

## Esercitazione 2: classi e oggetti

**Esercizio 1.** Individuare la classe `String` nella documentazione della libreria standard di Java e leggerne la descrizione, con particolare attenzione ai metodi `charAt`, `concat`, `length`, `substring` (la versione con due parametri). Facendo uso di tali metodi, implementare:

1. un metodo statico `longest` che ritorni la stringa più lunga in un array di stringhe;
2. un metodo statico `concatAll` che ritorni la concatenazione di tutte le stringhe in un array;
3. un metodo statico `trim` che rimuova tutti gli spazi iniziali e finali di una stringa. Per esempio, l'invocazione `trim(" Elimina gli spazi iniziali e finali ")` deve ritornare la stringa `"Elimina gli spazi iniziali e finali"` dove  indica uno spazio. Assicurarsi che `trim` funzioni correttamente anche su stringhe composte esclusivamente da spazi. Per la risoluzione di questo esercizio è vietato fare uso del metodo `trim` della classe `String`.

Per ciascun metodo, valutare quali sono le pre-condizioni e specificarle con la clausola `assert`.

**NOTA:** le asserzioni non sono abilitate di default durante l'esecuzione dei programmi Java. Per abilitarle, occorre eseguire i programmi con il comando:

```
java -ea NomeClasse
```

In TextPad: modificare l'opzione Configure → Preferences → Tools → Run Java → Parameters impostandola a: `-ea $BaseName`

**Esercizio 2.** Implementare una versione della calcolatrice RPN che usi la classe `Stack` vista a lezione.

**Esercizio 3.** Progettare e implementare una classe `Point` per rappresentare **punti del piano** in virgola mobile. Dotare la classe dei seguenti metodi:

1. Il costruttore che inizializza l'oggetto date le coordinate come argomenti;
2. Due metodi per ottenere le coordinate del punto;
3. Un metodo per **traslare** il punto di  $(d_x, d_y)$ ;
4. Un metodo per **ruotare** il punto di un angolo  $\theta$  rispetto all'origine del piano;
5. Un metodo per calcolare la **distanza** del punto da un altro punto dato come parametro (come verificare che il secondo punto, passato come argomento, sia valido?).

Risolvere l'esercizio due volte, una prima volta in modo tale che il punto sia **mutabile** (le operazioni di traslazione e rotazione modificano lo stato del punto su cui sono invocate), la seconda volta in modo tale che il punto sia **immutabile** (le operazioni ritornano un nuovo oggetto che rappresenta il punto risultante).