

- 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 (2 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. Il faut appeler la méthode `run()` de la classe `Thread` pour créer et exécuter un thread.  
☐ Vrai      ☐ Faux
2. Il n'est pas recommandé d'appeler la méthode `stop()` de la classe `Thread`.  
☐ Vrai      ☐ Faux
3. Le *design pattern* Singleton sert à garantir qu'une classe n'est instanciée qu'une fois.  
☐ Vrai      ☐ Faux
4. La classe `Pane` de JavaFX permet de placer des composants graphiques à des coordonnées précises.  
☐ Vrai      ☐ Faux

### Exercice II (2 points)

- (1) 1. À quoi sert le mot-clé **synchronized**?

- (1) 2. Quel est le rôle du *design pattern* Observer?

### Exercice III (6 points)

Voici (une partie de) la définition de la dictionnaire `Dictionnaire` :

```
/**
 * Le dictionnaire est represente par une liste de mots et une liste
 * de definitions, telles que le i-eme mot correspond
 * a la i-eme definition. On suppose qu'il n'y a pas de doublons.
 */
public class Dictionnaire {
    private List<String> mots;
    private List<String> definitions ;

    public Dictionnaire(){
        mots = new ArrayList<String>();
        definitions = new ArrayList<String>();
    }

    public void ajouterMot(String mot, String definition){
        mots.add(mot);
        definitions.add(definition);
    }
}
```

```

public int getNbMots(){
    return mots.size();
}

public String getMot(int i){
    return mots.get(i);
}

public String getDefinition(int i){
    return definitions.get(i);
}

```

On souhaite pouvoir représenter un dictionnaire par un fichier texte où chaque ligne associe un mot et sa définition. Par exemple :

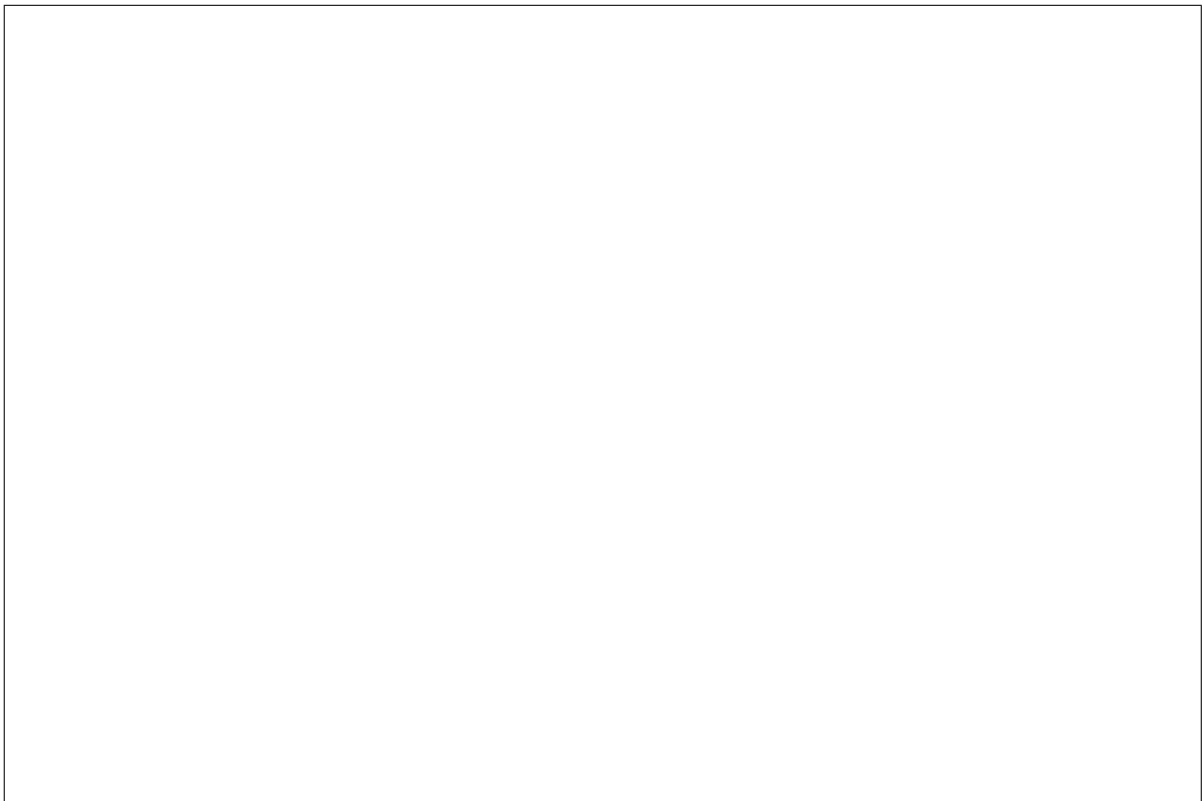
Java : langage de programmation orienté objet

Université : institution d'enseignement supérieur et de recherche

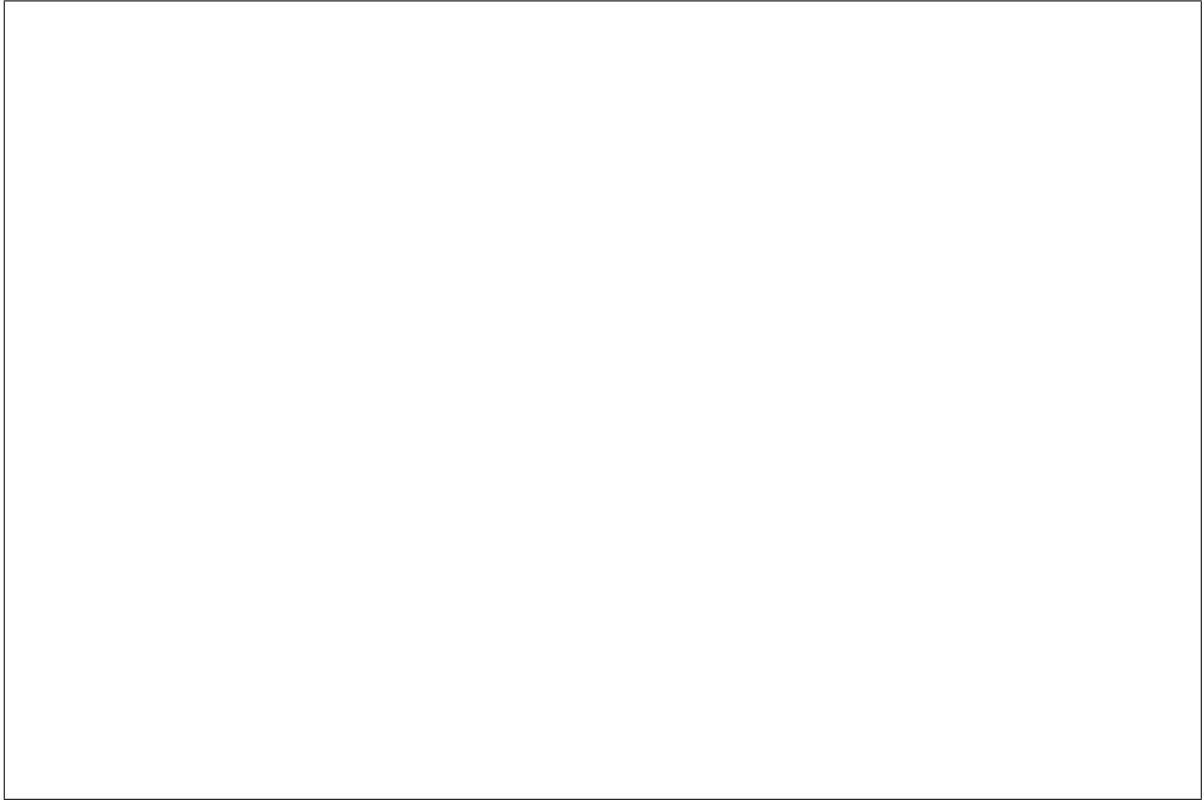
Étudiant : personne engagée dans un cursus universitaire

Examen : évaluation qui permet de déterminer si un étudiant a le niveau requis

- (3) 1. Écrivez une méthode qui lit un fichier texte et retourne le Dictionnaire correspondant. On suppose que le fichier texte est correctement formaté.



- (3) 2. Écrivez une méthode qui prend en paramètre un Dictionnaire et enregistre son contenu dans un fichier texte.



#### Exercice IV (6 points)

On peut représenter une opération mathématique sous forme d'un arbre dont les noeuds sont des opérateurs, et les feuilles sont des valeurs numériques. On souhaite représenter de telles expressions et pouvoir évaluer leur valeur.

- La classe (abstraite) `Operateur` représente aussi bien un noeud interne qu'une feuille de l'arbre.
- Les opérateurs arithmétiques habituels (Addition, Soustraction, Multiplication, Division), d'arité 2, sont des classes filles d'`Operateur`.
- Une `Valeur` a une arité nulle, et représente un nombre réel.

La classe `Operateur` contient une méthode abstraite **public abstract double** `evaluer()` ; qui retourne la valeur de cet opérateur pour ses opérandes.

- (3) 1. Implémentez ces classes.



(1) 2. Quel est le *design pattern* utilisé ici?

A rectangular box with a thin black border, intended for the user to write the answer to question 2.

(2) 3. Donnez des tests unitaires pour la méthode `evaluer()` de la classe `Addition`.

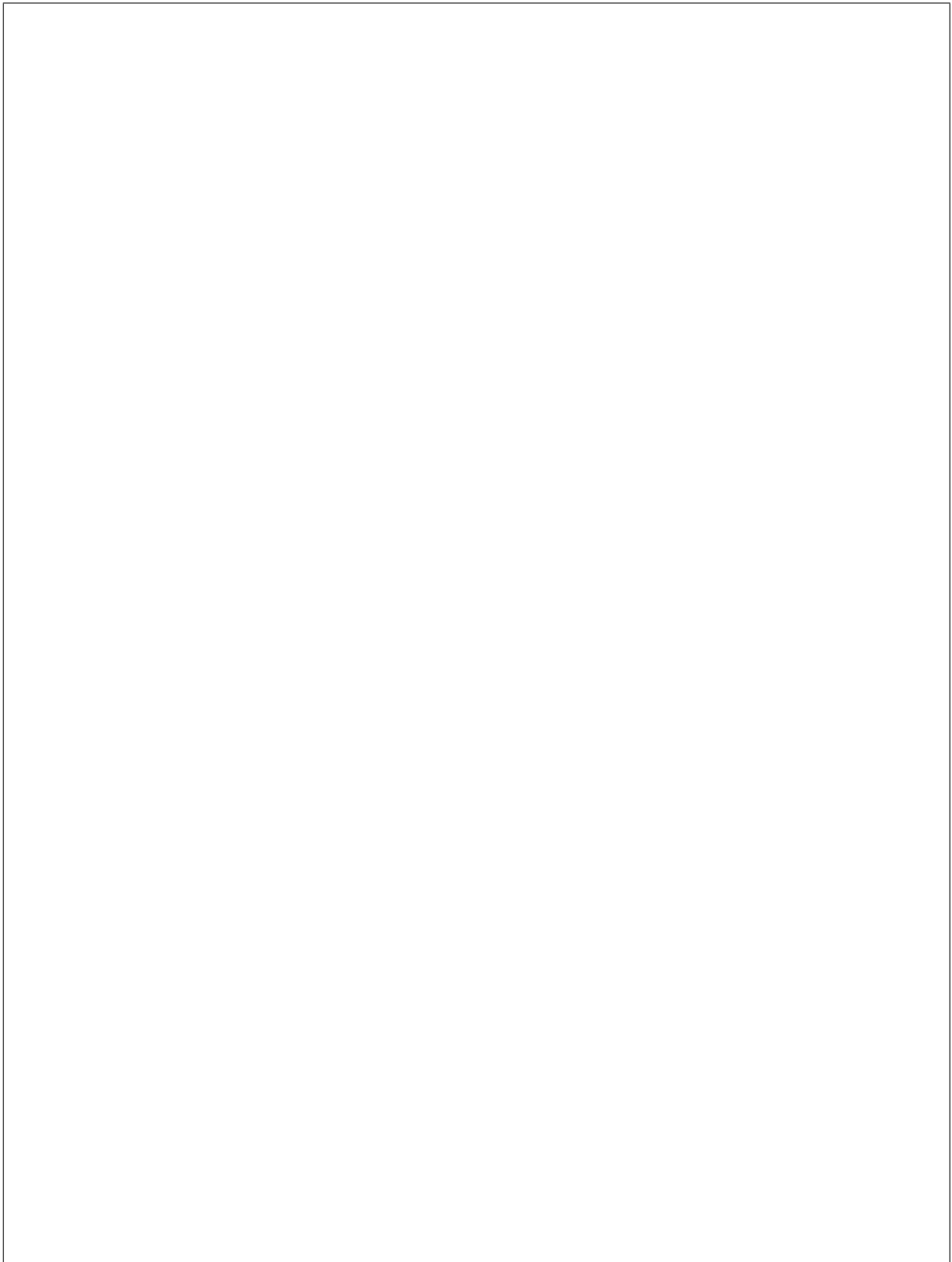


### Exercice V (4 points)

- (2) 1. Donnez le code Java qui permet de produire l'interface JavaFX suivante :



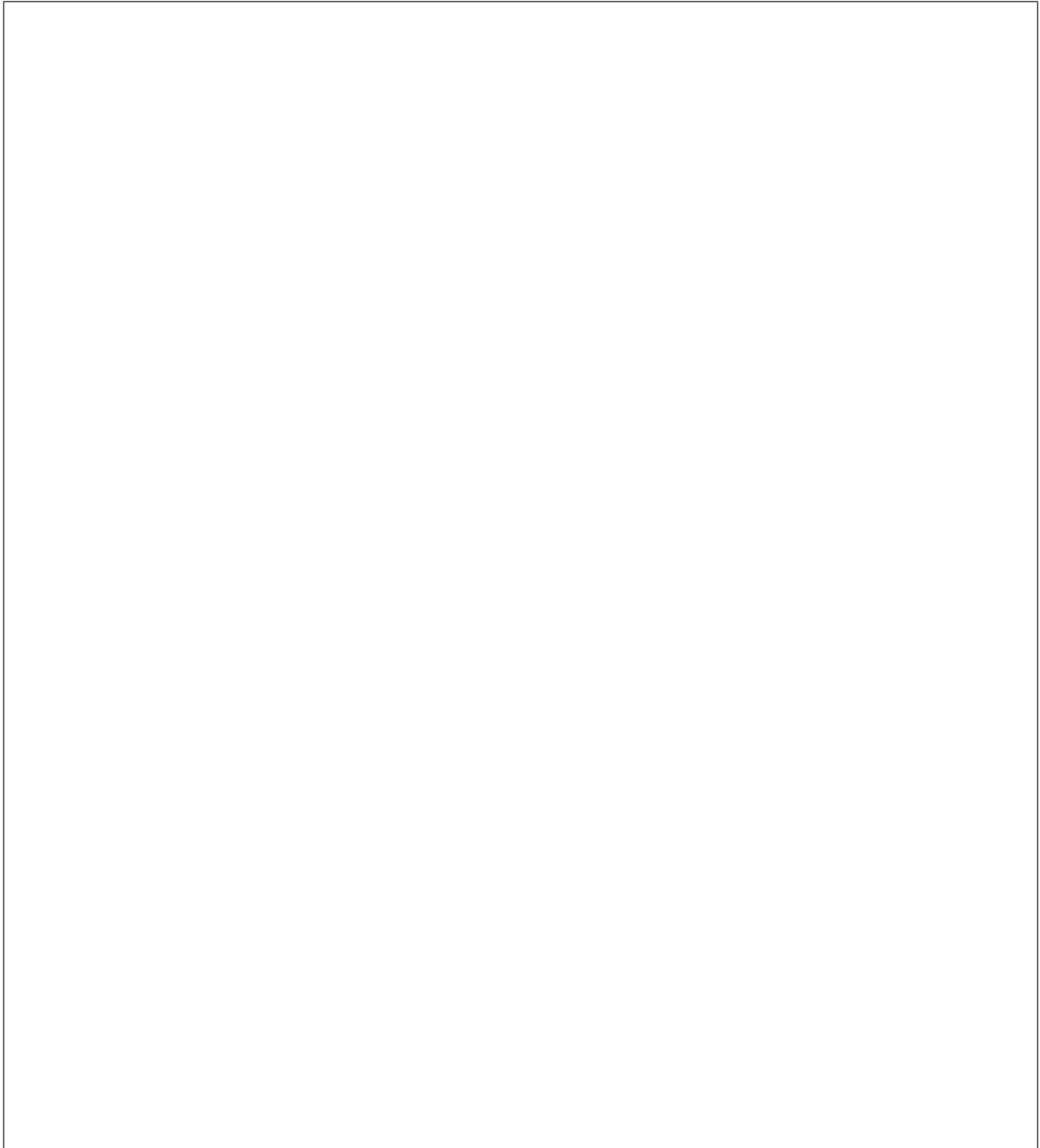
**Remarque :** La zone au dessus du bouton "Reset" est un label (qui est vide lors de l'initialisation de la fenêtre). Par contre, il n'y a rien entre le bouton "+" et le bouton "-", ni entre le bouton "-" et le bouton "Reset".



(2) 2. Indiquez les modifications à apporter au code précédent pour gérer les événements :

- lors d'un click sur le bouton "+", les valeurs présentes dans les deux champs de texte sont additionnées et affichées dans le label.
- lors d'un click sur le bouton "-", les valeurs présentes dans les deux champs de texte sont soustraites et affichées dans le label.
- lors d'un click sur le bouton "Reset", le contenu du label est effacé.

Pour cet exercice, on suppose que les valeurs présentes dans les champs de texte sont bien des nombres (il n'est pas nécessaire de le vérifier, ou de gérer les erreurs de l'utilisateur).

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for the user to provide their solution or work for the exercise.