

- Documents et calculatrices interdits
- Barème donné à titre indicatif
- Répondez sur le sujet quand c'est possible, sur la copie dans le cas contraire

**Ne pas consulter le sujet avant le signal
de début de l'épreuve**

Tous les stylos doivent être posés à la fin de l'épreuve

Exercice I (4 points)

Cochez les cases correspondant à vos réponses. Chaque bonne réponse vaut 0,5 point, chaque mauvaise réponse vaut -0,5 point. En cas d'absence de réponse, 0 point.

1. Une variable de type **int** est stockée sur 6 octets.

☐ Vrai ☐ Faux

2. **int** n = 2 ;
 int m = n ;
 n++ ;

m vaut 3 après ces instructions.

☐ Vrai ☐ Faux

3. La concaténation de String crée de nouveaux objets.

☐ Vrai ☐ Faux

4. Une classe abstraite peut implémenter une interface.

☐ Vrai ☐ Faux

5. Une classe peut implémenter plusieurs interfaces.

☐ Vrai ☐ Faux

6. **public class A {**

```
public void f(){  
    System.out.println("A");  
}
```

```
public class B extends A{  
    public void f(){  
        System.out.println("B");  
    }  
}
```

On crée la variable A obj = **new** B();. L'instruction obj.f() ; affiche "A".

☐ Vrai ☐ Faux

7. Une classe membre statique peut accéder à tous les membres de la classe englobante.

☐ Vrai ☐ Faux

8. Avec Junit 5, le tag @Before permet d'indiquer qu'une méthode doit être exécutée avant chaque test unitaire.

☐ Vrai ☐ Faux

Exercice II (2 points)

- (1) 1. Écrivez une méthode qui prend en entrée une liste d'Integer et retourne la liste des éléments d'indices pairs de la première liste. Par exemple, si la liste donnée en entrée contient les Integer [3,21,1,12,17] , la méthode retourne la liste [3,1,17] .

- (1) 2. Écrivez une méthode qui prend en entrée une liste d'Integer et retourne l'indice du plus grand élément de la liste. Par exemple, si la liste donnée en entrée contient les Integer

[3,21,1,12,17] , la méthode retourne 1 (l'indice du nombre 21).



Exercice III (2 points)

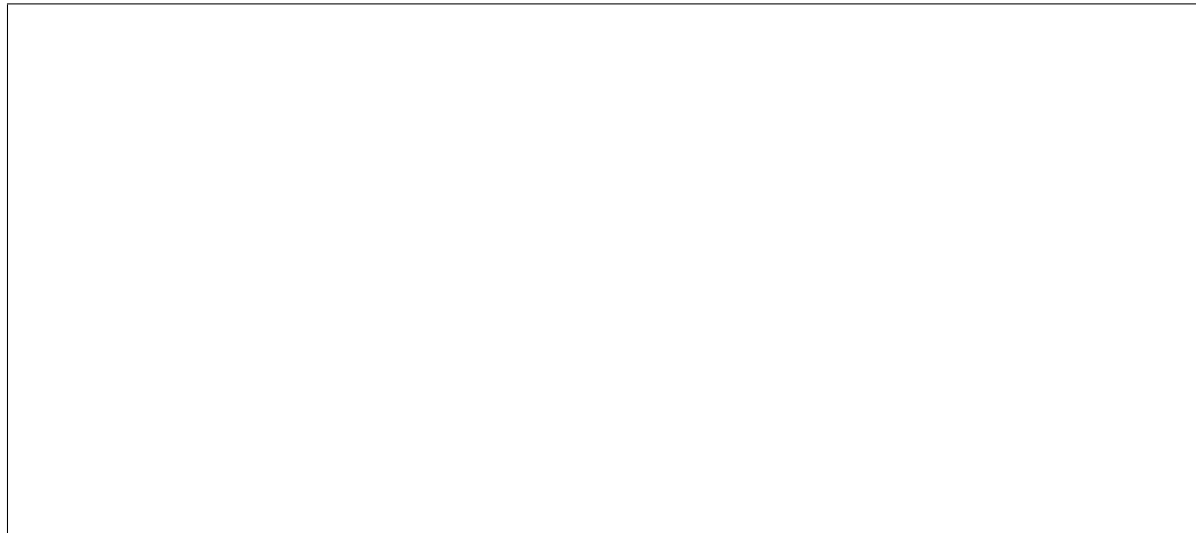
On considère la classe suivante (fournie sans explication). Ré-écrivez cette classe pour éviter les répétitions de code. Répondez sur la copie d'examen.

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        File file = new File("donnees.txt");  
        Lecture simple = new AffichageSimple();  
        try {  
            simple.ecrireEntiers(file);  
        } catch (IOException e) {  
            System.out.println("Traitement non effectue");  
        }  
        Lecture doublee = new AffichageDoublee();  
        try {  
            doublee.ecrireEntiers(file);  
        } catch (IOException e) {  
            System.out.println("Traitement non effectue");  
        }  
    }  
}
```

Exercice IV (2 points)

Voici (une partie de) la définition de la classe Coordonnees :

```
public class Coordonnees {  
    private double x ;  
    private double y ;  
  
    public Coordonnees(double x, double y){  
        this.x = x ;  
        this.y = y ;  
    }  
}
```



```
public static void main(String [] args){  
    System.out.println(new Coordonnees(6,7.8));  
}  
} // Fin de la classe Coordonnees
```

Complétez la définition de la classe Coordonnees pour que l'exécution de la fonction main donne exactement l'affichage suivant :

[6.0, 7.8]

Exercice V (8,5 points)

Pour les questions suivantes, répondez sur la copie d'examen.

- (1½) 1. Un dictionnaire est l'association d'un mot à ses différentes définitions. Définissez une classe Dictionnaire qui permet de représenter cela.
- (1½) 2. Dans la classe Dictionnaire, définissez une méthode qui permet d'ajouter une définition à un mot du dictionnaire : **public void** ajouterDef(String mot, String def)
- (1½) 3. Dans la classe Dictionnaire, définissez une méthode qui permet d'obtenir sous forme de String les différentes définitions d'un mot s'il apparaît dans le dictionnaire, ou un message d'erreur ("Le mot XXX n'est pas dans le dictionnaire.", avec XXX le mot en question) dans le cas contraire : **public** String chercher(String mot)
- (2) 4. On souhaite pouvoir afficher l'ensemble des mots d'un Dictionnaire avec leur définition, de la manière suivante : un mot du dictionnaire et ses définitions apparaissent sur une ligne, avec un nombre devant chaque définition (1 devant la première, 2 devant la deuxième,...). Un retour à la ligne sépare différents mots. Définissez une méthode qui permet d'obtenir une telle chaîne de caractère à partir d'un Dictionnaire.
- (2) 5. La question précédente permet d'afficher l'ensemble des mots d'un dictionnaire, mais sans respecter l'ordre alphabétique. Définissez une classe DictionnaireAlphabetique qui hérite de Dictionnaire et qui garantit que les mots seront affichés dans l'ordre alphabétique.

Exercice VI (1,5 points)

La classe Etudiant est définie comme suit :

```

public class Etudiant {
    private Map<String,Note> notes ;
    private String nom ;

    public Etudiant(String nom){
        this.nom = nom ;
        notes = new HashMap<String,Note>();
    }

    public void ajoutNote(String matiere, double note, int coef){
        if (!notes.containsKey(matiere)){
            notes.put(matiere, new Note(note, coef));
        }
    }
}

```

et la classe Note comme suit :

```

public class Note{
    private double note ;
    private int coef ;

    public Note(double note, int coef){
        this.note = note ;
        this.coef = coef ;
    }

    public double getNote(){
        return coef ;
    }

    public int getCoef(){
        return coef ;
    }
}

```

Définissez une méthode **public double** getMoyennePonderee(){ ... } dans la classe Etudiant qui permet d'obtenir la moyenne d'un étudiant en tenant compte des coefficients des notes.