

L'objectif des exercices ci-dessous est double :

- a) redéfinir les méthodes de la classe **Object** pour la comparaison et la duplication d'objets,
- b) démarrer le développement du package **Banque**.

Ces travaux doivent être exécutés sur les postes de travail du Département Informatique. L'IDE est pré installé dans l'espace des logiciels de développement.

Exercice 1. Finalisation de la feuille d'exercices de la semaine 2

Terminer les exercices de la semaine précédente.

Conserver dans un fichier texte le résultat de l'exécution du test de la classe **Point** (**T_Point**).

Exercice 2. Méthodes héritées de Object pour la classe Point

Introduire la redéfinition des méthodes *equals* et *clone* dans la classe **Point**.

Compléter la classe **T_Point** pour y introduire une mise en œuvre de ces méthodes.

NB : on pourra utiliser la méthode « confondus » pour implémenter la méthode « equals ».

Exercice 3. Modification des méthodes de la classe Point

Modifier **éventuellement** les autres méthodes de la classe en utilisant *equals* ou *clone* à chaque fois que l'on a besoin de comparer ou de dupliquer des objets.

Exécuter la classe **T_Point** pour vérifier que le résultat obtenu est bien le même que celui obtenu dans l'exercice

Exercice 4. Méthodes héritées de Object pour la classe Segment

Procéder de même (en 2 étapes) pour la classe **Segment**.

Exercice 5. Package Banque

Nous allons maintenant démarrer le développement d'un ensemble de classes pour gérer une banque.

Lire attentivement le document fourni en annexe afin de bien comprendre l'objectif attendu.

Exercice 6. Création de l'arborescence pour le package Banque

Dans votre répertoire de P : dédié au module M213, créer un sous répertoire **Banque** propre à ce package (indépendant des TD). On créera dans ce répertoire un sous-répertoire par classe et par la suite on pourra gérer différentes versions d'une même classe dans des sous-répertoires de la classe.

Exercice 7. Développement de la première classe Titulaire

Lire attentivement le document fourni en annexe qui donne la description détaillée de la classe à implémenter.

Pour développer cette classe, vous aurez besoin de vous documenter sur la classe **String**.

Développer cette classe **Titulaire** et la classe de test associée **T_Titulaire** en parallèle et de façon incrémentale.

Exercice 8. Développement de la classe SBanque

Les différents contrôles à faire sur la validité de l'identité d'un titulaire ou de sa date de naissance peuvent être regroupés dans une classe dédiée à la fourniture de services pour les autres classes. Cette classe dédiée aux services peut aussi servir à regrouper tous les paramètres de l'application. Une telle classe doit être abstraite (ne peut être instanciée) et ne contenir que des méthodes de classe (static).

Développer cette classe contenant les services *identiteValide*, *dateValide* et *dateJour*, qui retournent un booléen pour les 2 premiers et une chaîne de caractères pour le 3^e. Pour cette dernière méthode, vous aurez besoin de vous documenter sur la gestion du temps et des dates en Java.

Développer cette classe **SBanque** et la classe de test associée **T_SBanque** en parallèle et de façon incrémentale.

Modifier la classe **Titulaire** pour faire appel à ces services, une fois validés.