

FICHE 1 : DES CLASSES ET DES OBJETS

objectifs

- Comprendre la notion de **classes** et d'**objets**.
- Manipuler l'**état** et le **comportement** d'une classe.
- Comprendre ce qu'est un **constructeur** dans une classe et comment l'invoquer (**new**).
- Savoir écrire une classe avec des **attributs** et des **méthodes** en Java.

Vocabulaire

classe	objet	état	comportement
attributs	méthodes	new	instanciation
instance	responsabilité		

Consigne

Créez un répertoire APOO dans lequel se trouveront tous les projets IntelliJ créés pour le cours d'APOO. Si vous travaillez sur les ordinateurs de l'école, créez ce répertoire directement sur le U: et n'oubliez pas de récupérer vos projets à la fin de la séance (transférez-les sur votre onedrive ou ...) afin de pouvoir continuer à travailler chez vous. Pour la fiche de cette semaine, créez, dans le répertoire APOO, un projet IntelliJ nommé **APOO_fiche1** dans lequel vous allez mettre toutes les classes demandées dans les exercices de cette fiche.

Exercices

1. Des groupes de musique

Voici la description d'un mini mOOnde. On vous demande de représenter la classe `GroupeDeMusique` en UML ainsi que les instances ci-dessous.

Il existe un certain nombre de groupes de musique dans le mini monde:

- Hooverphonic : formé en 1995 , belge, trip-hop
- Portishead : formé en 1991, anglais, trip-hop
- Garbage : formé en 1993, international, rock
- Texas : formé en 1985, écossais, rock

2. Cercles

Utilisez IntelliJ afin d'implémenter les classes `Cercle` et `TestCercle` vues au cours théorique. Exécutez la classe `TestCercle`.

3. Rectangles

Un rectangle possède une longueur et une largeur et doit être capable de donner son aire et son périmètre.

- Dessinez le diagramme UML de la classe `Rectangle`. **Faites-le valider par le professeur !**
- Implémentez la classe `Rectangle` en java.
- Écrivez la classe `TestRectangle` dans laquelle vous devez créer 3 instances de `Rectangle`, le premier de longueur 5 cm et de largeur 3 cm, le deuxième de

longueur 10 cm et de largeur 6 cm et le troisième de longueur 8 cm et de largeur 2 cm. Ensuite le test affichera l'aire du premier rectangle créé, le périmètre du second rectangle créé et l'état du troisième.

4. Des livres

Pour un livre, nous avons besoin de garder les informations suivantes :

- son ISBN
- son titre ;
- le nom et le prénom de son auteur ;
- le prix conseillé ;
- son année d'édition ;
- son nombre de page ;

Sachant cela, on vous demande de :

- Réaliser le diagramme UML correspondant à la classe `Livre`. Donnez également 2 instances de `Livre` en UML. **Faites valider par le professeur !**
- Implémentez la classe `Livre` en java.
- Écrivez la classe `TestLivre` dans laquelle vous devez créer les deux instances de `Livre`. Ensuite, la classe doit afficher le titre, l'année d'édition, le prénom et le nom de l'auteur du premier livre et toutes les informations du deuxième livre.

5. Des personnes

Voici une classe `Personne` qui se limite aux informations suivantes :

- Une personne a un nom et un prénom
- Une personne est née à une certaine date : jour, mois et année. (3 entiers)
- Une personne a une adresse composée de la rue, du numéro, du code postal et du nom de la ville.

Une personne peut faire un certain nombre de choses :

- Elle peut donner l'âge qu'elle aura à la fin de l'année actuelle.
- Elle peut donner l'âge qu'elle avait (qu'elle aura) au 31 décembre d'une certaine année. Par exemple, l'âge de Kevin au 31/12/2001.
- Elle peut également donner son adresse complète (c.-à-d. rue, numéro, CP et ville).
- Finalement, elle peut donner sa date de naissance complète.

Sachant cela, on vous demande de :

- Réaliser le diagramme UML correspondant à une `Personne`. **Faites valider par le professeur !**
- Implémentez la classe `Personne` en java.
- Enfin, écrivez une classe `TestPersonne` qui crée 3 objets de type `Personne`, affiche l'état de ces personnes et affiche également l'âge qu'elles auront à la fin de l'année et enfin l'âge de chacune d'elle fin 2025.

Bonus

- À quoi sert la méthode `toString` ?
- Ajoutez la méthode `toString` à vos classes et invoquez-la dans vos tests.