

# TD 1

## Approche orientée objet, Classes, Objets

### EXERCICE #1 ► Chronomètre

La figure 1.1 propose un diagramme de classe Chronomètre.

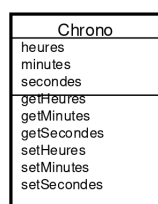


Figure 1.1: Classe chronomètre

- Complétez les signatures (nom, type de retour, nombre et type d'arguments) des méthodes définies dans les diagrammes de classe ci-dessus.
- Écrire en Java les définitions de ces classes (code). A la création de l'objet, le temps doit être initialisé à 00:00:00 (définir un constructeur approprié).
- Écrire une nouvelle méthode rebours permettant de soustraire une seconde au chronomètre.
- En utilisant cette classe, écrire un programme Java affichant les valeurs successives d'un compte à rebours à partir de 01:30:00 et se terminant par l'affichage de "Fin du cours!"

### EXERCICE #2 ► Une classe fraction

L'objectif de cet exercice est de concevoir une classe Fraction qui implémente un nouveau type de données "fraction" ( $\frac{1}{3}$ , par exemple).

- Pourquoi à votre avis une telle une classe Fraction peut être utile ?
- Écrire le diagramme d'une classe Fraction (dénominateur et numérateur entiers). Le diagramme devra spécifier attributs, accesseurs et mutateurs, ainsi que les signatures complètes des méthodes suivantes :
  - add : ajoute un entier x à la fraction
  - sub : soustrait un entier x à la fraction
  - mult : multiplie par un entier x la fraction
  - réduire : réduit la fraction
  - addFraction : ajoute une fraction f à la fraction
  - multFraction : multiplication de fractions
- Implémenter un constructeur à deux paramètres pour la classe Fraction, ainsi que les méthodes ci-dessus. On supposera que la classe Fraction possède une méthode `int pgcd(int x, int y)`.
- En utilisant la classe Fraction, écrire un programme Java (classe + méthode main) effectuant le calcul suivant :  $(123/456 + 789/10 + 11) * 12$  et réduisant le résultat.
- **Question bonus** : Implémenter la méthode pgcd en Java.

**EXERCICE #3 ► Des rectangles**

Source : O. Carton (IRIF) On suppose une classe `Point` fournie.

- Définir une classe `Rectangle` permettant de modéliser des rectangles. Définir trois constructeurs prenant respectivement en paramètre 2 points, 1 points et 2 longueurs ou 4 longueurs.
- Écrire une méthode `surface` qui calcule la surface du rectangle.
- Écrire une méthode `contains` qui teste si un point donné (en paramètre) est à l'intérieur du rectangle. Écrire une autre méthode `contains` qui teste si un rectangle donné est à l'intérieur du rectangle.
- Écrire une méthode `sameAs` qui teste l'égalité de deux rectangles.
- Ajouter à la classe `Rectangle` un attribut `nbr` qui comptabilise le nombre d'objets de la classe `Rectangle`.
- Écrire une méthode `hull` qui calcule le rectangle englobant d'un ensemble de rectangles. Cette méthode prend en paramètre un tableau de rectangles et retourne le plus petit rectangle qui contient tous les rectangles du tableau.