

TD3 POO / Java : une classe en tant que type : cas de la composition ; polymorphisme

Objectif : Manipuler des classes en tant que type.

Matériel / Logiciel : Environnement de développement sur PC (sous Linux) : JDK1.8 (ou supérieur) : compilateur java (javac) et JVM (java) IDE Geany / éditeur gedit et console / Eclipse.

Acquisition : différence entre classe et objet. Composition. Mécanisme du polymorphisme dynamique.

1 Finalisation du TD 2

Important : pour cette séance et pour la suite, l'utilisation de l'IDE¹ Eclipse est recommandée.

Reprendre les classes relatives au problème du compteur et apporter les améliorations utiles à vos codes : méthode toString(), non redondance du code, accesseurs et mutateurs, usage de « this. » et « this() ».

2 La classe Contact

On considère dans cet exercice, les « contacts » que l'on trouve par exemple dans un répertoire téléphonique d'un téléphone portable. On se limitera aux 3 informations suivantes : le nom du contact, son numéro de téléphone et son adresse mail.

Remarque : en terme de fonctionnalités, il s'agit d'une classe de stockage, à l'image de la classe « Etudiant » traitée lors de la première séance.

Définir le diagramme de classe UML de ce problème.

Écrire les classes « Contact » et « TestContact ».

3 La classe Segment

Reprendre l'exemple du CM et réaliser une classe « Segment » composée de 2 points. Vous prévoirez en particulier la méthode qui permet de translater (déplacer) ce Segment.

Définir le modèle UML d'une telle classe. Vous prévoirez en particulier un constructeur prenant en paramètre les coordonnées des 2 points du Segment (4 valeurs : xa, ya, xb, yb).

Écrire les classes « Segment » et « TestSegment ».

Ajouter une méthode getLongueur() à votre classe et tester cette nouvelle fonctionnalité.

4 La classe Chronometre

On souhaite concevoir puis implémenter une classe « Chronometre » dont on simulera le fonctionnement sans tenir compte du temps réel. On retiendra le format « HH:MM:SS » pour le chronomètre qui pourra être incrémenté ou décrétementé par pas de 1.

Définir le diagramme de classe UML de ce problème.

Écrire les classes « Chronometre » et « TestChronometre ».

Important : apportez les améliorations de codage nécessaire à cette classe en respectant en particulier les conventions de nommage.

5 Polymorphisme dynamique

Vérifier l'exemple du CM du mécanisme du polymorphisme avec une référence de type Point et un objet de type PointColore.

¹ IDE : Integrated Development Environment.