

Calculatrices non autorisées.

Documents non autorisés.

Les exercices sont indépendants. Les temps donnés sont indicatifs.

1–Questions de cours (20 minutes)

Répondez brièvement aux questions suivantes :

1. Que signifie le terme *bytecode* ?
2. Quelles sont les propriétés d'une *classe abstraite* ?
3. Donnez un exemple d'héritage et un exemple de coercion de type ("cast")

2–Points, Droites (40 minutes)

On vous propose de créer 2 classes permettant de modéliser des points et des droites.

```
public class Point {
    public double x;
    public double y;

    public Point(double x,double y){
        this.x = x;
        /* A completer */
    }

    public double distance(Point p){
        double dX = p.x - this.x;
        double dY = /* A completer */
        return Math.sqrt(dX*dX+dY*dY);
    }

    public void translater(double dX, double dY){
        /* A completer */
    }

    public Point milieu(Point p){
        /* A completer */
    }

    public boolean equals(Point p){
        /* A completer */
    }

    public String toString(){
        return " ( x = " + x +
            " , y = " + y + " ) ";
    }
}

public class Droite {
    public Point p1 = null;
    public Point p2 = null;

    public Droite(){
        /* A completer */
    }

    /*
     * A completer
     */
}
```

Question 1 Complétez le corps du constructeur de la classe `Point`. Complétez le corps de la méthode `double distance(Point p)`.

Question 2 Complétez le corps de la méthode `void translater(double deltaX, double deltaY)`.

Question 3 Donnez un exemple d'appel au constructeur, à la méthode `translater`, ainsi qu'à la méthode `toString`

Question 4 Complétez le corps de la méthode `Point milieu(Point p)`. Cette méthode doit créer et retourner un nouveau point situé au milieu des `Point this` et `p`.

Question 5 Complétez la méthode `equals`.

Question 6 Complétez la classe `Droite` : un droite sera représentée par une paire de points `p1` , `p2`.

Vous devrez créer des méthodes équivalentes à celles de la classe `Point` et les compléter. Vous expliquerez vos choix en ajoutant des commentaires à vos méthodes, en particulier celles de la méthode `equals` et de la méthode `milieu` dont vous définirez le comportement.

3–Héritage (25 minutes)

Les classes `Bicyclette` et `Tandem` vous sont données :

```
public class Bicyclette {
    public Bicyclette(){}
    public void freiner(){}
}

public class Tandem extends Bicyclette {
    public Tandem(){}
    public void ralentir(){}
}
```

A l'aide de ces deux classes, trouvez et expliquez les erreurs contenues dans chacune des 6 portions de code suivantes. (Si possible, précisez quand ces erreurs apparaissent : lors de la compilation, ou lors de l'exécution)

(1)

```
Bicyclette bc = new Tandem();
bc.freiner();
bc. ralentir();
```

(2)

```
Bicyclette bc = new Tandem();
Tandem td = new Bicyclette();
```

(3)

```
Bicyclette bc = new Tandem();
Tandem td = (Tandem) new Bicyclette();
td. ralentir();
```

(4)

```
Tandem td = new Tandem();
Bicyclette bc = td;
Tandem td2 = bc;
```

(5)

```
Bicyclette bc = new Tandem();
Tandem td = (Bicyclette) bc;
```

(6)

```
Bicyclette bc = new Tandem();
Tandem td = (Tandem) new Bicyclette();
bc. ralentir();
td. ralentir();
```

4–Tri d'un tableau (35 minutes)

Vous allez réaliser une méthode de tri d'un tableau d'entiers. On rappelle que la taille d'un tableau `t` est donnée par sa variable d'instance `t.length`

Question 1 Proposez une méthode `int max(int t[])` permettant de trouver la valeur maximale du tableau `t` passé en paramètre.

Question 2 Proposez une méthode `int maxIndex(int t[])` qui retourne la position (un entier entre 0 et `t.length - 1`) de la valeur maximale du tableau `t`.

Remarque : si la valeur maximale apparaît plus d'une fois dans le tableau, vous ne considérerez que la position de sa première occurrence.

Question 3 En vous aidant éventuellement de la méthode de la question 2, proposer une méthode `void trier(int t[])` permettant de trier le tableau `t`.