

## TP02

Matière : ATELIER DE PROGRAMMATION OBJET

Classes : DSI2 – RSI2 – SEM2

L'objectif de ce TP est d'appliquer les concepts fondamentaux de la programmation objet pour modéliser un ensemble de classes et de les instancier. La modélisation consiste à identifier les attributs d'une classe et de choisir leurs types et leurs visibilités adéquats, à identifier et implémenter les constructeurs, les getters, les setters et les autres méthodes. Pour chaque classe, une classe de test qui contient le main permet d'instancier des objets et d'appeler les différentes méthodes.

**Exercice 1**

Un Stylo est défini par sa marque, sa couleur et son niveau d'encre (0..100). Les opérations que permet un Stylo sont :

- afficher : permet d'afficher les caractéristiques du Stylo,
- écrire : prend en paramètre une chaîne de caractère et l'affiche si le niveau d'encre est non nul, dans ce cas le niveau d'encre diminue d'une unité,
- charger : permet de remettre le niveau d'encre à 100.

- 1- Identifier les attributs de la classe Stylo et donner le type et la visibilité adéquats de chaque attribut.
- 2- Implémenter quatre constructeurs de la classe Stylo :
  - a- Le 1<sup>er</sup> ne prend aucun paramètre, dans ce cas la marque sera « BIC », la couleur sera « Bleu » et le niveau d'encre sera 100,
  - b- Le 2<sup>ème</sup> prend en paramètre la marque,
  - c- Le 3<sup>ème</sup> prend en paramètre la marque et la couleur.
  - d- Le 4<sup>ème</sup> prend en paramètre un Stylo.
- 3- Implémenter les trois méthodes de la classe Stylo.
- 4- Implémenter une classe TestStylo qui contient le main () et qui permet de tester les différentes méthodes de la classe Stylo. En particulier utiliser les quatre constructeurs de la classe,
- 5- Ajouter deux méthodes getMarque pour obtenir la marque et getCouleur pour obtenir la couleur,
- 6- Appeler ces deux méthodes à partir du main ().

**Exercice 2**

Un Rectangle est défini par sa longueur, sa largeur et le caractère qui est utilisé lors de son affichage (\* - .). Les opérations que permet un Rectangle sont :

- getSurface : permet de retourner la surface du Rectangle,
- getPerimetre : permet de retourner le périmètre du Rectangle,
- agrandir : prend en paramètre un entier et permet d'agrandir la longueur et la largeur en les multipliant par ce paramètre,
- setCaractere : prend en paramètre un caractère qui sera le caractère utilisé pour l'affichage du Rectangle,
- afficher : permet d'afficher le Rectangle, exemple pour (5,2) le rectangle à afficher est :

\*\*\*\*\*

\*\*\*\*\*

- 1- Identifier les attributs de la classe Rectangle et donner le type et la visibilité adéquats de chaque attribut.
- 2- Implémenter quatre constructeurs de la classe
- 3- Implémenter les cinq méthodes de la classe Rectangle.
- 4- Implémenter une classe TestRectangle qui contient le main () et qui permet de tester les différentes méthodes de la classe Rectangle.

### Exercice 3

Un compteur cyclique est défini par sa valeur et par son cycle (10 par défaut). Il possède aussi un pas qui est la valeur à ajouter à chaque incrémentation, par défaut le pas est 1. Les opérations qu'on peut effectuer sur un tel compteur sont : la lecture, le changement et l'affichage de sa valeur, la lecture du cycle cependant le changement du cycle n'est pas autorisé, la lecture et le changement du pas et l'incrémentation du compteur.

- 1- Implémenter la classe compteur et tester la par une classe TestCompteur,
- 2- On veut modifier la méthode Incrémenter pour qu'elle retourne vrai si un cycle est achevé et faux sinon, donner la nouvelle implémentation de cette méthode.

### Exercice 4

Un Ascenseur permet aux habitants d'un immeuble de se déplacer entre un certain nombre d'étages. L'Ascenseur possède une capacité qui est le poids maximum de la charge qu'il peut supporter. De plus l'Ascenseur est défini par son état (Ouvert/Fermé), par l'étage où il se trouve et par le poids de la charge qu'il contient.

L'Ascenseur possède deux boutons ouvrir et fermer qui permettent de changer l'état de l'Ascenseur. Lorsqu'une personne monte ou descend de l'Ascenseur son poids est ajouté ou retranché du poids de la charge. De plus l'Ascenseur peut faire descendre toutes les personnes, dans ce cas le poids de sa charge passe à zéro.

Lorsqu'une personne est dans l'Ascenseur, elle peut demander d'atteindre un étage cible. L'Ascenseur exécute cette demande et retourne vrai si son état est fermé, le poids de sa charge est inférieur à sa capacité et l'étage cible est différent de l'étage en cours dans le cas contraire la méthode retourne faux. Si la méthode est bien exécutée l'Ascenseur change d'étage et affiche les étages parcourus.

- 1- Identifier les attributs de la classe Ascenseur et donner le type adéquat de chaque attribut.
- 2- Implémenter trois constructeurs de la classe Ascenseur.
- 3- Implémenter une méthode afficher qui affiche les différents attributs de l'ascenseur,
- 4- Implémenter les autres méthodes de la classe Ascenseur.
- 5- Modifier la méthode atteindre pour qu'elle ferme les portes s'ils sont ouverts et exécute la demande de la personne,
- 6- Implémenter une classe TestAscenseur qui contient le main () et qui permet de tester les différentes méthodes de la classe Ascenseur.

