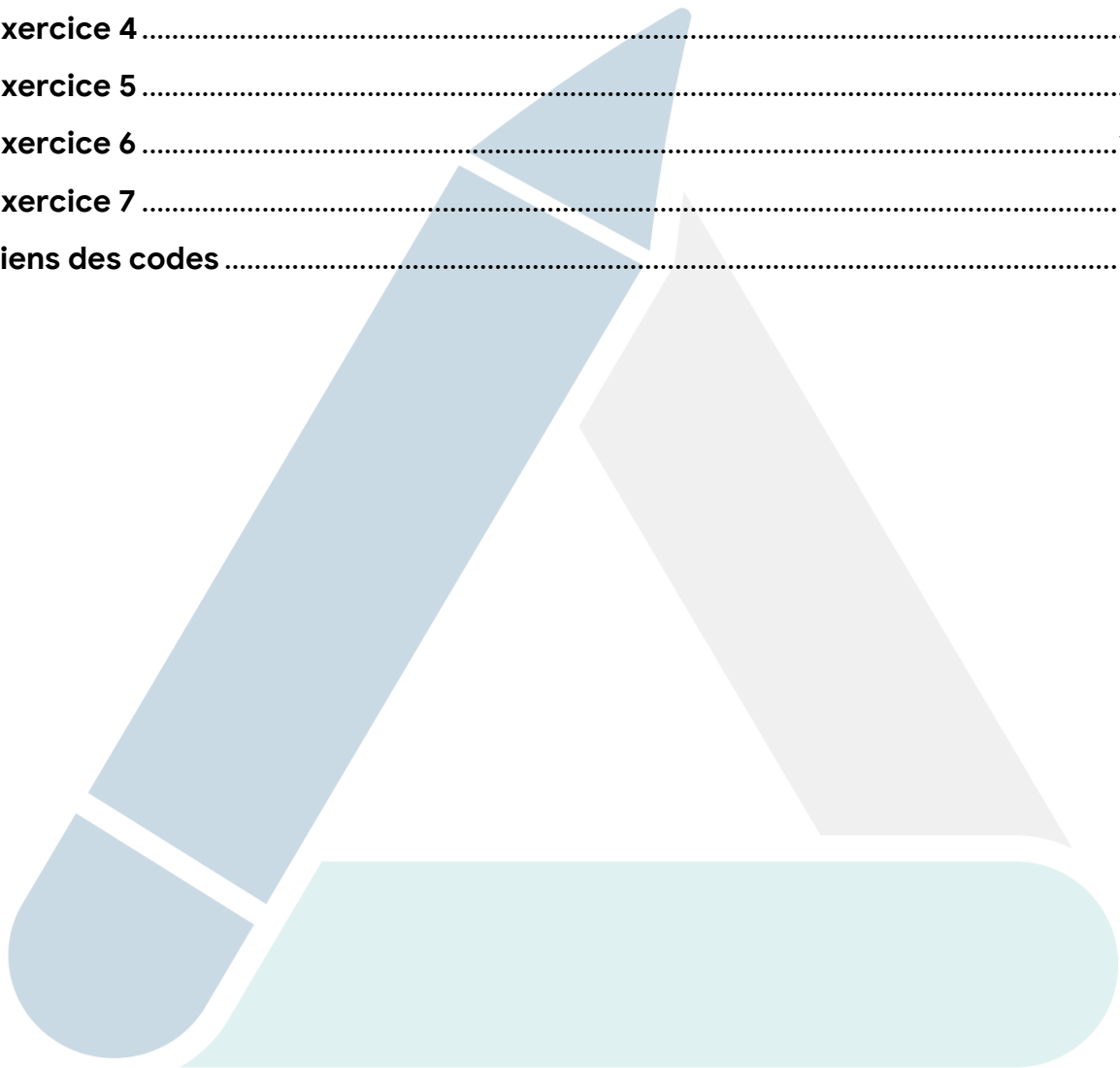
Réalisé par

Résumily



TABLE DES MATIERES

| | |
|------------------------------|-----------|
| Introduction..... | 3 |
| Exercice 1..... | 4 |
| Exercice 2 | 5 |
| Exercice 3 | 6 |
| Exercice 4 | 7 |
| Exercice 5 | 8 |
| Exercice 6 | 10 |
| Exercice 7 | 12 |
| Liens des codes | 14 |



INTRODUCTION

Bienvenue dans ce document qui contient la solution détaillée de la série 1 du module de Programmation Orientée Objet de l'année universitaire 2022/2023. Ce document a été créé par l'équipe Résumily afin de vous aider à mieux comprendre les concepts fondamentaux de la programmation orientée objet et de vous fournir des solutions claires et détaillées pour les exercices proposés dans cette série.

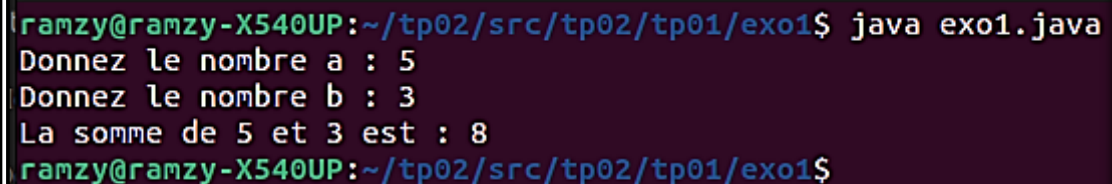
Nous espérons que ce document vous sera utile dans votre apprentissage de la programmation orientée objet et nous vous souhaitons bonne lecture !

[Énoncé de la Série n°1](#)

EXERCICE 1

```
1 import java.util.Scanner;
2
3 public class exo1 {
4
5     public static void main(String[] args) {
6         int a,b,s;
7         Scanner scan = new Scanner(System.in);
8         System.out.print("Donnez le nombre a : ");
9         a = scan.nextInt();
10        System.out.print("Donnez le nombre b : ");
11        b = scan.nextInt();
12        s = a+b;
13        System.out.println("La somme de "+a+" et "+b+" est :
14                               "+s);
15    }
16 }
```

Output :



```
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo1$ java exo1.java
Donnez le nombre a : 5
Donnez le nombre b : 3
La somme de 5 et 3 est : 8
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo1$
```

Ce programme permet à l'utilisateur de saisir deux nombres entiers, calcule leur somme, et affiche le résultat à l'écran.

Pour ce faire, le programme crée trois variables entières a, b et s, où a et b sont initialisées avec les valeurs saisies par l'utilisateur à l'aide de l'objet Scanner. La somme est ensuite calculée en ajoutant les valeurs de a et b, puis stockée dans s.

Enfin, le résultat est affiché à l'écran à l'aide de la méthode println de l'objet System.out.

EXERCICE 2

```
1 import java.util.Scanner;
2
3 public class exo2 {
4
5     public static void main(String[] args) {
6         int C,F;
7         Scanner scan = new Scanner(System.in);
8         System.out.print("Donnez la temperature en C°
9         : ");
10        C = scan.nextInt();
11        F = (C*9/5)+32;
12        System.out.println(C+" C° = "+F+" F°");
13    }
14 }
```

Output :

```
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo2$ java exo2.java
Donnez la temperature en C° : 32
32 C° = 89 F°
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo2$
```

Le programme demande à l'utilisateur de saisir une température en degrés Celsius. Ensuite, il utilise la formule de conversion pour calculer la température en degrés Fahrenheit. Enfin, il affiche la température en degrés Fahrenheit à l'écran.

EXERCICE 3

```
1  import java.util.Scanner;
2
3  public class exo3 {
4
5      public static void main(String[] args) {
6          final float TVA = 0.2f;
7          float HT, TTC;
8          Scanner scan = new Scanner(System.in);
9          System.out.print("Donnez le prix HT : ");
10         HT = scan.nextFloat();
11         TTC = HT + (HT*TVA);
12         System.out.println("Le prix TTC est : "+TTC+" DA");
13     }
14
15 }
```

Output :

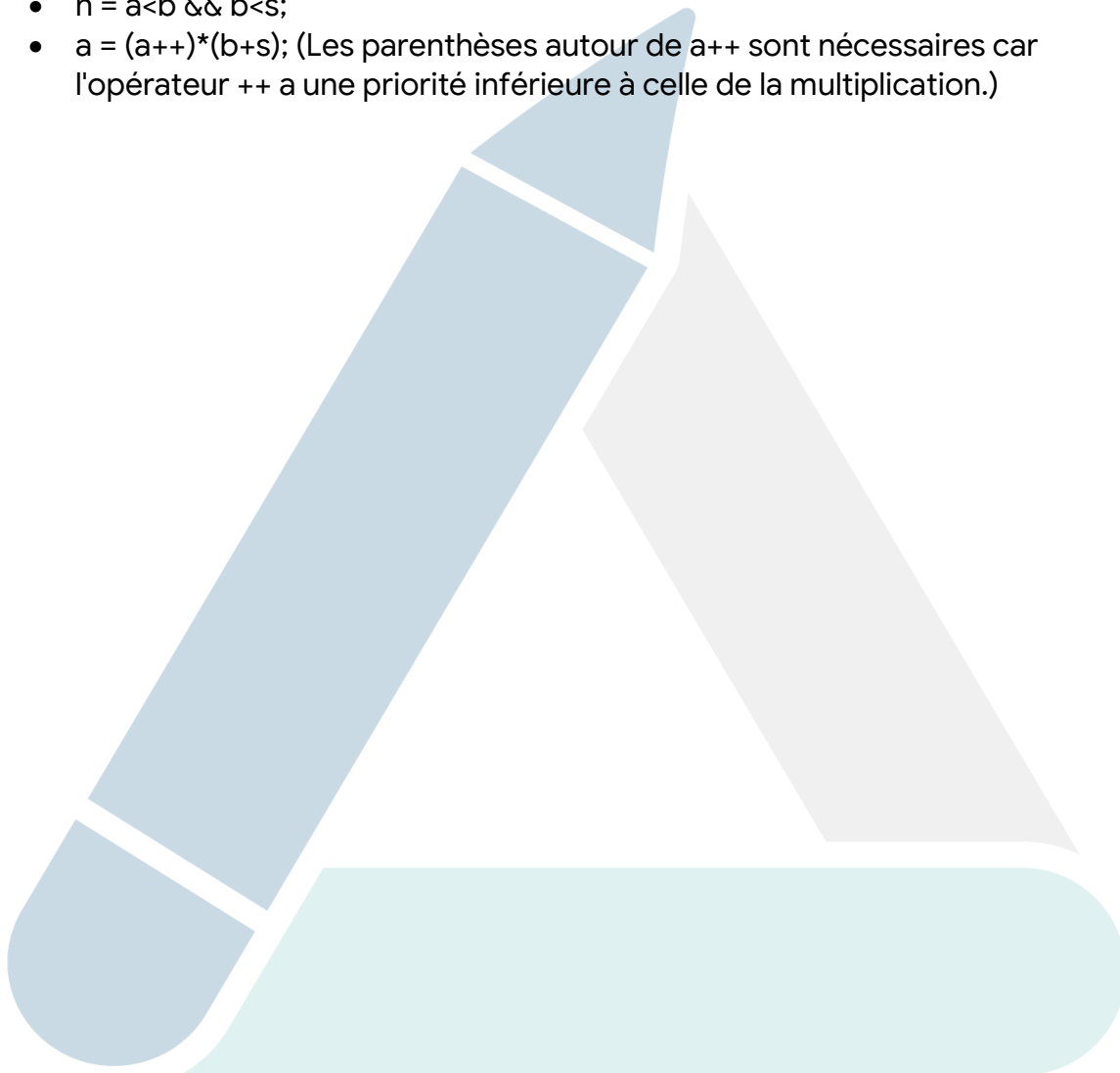
```
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo3$ java exo3.java
Donnez le prix HT : 4750
Le prix TTC est : 5700.0 DA
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo3$
```

Ce programme permet de calculer le prix toutes taxes comprises (TTC) en fonction du prix hors taxes (HT) et d'un taux de TVA fixé à 20%. Il utilise une variable constante TVA pour stocker la valeur fixe du taux de TVA et demande à l'utilisateur d'entrer le prix HT. Le programme calcule ensuite le prix TTC en ajoutant au prix HT le produit du prix HT par le taux de TVA. Enfin, il affiche le résultat à l'écran avec une unité monétaire (DA).

EXERCICE 4

Voici les expressions corrigées avec les parenthèses inutiles éliminées :

- `a = b+5;`
- `a = b=s + 2;`
- `boolean n = a==b;`
- `n = a<b && b<s;`
- `a = (a++)*(b+s);` (Les parenthèses autour de `a++` sont nécessaires car l'opérateur `++` a une priorité inférieure à celle de la multiplication.)



EXERCICE 5

```
1  import java.util.Scanner;
2
3  public class exo5 {
4
5      public static void main(String[] args) {
6          int i, j, n ;
7          i = 0 ; n = i++ ;
8          System.out.println("A : i=" + i + " n=" + n ) ;
9          i = 10 ; n = ++ i ;
10         System.out.println("B : i=" + i + " n=" + n );
11         i = 20 ; j = 5 ; n = i++ * ++ j ;
12         System.out.println("C : i=" + i + " j=" + j + " n=" +
13             n ) ;
14         i = 15 ; n = i += 3 ;
15         System.out.println("D : i=" + i + " n=" + n ) ;
16         i = 3 ; j = 5 ; n = i *-- j ;
17         System.out.println( "E : i=" + i + " j=" + j + " n=" +
18             n ) ;
19     }
```

Dans le premier bloc, la valeur de *i* est initialisée à 0 puis est assignée à *n* qui prend donc la valeur 0. Après cela, la valeur de *i* est incrémentée de 1. Donc, à la fin de ce bloc, *i*=1 et *n*=0.

Dans le deuxième bloc, la valeur de *i* est d'abord incrémentée de 1 (donc *i*=11). Ensuite, la nouvelle valeur de *i* est assignée à *n* qui prend donc la valeur de 11. Donc, à la fin de ce bloc, *i*=11 et *n*=11.

Dans le troisième bloc, la valeur de *i* est initialisée à 20 et la valeur de *j* à 5. Ensuite, *j* est incrémentée de 1 (donc *j*=6) et la multiplication de *i* (20) et de *j* (6) est assignée à *n*, donc *n*=120. Après cela, la valeur de *i* est incrémentée de 1, donc *i*=21. Donc, à la fin de ce bloc, *i*=21, *j*=6 et *n*=120.

Dans le quatrième bloc, la valeur de *i* est initialisée à 15. Ensuite, la valeur 3 est ajoutée à *i* (donc *i*=18) et la nouvelle valeur de *i* est assignée à *n* qui prend donc la valeur de 18. Donc, à la fin de ce bloc, *i*=18 et *n*=18.

Dans le cinquième bloc, la valeur de i est initialisée à 3 et la valeur de j à 5. Ensuite, j est décrémentée de 1 (donc $j=4$) et la multiplication de i (3) et j (4) est assignée à n , donc $n=12$. Après cela, la valeur de i est assignée à la nouvelle valeur de i multipliée par j , donc $i=12$. Donc, à la fin de ce bloc, $i=12$, $j=4$ et $n=12$.

Lorsque le code est compilé et exécuté, il affiche :

```
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo5$ java exo5.java
A : i=1 n=0
B : i=11 n=11
C : i=21 j=6 n=120
D : i=18 n=18
E : i=12 j=4 n=12
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo5$
```

EXERCICE 6

```
1 public class exo6 {
2
3     public static void main(String[] args) {
4         int n=10, p=5, q=10, r;
5         r = (int) (p = q) ;
6         System.out.println("A : n=" + n + " p=" + p + " q="
7 + q+ " r=" + r ) ;
8         n = p = q = 5 ;
9         n += p += q ;
10        System.out.println( "B : n=" + n + " p=" + p + " q="
11 + q );
12        q = n < p ? n++ : p++ ;
13        System.out.println("C : n=" + n + " p=" + p + " q=" +
14 q ) ;
15        q = n > p ? n++ : p++ ;
16        System.out.println( "D : n=" + n + " p=" + p + " q=" +
17 q ) ;
18    }
19 }
```

Output :

```
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo6$ java exo6.java
A : n=10 p=10 q=10 r=10
B : n=15 p=10 q=5
C : n=15 p=11 q=10
D : n=16 p=11 q=15
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo6$
```

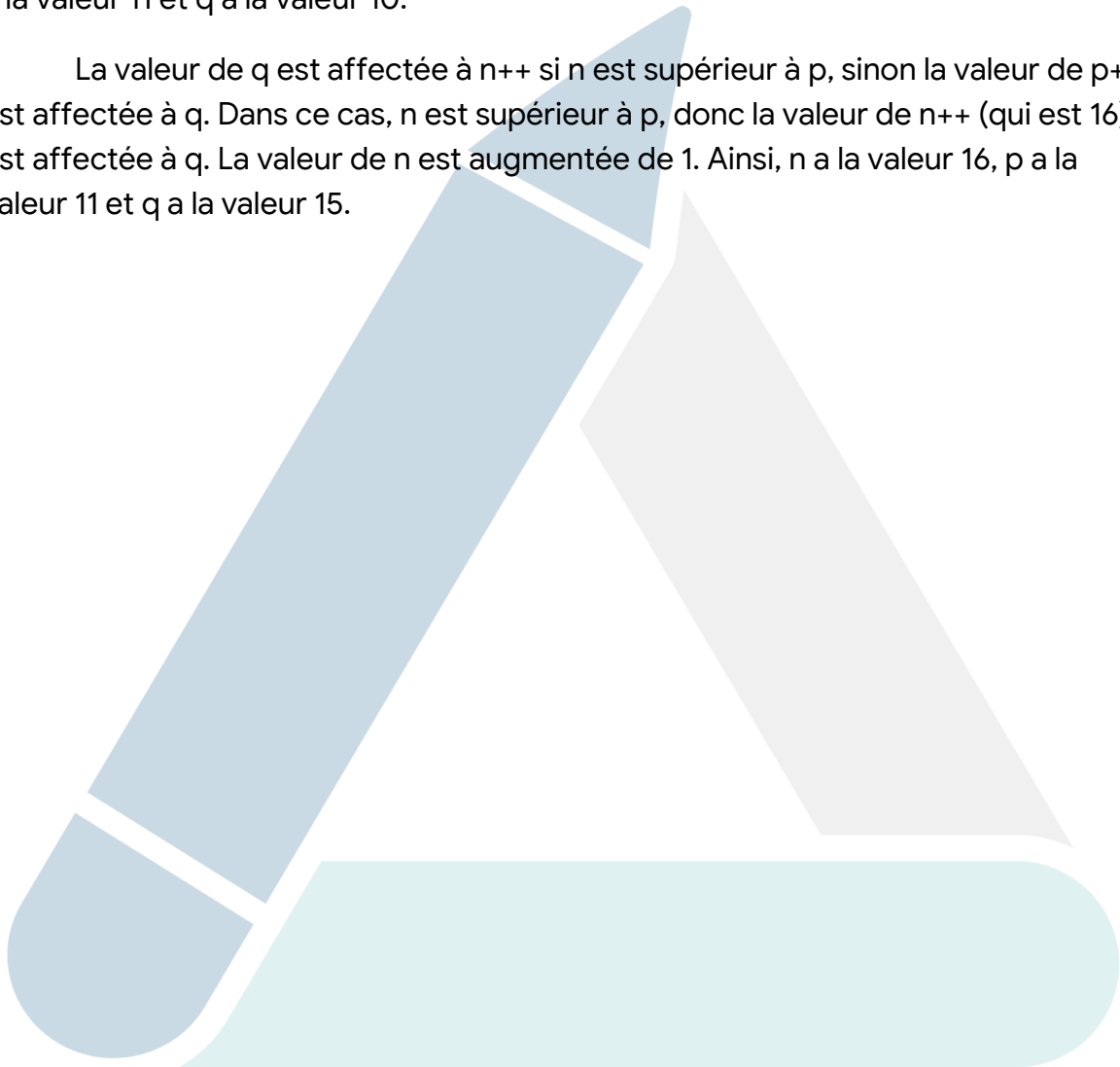
Les variables n, p et q sont initialisées respectivement à 10, 5 et 10.

La variable r est initialisée à la valeur de p après que p a été affecté à la valeur de q grâce à l'opérateur d'assignation =. Le résultat de l'opération est ensuite converti en un entier à l'aide de la notation (int). Ainsi, r a la valeur 10, tout comme q et p.

Les variables n , p et q sont toutes initialisées à 5 à l'aide de l'opérateur d'assignation $=$. Ensuite, la valeur de q est ajoutée à p , puis le résultat est ajouté à n grâce à l'opérateur d'addition composée $+=$. Ainsi, n a la valeur 15, tandis que p prend la valeur 10, et q prend la valeur 5.

La valeur de q est affectée à $n++$ si n est inférieur à p , sinon la valeur de $p++$ est affectée à q . Dans ce cas, n est supérieur à p , donc la valeur de $p++$ (qui est 10) est affectée à q . La valeur de n et de p est augmentée de 1. Ainsi, n a la valeur 15, p a la valeur 11 et q a la valeur 10.

La valeur de q est affectée à $n++$ si n est supérieur à p , sinon la valeur de $p++$ est affectée à q . Dans ce cas, n est supérieur à p , donc la valeur de $n++$ (qui est 16) est affectée à q . La valeur de n est augmentée de 1. Ainsi, n a la valeur 16, p a la valeur 11 et q a la valeur 15.



EXERCICE 7

```
1 public class exo7 {
2
3     public static void main(String[] args) {
4         int n, p, q ; boolean k;
5         n = 5 ; p = 2 ; /* cas 1 */
6         k = n++ > p || p++ != 3 ;
7         System.out.println( "A : n = " + n + " p = " + p + " k = " + k ) ;
8         n = 5 ; p = 2 ; /* cas 2 */
9         k = n++ < p || p++ != 3 ;
10        System.out.println( "B : n = " + n + " p = " + p + " k = " + k ) ;
11        n = 5 ; p = 2 ; /* cas 3 */
12        k = ++n == 3 && ++p == 3 ;
13        System.out.println( "C : n = " + n + " p = " + p + " k = " + k ) ;
14        n = 5 ; p = 2 ; /* cas 4 */
15        k = ++n == 6 && ++p == 3 ;
16        System.out.println( "D : n = " + n + " p = " + p + " k = " + k ) ;
17    }
18
19 }
```

Output :

```
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo7$ java exo7.java
A : n = 6 p = 2 k = true
B : n = 6 p = 3 k = true
C : n = 6 p = 2 k = false
D : n = 6 p = 3 k = true
ramzy@ramzy-X540UP:~/tp02/src/tp02/tp01/exo7$
```

Le code utilise les opérateurs logiques || et && pour effectuer des tests de condition sur les variables n, p et k. Voici les résultats attendus pour chaque cas :

Cas 1 : Dans ce cas, la première condition est vraie ($5 > 2$) donc le programme n'a pas besoin de vérifier la seconde condition ($p++ \neq 3$), qui ne sera donc pas exécutée. La variable n est ensuite incrémentée ($n = 6$) et la variable k est vraie car la première condition est vraie. La sortie sera : "A : n = 6 p = 2 k = true".

Cas 2 : La première condition est fausse (5 n'est pas inférieur à 2), donc le programme doit vérifier la seconde condition ($p++ \neq 3$). Dans ce cas, p sera incrémenté ($p = 3$) et la seconde condition est vraie car p est différent de 3. La variable k sera donc vraie. La sortie sera : "B : n = 6 p = 3 k = true".

Cas 3 : Dans ce cas, la première condition est fausse (n est égal à 6 après avoir été préalablement incrémenté), donc le programme n'a pas besoin de vérifier la

seconde condition ($p++ == 3$), qui ne sera donc pas exécutée. La variable k sera fausse car la première condition est fausse. La sortie sera : "C : $n = 6$ $p = 2$ $k = false$ ".

Cas 4 : Dans ce cas, les deux conditions sont vraies, donc les variables n et p seront incrémentées ($n = 6$ et $p = 3$). La variable k sera vraie car les deux conditions sont vraies. La sortie sera : "D : $n = 6$ $p = 3$ $k = true$ ".



LIENS DES CODES

Vous trouverez tous les codes (fichier .java) dans notre Github

github.com/Resumily/poo-tp1

