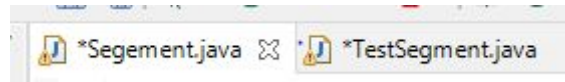


Compte rendu à rendre avant le jeudi 10 septembre 2020

Classe UML segment

Dans un premier temps, nous devons créer nous avons créé deux classes dont le premier se nomme **segment.java** et le second s'intitule **TestSegment.java** comme ci-dessous :



Une fois que ceci a été fait, nous allons débiter à inscrire les variables de la classe **segment.java** comme ci dessous :

```
1 public class segment {
2     public static void main(String[] args) {
3     }
4
5     private int extr1;
6     private int extr2;           //Introduction des variables dans la classe en précisant le type de donnée.
7     private int Longueur;
8 }
```

Une fois ceci fait, nous devons faire le constructeur par les deux extrémités en paramètre comme ci-dessous :

```
public segment (int extr1, int extr2) {
    this.extr1= extr1;           // Introduction du constructeur avec les variables déclarées précédemment dans les paramètres.
    this.extr2=extr2;
}
```

Une fois que ceci a été fait nous allons commencé a créer les méthodes dont la première méthode se nomme : “ **public void ordonne()** ”. Le mot clé “VOID” signifie qu’elle ne retourne pas de valeurs. Voici la capture:

```
public void ordonne() {
    int a = this.extr1;           // Introduction de la méthode en inialisant les variables en lettres a et b.
    int b= this.extr2;

    if(a < b) {
    }else if (a > b) {               // Création de d'une condition dans lequel nous allons précise
        this.extr1=b;
        this.extr2=a;             // Si a est supérieur à b, les valeurs s'inverseront.
    }else {
        System.out.println("Erreur!");
    }
}
```

Dans cette méthode, nous avons initialisé les extrémités en lettres. Donc notre but est lorsque la valeur est supérieure, elle changera avec l’autre extrémité. Pour cela nous avons créer un e condition “ **if** ”. Dans cette condition nous avons précisé que si la valeur A c’est-à-dire “extr1” est inférieur a B alors les valeurs vont s'échanger et finira par afficher “erreur”.

Ensuite, nous avons créer la méthode Longueur qui permettra de retourner la valeur donc pour calculer la longueur il faut savoir si c’est l’extrémité 1 + l’extrémité 2, alors nous avons fait pareil dans le code et de nous retourner la valeur de la longueur comme ci-dessous :

```

public int getLongueur() {
    this.Longueur= extr1+extr2;    // Nous avons préciser que la longueur se calcule en additionnant les deux extrémités
    return Longueur;              // et nous retournera la valeur de la longueur.
}

```

Par la suite, nous avons créer une nouvelle méthode qui s'intitule “ **appartient()** ” dont en paramètre, la valeur “ **i** ”.

Par la suite, nous avons créer une condition dans cette méthode. Cette condition aura pour but de dire si la valeur que l'utilisateur a saisi appartient ou non au segment comme ci-dessous :

```

public String appartient(int i) {
    String rep = null;

    if (i > this.extr1 && i < this.extr2) { // Si la valeur saisie est supérieure à A et inférieure à B, le point appartient au segment.
        rep = "Félicitation le point " + i + " appartient au segment !";
    }else {
        rep = "!";
    }
    return rep;
}

```

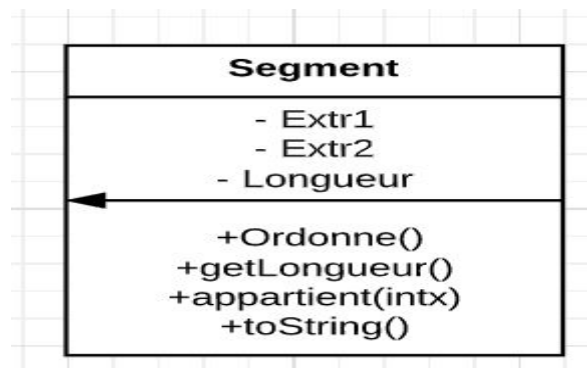
Ensuite nous avons créer une méthode “**ToString**” comme ci-dessous :

```

public String toString() {
    return ("Extr1: " + this.extr1 + " | Extr2: " + this.extr2 + " | Longueur: " + this.Longueur); // Affiche les valeurs au début
}

```

Voici la représentation UML (Unified Modeling Language) de la classe **Segment** :



Maintenant, nous avons créer la classe main qui s'intitule “**TestSegment**” comme ci-dessous:



Maintenant, nous allons créer l'objet de la classe avec les valeurs des extrémités dans le main comme ci-dessous :

```

segment.java  TestSegment.java
1 import java.util.Scanner;
2 public class TestSegment {
3
4 public static class Main {
5
6     public static void main(String[] args) {
7         System.out.println("Création segment!");
8
9         Scanner userScan = new Scanner (System.in); //Pour faire le lien avec la classe Segment.
10
11
12         segment segment1 = new segment(24,12); //création d'un nouveau segment

```

Par la suite nous allons afficher l'état de cet objet en utilisant la méthode "ToString" comme ci-dessous :

```

    System.out.println(segment1.appartient(i));

    System.out.println(segment1.toString()); // Affiche la valeur du segment1

```

Nous allons afficher la longueur en utilisant l'accesseur "GET" pour faire appeler la valeur

```

    segment1.ordonne(); //on met une condition pour la méthode Ordonne(){
    System.out.println(segment1.segmentDraw());
    segment1.getLongueur(); // permet d'afficher la longueur

```

Code main complet :

```

public static class Main {

    public static void main(String[] args) {
        System.out.println("Création segment!");

        Scanner userScan = new Scanner (System.in); //Pour faire le lien avec la classe Segment.

        segment segment1 = new segment(24,12); //création d'un nouveau segment

        segment1.ordonne(); //on met une condition pour la méthode Ordonne(){
        System.out.println(segment1.segmentDraw());
        segment1.getLongueur(); // permet d'afficher la longueur

        System.out.println("Veuillez introduire une valeur |");
        int i = userScan.nextInt(); // Saisir une valeur

        while( segment1.appartient(i) == "!" ) { // Tant que la valeur ne correspond pas au segment.
            System.out.println("Désolé ce nombre n'est pas dans le segment !");
            System.out.println("Veuillez introduire un autre nombre :");
            i = userScan.nextInt();
            segment1.appartient(i);
        }

        System.out.println(segment1.appartient(i));

        System.out.println(segment1.toString()); // Affiche la valeur du segment1

```

Maintenant testons le code :

```

TestSegment.Main [Java Application] C:\Program F
Création segment!
12 <-----> 24
Veuillez introduire une valeur :

```

Nous avons bien les deux extrémités et la possibilité de saisir notre valeur...

11

```
Désolé ce nombre n'est pas dans le segment !  
Veillez introduire un autre nombre :
```

Nous pouvons voir que lorsque je saisis une valeur inférieure à l'extrémité 1 un message s'affiche à l'écran comme quoi le nombre n'est pas dans le segment.

Puis si on saisie un nombre qui est égal ou supérieur à l'extrémité 1 nous aurons ceci :

13

```
Félicitation le point 13 appartient au segment !  
Extr1: 12 | Extr2: 24 | Longueur: 36
```

Nous pouvons voir que le point correspond au segment avec les 3 critères du segment qui s'affiche en bas (extr1 , extr2 , Longueur).