

Chapitre 4 : Objets et classes

- **Durée** : 6 heures 25
- **Mots-clés** : classe, objet, attribut, constructeur, méthode, création
- **Objectif** :

Dans ce chapitre, vous apprendrez à :

- ▶▶ déclarer une classe, les attributs et les méthodes ;
- ▶▶ créer des objets à partir d'une classe ;
- ▶▶ accéder aux attributs au sein des méthodes d'une classe ;
- ▶▶ appeler les méthodes ;
- ▶▶ utiliser les constructeurs ;
- ▶▶ réaliser des interactions entre deux objets.

Pré-requis

Pour valider les pré-requis nécessaires avant d'aborder le TP, répondez aux questions ci-après qui ne concernent que le langage Java :

1. Quel est le but d'une classe ?
2. Parmi les éléments suivants, que peut contenir la déclaration d'une classe ?
 - a. des attributs ;
 - b. une liste des autres classes avec lesquelles elle accepte de communiquer ;
 - c. des méthodes.
3. Comment les objets sont-ils créés à partir d'une classe ?
4. Au sein d'une méthode, quelles sont les variables accessibles ?
 - a. les attributs de la classe dans laquelle est déclarée la méthode ;
 - b. les variables locales de la méthode ;
 - c. les paramètres de la méthode ayant appelé cette méthode ;
 - d. Les variables locales de la méthode ayant appelé cette méthode ;
 - e. les paramètres de la méthode.

5. Parmi les éléments suivants, lequel est désigné par le mot-clé `this` ?
 - a. le programme principal ;
 - b. le dernier objet créé ;
 - c. l'objet qui a reçu l'appel de la méthode ;
 - d. une variable locale spécifiée dans le programme.
6. À quoi sert un constructeur ?
 - a. à initialiser un objet lors de sa création ;
 - b. à déterminer les règles d'interaction entre les objets d'une classe ;
 - c. à construire la classe.
7. Quand le constructeur d'une classe est-il invoqué ?
8. Combien est-il possible de définir de constructeurs dans une classe ?
9. Comment est réalisé l'appel d'une méthode ?
10. Parmi les éléments suivants, auxquels une classe peut-elle servir de type ?
 - a. à une variable définie dans le programme principal ;
 - b. à un attribut d'une classe ;
 - c. à un paramètre d'une méthode.
11. Quelle est la différence entre les modes d'encapsulation "public", "private" et "protected" ?

Énoncé 4.1 : La classe Rectangle

Durée estimative : 20 minutes

Écrivez en Java la classe `Rectangle` qui permet de représenter des rectangles en définissant leur hauteur et leur largeur (les rectangles sont horizontaux et ne vous préoccupez pas de leurs coordonnées).

Écrivez les méthodes suivantes :

- le constructeur permettant de donner la valeur initiale de la hauteur et de la largeur ;
- la méthode `setLargeur` pour modifier la largeur ;
- la méthode `setHauteur` pour modifier la hauteur ;
- la méthode `getLargeur` pour obtenir la largeur ;
- la méthode `getHauteur` pour obtenir la hauteur ;
- la méthode `surface` qui calcule la surface du rectangle ;
- la méthode `périmètre` qui calcule le périmètre du rectangle.

Écrivez ensuite un programme principal qui crée un objet de la classe `Rectangle` après avoir préalablement lu au clavier sa largeur et sa hauteur. Ensuite, le programme affiche la surface et le périmètre du rectangle.

Énoncé 4.2 : La classe Calculatrice

Durée estimative : 25 minutes

Écrivez en Java la classe `Calculatrice` qui permet de représenter une calculatrice.

Écrivez les méthodes suivantes :

- le constructeur permettant de fixer la valeur initiale de la calculatrice ;
- la méthode `getValeur` pour obtenir la valeur de la calculatrice ;
- la méthode `ajoute` pour ajouter à la valeur de la calculatrice une valeur passée en paramètre ;
- la méthode `multiplie` pour multiplier la valeur de la calculatrice par une valeur passée en paramètre ;
- la méthode `divise` pour diviser la valeur de la calculatrice par une valeur passée en paramètre.

Écrivez ensuite un programme principal qui propose en boucle à l'utilisateur de choisir entre ajouter un nombre, multiplier ou diviser la valeur de la calculatrice par un nombre ou terminer l'exécution du programme. La valeur initiale de la calculatrice est zéro.

Énoncé 4.3 : La classe Complexe

Durée estimative : 30 minutes

Écrivez en Java la classe `Complexe` qui permet de représenter des nombres complexes en définissant leur partie réelle et leur partie imaginaire qui sont deux nombres réels indépendants.

Écrivez les méthodes suivantes :

- le constructeur permettant de donner la valeur initiale de la partie réelle et de la partie complexe ;
- la méthode `getRéel` pour obtenir la partie réelle ;
- la méthode `setRéel` pour modifier la partie réelle ;
- la méthode `getImaginaire` pour obtenir la partie imaginaire ;

- la méthode `setImaginaire` pour modifier la partie imaginaire ;
- la méthode `module` pour calculer le module du nombre complexe ;
- la méthode `ajouteRéel` pour ajouter un nombre réel au nombre complexe ;
- la méthode `multiplieRéel` pour multiplier le nombre complexe par un nombre réel.

Écrivez ensuite un programme principal qui crée un objet de la classe `Complexe` après avoir préalablement lu au clavier sa partie réelle et sa partie imaginaire. Ensuite, le programme affiche le module puis ajoute un premier nombre réel au nombre complexe et multiplie le nombre complexe par un second nombre réel. Ces deux nombres réels sont lus au clavier.

Indice pour l'énoncé 4.3

Soit a la partie réelle d'un nombre complexe et b sa partie imaginaire :

- *La valeur du module est égale à la racine carrée de a^2+b^2 .*
- *Additionner le nombre réel v à ce nombre complexe ne modifie que sa partie réelle qui devient $a+v$.*
- *Multiplier ce nombre complexe par le nombre réel u modifie les deux parties, la partie réelle devenant $a*u$ et la partie imaginaire devenant $b*u$.*

Énoncé 4.4 : La classe `NombreAléatoire`

Durée estimative : 25 minutes

Écrivez en Java la classe `NombreAléatoire` qui calcule un nombre entier aléatoire entre une borne inférieure et une borne supérieure.

Écrivez les méthodes suivantes :

- le constructeur qui prend comme paramètre la borne inférieure et la borne supérieure de l'intervalle ;
- la méthode `getBorneInf` qui renvoie la borne inférieure de l'intervalle ;
- la méthode `getBorneSup` qui renvoie la borne supérieure de l'intervalle ;
- la méthode `getValeur` qui renvoie le nombre généré ;
- la méthode `compare` qui prend une valeur comme paramètre et qui renvoie :
 - 0 si la valeur est égale au nombre choisi aléatoirement ;
 - 1 si la valeur est strictement supérieure au nombre aléatoire ;
 - 1 si la valeur est strictement inférieure au nombre aléatoire ;
- la méthode `reCalcule` qui recalcule un nouveau nombre entier entre la borne inférieure et la borne supérieure.

Écrivez ensuite un programme principal de test qui affiche 20 nombres entiers aléatoires compris entre 1 et 10.

Indice pour l'énoncé 4.4

Pour calculer le nombre aléatoire, vous pouvez utiliser la fonction de l'exercice 3.5.

Énoncé 4.5 : La classe Vecteur

Durée estimative : 20 minutes

Écrivez la classe `Vecteur`. Un vecteur est un tableau d'entiers de taille fixe. Cette taille est fixée lors de la création du vecteur. La valeur de la taille n'est pas conservée dans un attribut de la classe.

Écrivez les méthodes suivantes :

- le constructeur qui prend la taille en paramètre ;
- la méthode `getÉlément` qui prend comme paramètre un indice et qui renvoie la valeur du vecteur à cet indice ;
- la méthode `setÉlément` qui prend comme paramètres un indice du vecteur ainsi que la valeur à affecter à cet indice ;
- la méthode `lis` qui lit au clavier tous les éléments du vecteur ;
- la méthode `affiche` qui écrit tous les éléments du vecteur à l'écran ;
- la méthode `remplisAléatoirement` qui prend deux paramètres entiers `min` et `max` et remplit les valeurs du vecteur avec des nombres aléatoires entiers compris entre `min` et `max` (bornes incluses).

Écrivez un programme qui crée un vecteur dont la taille est fixée par l'utilisateur, lit ses valeurs au clavier puis les affiche à l'écran.

Ensuite le programme remplit les valeurs du vecteur aléatoirement (entre 1 et 20) puis les affiche à l'écran.

Énoncé 4.6 : La classe Complexe2

Durée estimative : 30 minutes

Écrivez la classe `Complexe2` en reprenant tous les éléments de la classe `Complexe` de l'exercice 4.3.

Ajoutez les méthodes :

- la méthode `ajouteComplexe` pour ajouter un nombre complexe au nombre complexe ;
- la méthode `multiplieComplexe` pour multiplier le nombre complexe par un autre nombre complexe.

Ces deux méthodes prennent un objet de la classe `Complexe2` en paramètre.

Écrivez ensuite une extension au programme principal pour qu'il réalise une addition puis une multiplication de nombres complexes.

Indice pour l'énoncé 4.6

Soit a la partie réelle d'un premier nombre complexe et b sa partie imaginaire, puis c la partie réelle d'un second nombre complexe et d sa partie imaginaire

- *La somme des deux nombres complexes a pour partie réelle $a+c$ et pour partie imaginaire $b+d$.*
- *La multiplication des deux nombres complexes a pour partie réelle $a*c - b*d$ et pour partie imaginaire $a*d + b*c$.*

Énoncé 4.7 : La classe Angle

Durée estimative : 20 minutes

Écrivez en Java la classe `Angle` qui contient la valeur d'un angle. Sa valeur initiale est zéro.

Écrivez les méthodes suivantes :

- la méthode `getAngleGrades` pour obtenir la valeur de l'angle en grades ;
- la méthode `setAngleGrades` pour fixer la valeur de l'angle en grades ;
- la méthode `getAngleDegrés` pour obtenir la valeur de l'angle en degrés ;
- la méthode `setAngleDegrés` pour fixer la valeur de l'angle en degrés ;
- la méthode `getAngleRadians` pour obtenir la valeur de l'angle en radians ;
- la méthode `setAngleRadians` pour fixer la valeur de l'angle en radians.

Bien entendu, la valeur de l'angle n'est stockée en interne que dans une seule unité, à savoir dans le cadre de cet exercice, en grades.