

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, ses attributs et ses 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.

Prérequis

Pour valider les prérequis nécessaires avant d'aborder le TP, répondez aux questions ci-après.

Questions générales

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. 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.

4. 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.
5. Quand le constructeur d'une classe est-il invoqué?
6. 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.

Questions Java

7. Comment les objets sont-ils créés à partir d'une classe?
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, 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.
11. Quelle est la différence entre les modes d'encapsulation `public`, `private` et `protected`?

Questions Python

12. Comment les objets sont-ils créés à partir d'une classe?
13. Quel est le nom de la méthode spéciale remplissant le rôle de constructeur?
14. Comment sont déclarés les attributs des objets?
15. Quel est le rôle du premier paramètre nommé généralement `self` dans la définition de chaque méthode?
16. Python offre-t-il plusieurs de niveaux de visibilité pour les attributs et les méthodes?

Énoncé 4.1 La classe Rectangle

Durée estimative : 20 minutes

Écrivez en Java/Python la classe `Rectangle` qui permet de représenter des rectangles en définissant leur hauteur et leur largeur (les rectangles sont horizontaux et ne vous préoccupe 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` (Java)/`set_largeur` (Python) pour modifier la largeur ;
- la méthode `setHauteur` (Java)/`set_hauteur` (Python) pour modifier la hauteur ;
- la méthode `getLargeur` (Java)/`get_largeur` (Python) pour obtenir la largeur ;
- la méthode `getHauteur` (Java)/`get_hauteur` (Python) pour obtenir la hauteur ;
- la méthode `surface` qui calcule la surface du rectangle ;
- la méthode `perimetre` 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.

Corrigé p. 206

Énoncé 4.2 La classe Calculatrice

Durée estimative : 25 minutes

Écrivez en Java/Python 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` (Java)/`get_valeur` (Python) 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.

Corrigé p. 208

Énoncé 4.3 La classe Complexe

Durée estimative : 30 minutes

Écrivez en Java/Python 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 celle de la partie complexe ;
- la méthode `getReel` (Java)/`get_reel` (Python) pour obtenir la partie réelle ;
- la méthode `setReel` (Java)/`set_reel` (Python) pour modifier la partie réelle ;
- la méthode `getImaginaire` (Java)/`get_imaginaire` (Python) pour obtenir la partie imaginaire ;
- la méthode `setImaginaire` (Java)/`set_imaginaire` (Python) pour modifier la partie imaginaire ;
- la méthode `module` pour calculer le module du nombre complexe ;
- la méthode `ajouteReel` (Java)/`ajoute_reel` (Python) pour ajouter un nombre réel au nombre complexe ;
- la méthode `multiplieReel` (Java)/`multiplie_reel` (Python) 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 $b*u$.

Corrigé p. 211

Énoncé 4.4 La classe NombreAleatoire

Durée estimative : 25 minutes

Écrivez en Java/Python la classe `NombreAleatoire` 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` (Java)/`get_borne_inf` (Python) qui renvoie la borne inférieure de l'intervalle ;
- la méthode `getBorneSup` (Java)/`get_borne_sup` (Python) qui renvoie la borne supérieure de l'intervalle ;
- la méthode `getValeur` (Java)/`get_valeur` (Python) 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 vingt 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.

Corrigé p. 213

Énoncé 4.5 La classe Vecteur

Durée estimative : 20 minutes

Écrivez la classe `Vecteur` en Java/Python. 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 `getElement` (Java)/`get_element` (Python) qui prend comme paramètre un indice et qui renvoie la valeur du vecteur à cet indice ;
- la méthode `setElement` (Java)/`set_element` (Python) qui prend comme paramètres un indice ainsi que la valeur à affecter à l'élément du vecteur se situant à cet indice ;
- la méthode `lit` qui lit au clavier tous les éléments du vecteur ;
- la méthode `affiche` qui écrit tous les éléments du vecteur à l'écran ;

- la méthode `remplitAleatoirement` (Java)/`remplit_aleatoirement` (Python) 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 principal 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.

Corrigé p. 216

Énoncé 4.6 La classe `Complexe2`

Durée estimative : 30 minutes

En reprenant tous les éléments de l'exercice 4.3, écrivez la classe `Complexe2` en Java/Python. Ajoutez les méthodes :

- la méthode `ajouteComplexe` (Java)/`ajoute_complexe` (Python) pour ajouter un autre nombre complexe au nombre complexe ;
- la méthode `multiplieComplexe` (Java)/`multiple_complexe` (Python) 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 $a*d + b*c$.*

Corrigé p. 218

Énoncé 4.7 La classe `Angle`

Durée estimative : 20 minutes

Écrivez en Java/Python la classe `Angle` qui contient la valeur d'un angle. Sa valeur initiale est 0.

Écrivez les méthodes suivantes :

- la méthode `getAngleGrades` (Java)/`get_angle_grades` (Python) pour obtenir la valeur de l'angle en grades ;
- la méthode `setAngleGrades` (Java)/`set_angle_grades` (Python) pour fixer la valeur de l'angle en grades ;

- la méthode `getAngleDegres` (Java)/`get_angle_degres` (Python) pour obtenir la valeur de l'angle en degrés ;
- la méthode `setAngleDegres` (Java)/`set_angle_degres` (Python) pour fixer la valeur de l'angle en degrés ;
- la méthode `getAngleRadians` (Java)/`get_angle_radians` (Python) pour obtenir la valeur de l'angle en radians ;
- la méthode `setAngleRadians` (Java)/`set_angle_gradians` (Python) 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.

Écrivez ensuite un programme qui permet de lire la valeur de l'angle dans l'unité choisie par l'utilisateur puis affiche sa valeur dans les trois unités.

Indice pour l'énoncé 4.7

Pour rappel, 180 degrés correspondent à π radians et à 200 grades.

Corrigé p. 222

Énoncé 4.8 La classe Fraction

Durée estimative : 35 minutes

Le but de l'exercice est d'élaborer en Java/Python la classe `Fraction` qui représente les nombres rationnels positifs, à savoir ceux qui peuvent être représentés sous la forme d'une fraction (a/b) constituée d'un numérateur (a) et d'un dénominateur (b). a est un entier positif, b est un entier positif non nul.

Écrivez les méthodes suivantes :

- Le constructeur qui prend en paramètres le numérateur et le dénominateur. Ces paramètres ne sont pris en compte que s'ils répondent aux conditions énoncées ci-dessus. Sinon, le numérateur vaut 0 et le dénominateur 1.
- La méthode `reel` qui renvoie la valeur de la fraction sous la forme d'un nombre réel.
- La méthode `reduit` qui réduit au maximum la valeur du numérateur et du dénominateur sans changer la valeur mathématique de la fraction. N'oubliez pas le cas particulier du numérateur de valeur nulle.
- La méthode `addition` qui prend une autre fraction en paramètre et renvoie comme résultat la somme des deux fractions. Le résultat doit être une fraction réduite.

Écrivez un programme principal qui crée une fraction dont le numérateur et le dénominateur sont lus au clavier. Le programme affiche alors la valeur réelle de la fraction ainsi que la fraction réduite. Ensuite le programme ajoute une seconde fraction à cette première fraction. Le numérateur et le dénominateur de la seconde fraction sont également lus au clavier. Enfin, le programme affiche le résultat de l'addition.

Indice pour l'énoncé 4.8

La somme de la fraction a/c et de la fraction b/d est la fraction $(ad+bc)/cd$ avant toute réduction.

Pour réduire une fraction, il faut diviser le numérateur et le dénominateur par leur PGCD. Référez-vous à l'exercice 1.13 pour savoir comment calculer le PGCD.

Corrigé p. 225

Énoncé 4.9 La classe Rectangle2

Durée estimative : 40 minutes

Le but de l'exercice est d'écrire en Java/Python la classe `Rectangle2` qui représente des rectangles dont les côtés sont parallèles avec l'axe des abscisses (X) et l'axe des ordonnées (Y).

Écrivez les méthodes suivantes :

- le constructeur qui prend comme paramètres quatre nombres réels : `xmin`, `xmax`, `ymin` et `ymax` ;
- les méthodes `getXmin`, `getXmax`, `getYmin` et `getYmax` (Java)/`get_xmin`, `get_xmax`, `get_ymin` et `get_ymax` (Python) qui renvoient les quatre coordonnées du rectangle ;
- la méthode `intersection` qui prend un autre rectangle en paramètre et renvoie comme résultat un nouveau rectangle qui est l'intersection des deux rectangles.

Écrivez un programme qui crée deux rectangles, calcule leur intersection et affiche les quatre coordonnées de celle-ci.

Indices pour l'énoncé 4.9

Pour écrire la méthode `intersection`, le calcul se fait indépendamment sur chaque axe. Il faut utiliser une fois le maximum et une fois le minimum pour effectuer le calcul sur chaque axe. En cas d'intersection vide, pensez surtout à retourner un rectangle vide respectant `xmin=xmax=0` et `ymin=ymax=0`.

N'hésitez pas à travailler sur un exemple pour trouver comment il faut écrire la méthode `intersection`.

Corrigé p. 228

Énoncé 4.10 La classe Ensemble

Durée estimative : 25 minutes

Dans les exercices 2.5 et 2.6, un ensemble d'entiers est représenté par deux variables :

- un tableau d'entiers dont la taille représente la taille maximale de l'ensemble ;
- une variable de type entier dont la valeur est la taille de l'ensemble.

L'utilisation d'une classe est tout à fait adaptée pour une telle représentation. Par conséquent, écrivez en Java/Python la classe `Ensemble`. L'état initial d'un ensemble est vide.

En vous basant sur les corrigés des exercices 2.5 et 2.6, écrivez les méthodes suivantes :

- Le constructeur qui prend pour paramètre la taille maximale de l'ensemble.
- La méthode `insere` qui ajoute un nouvel entier dans l'ensemble. La valeur de retour est à vrai si le nouvel entier a été inséré (même s'il était déjà présent) et à faux si l'insertion n'est pas possible (lorsqu'il n'y a plus de place dans le tableau).
- La méthode `supprime` pour supprimer un élément de l'ensemble. La valeur de retour est à vrai si l'élément était présent et à faux dans le cas contraire.
- La méthode `affiche` qui sert à écrire à l'écran les différentes valeurs de l'ensemble.

Écrivez ensuite le programme principal en Java/Python qui lit au clavier des valeurs entières pour les introduire dans l'ensemble jusqu'à ce que la taille maximale de celui-ci soit atteinte. Ensuite, le programme donne la possibilité à l'utilisateur de supprimer une valeur.

Corrigé p. 231

Énoncé 4.11 La classe `EnsembleTrie`

Durée estimative : 35 minutes

Écrivez la classe `EnsembleTrie` en Java/Python. Ses attributs sont les mêmes que ceux de la classe `Ensemble` introduite dans l'exercice précédent.

Il faut également écrire les mêmes méthodes que dans l'énoncé précédent mais en y incluant les mécanismes qui garantissent que l'ensemble reste en permanence trié et qui sont décrits dans les exercices 2.7 et 2.8. Utilisez les corrigés de ces deux exercices pour écrire les méthodes.

Modifiez le programme principal de l'exercice précédent pour qu'il utilise la classe `EnsembleTrie` à la place de la classe `Ensemble`.

Corrigé p. 234

Énoncé 4.12 La classe `Matrice`

Durée estimative : 35 minutes

Écrivez en Java/Python la classe `Matrice` qui représente des matrices rectangulaires. Les attributs définis dans la classe sont :

- le nombre de lignes (`nombreLignes`) ;
- le nombre de colonnes (`nombreColonnes`) ;
- le tableau contenant les valeurs de la matrice.