

## Programmation par Contrat

Le but de ce TD est d'entrevoir la programmation par contrat.  
Syntaxe des assertions :

```
assert expressionVrai;  
assert expressionVrai : message à afficher;
```

Pour activer les assertion utiliser :

```
java -enableassertions  
ou  
java -ea
```

### Exercice 1

#### Expression bien parenthésée.

Étant donné une chaîne qui peut contenir les caractères '(', ')', '{', '}', '[' et ']', déterminez si une chaîne passée en entrée est valide (bien parenthésée).

Une chaîne d'entrée est valide si :

1. Les ouvertures '(', '[' et '{' sont fermées par le même type de supports.
2. Les ouvertures '(', '[' et '{' doivent être fermés dans le bon ordre.

Notez qu'une chaîne vide est considérée comme valide.

**Question 1** • Donner les pré-conditions et les post-conditions du programme "Parenthese".

**Question 2** • Écrire le contrat à remplir (pré et post-conditions) en Java. en Java le programme "Parenthese".

**Question 3** • Remplir le contrat en développant le programme "Parenthese" en Java.

**Question 4** • Ajouter des invariants sous forme d'assertions dans votre programme.

**Question 5** • Lancer l'exécution en mode *"enable assertions"*.

**Question 6** • Écrire quelques tests unitaires.

**Question 7** • Déboguer votre programme si besoin.

## Exercice 2

### Le plus long préfixe commun.

On souhaite développer une fonction qui trouve le préfixe commun le plus long parmi un tableau de chaînes. Si aucun préfixe de commun alors on retourne une chaîne vide "".

- Question 1** • Donner les pré-conditions et les post-conditions du programme "Prefix".
- Question 2** • Écrire le contrat à remplir (pré et post-conditions) en Java. en Java le programme "Prefix".
- Question 3** • Remplir le contrat en développant le programme "Prefix" en Java.
- Question 4** • Ajouter des invariants sous forme d'assertions dans votre programme.
- Question 5** • Lancer l'exécution en mode *"enable assertions"*.
- Question 6** • Écrire quelques tests unitaires.
- Question 7** • Déboguer votre programme si besoin.

## Exercice 3

### Carrés d'un tableau trié.

Étant donné un tableau d'entiers A trié dans un ordre croissant, on veut développer un programme qui retourne un tableau des carrés de chaque nombre, également dans un ordre croissant.

Par exemple :

- **Input** : [-4,-1,0,3,10]
- **Output** : [0,1,9,16,100]

- Question 1** • Donner les pré-conditions et les post-conditions du programme "sortedSquares".
- Question 2** • Écrire le contrat à remplir (pré et post-conditions) en Java. en Java le programme "sortedSquares".
- Question 3** • Remplir le contrat en développant le programme "sortedSquares" en Java.
- Question 4** • Ajouter des invariants sous forme d'assertions dans votre programme.
- Question 5** • Lancer l'exécution en mode *"enable assertions"*.
- Question 6** • Écrire quelques tests unitaires.
- Question 7** • Déboguer votre programme si besoin.