



TI215 : programmation orientée objet 1

Exercices : série introductive

Objectifs

- Prendre en main l'environnement de développement intégré (IDE)
- Manipuler les notions élémentaires : définir une classe, créer des objets à partir de cette classe et appeler des méthodes

Exercice 1 : Prise en main de l'IDE et entrées-sorties

Objectifs spécifiques

- Installer eclipse
- Créer un projet
- Afficher des informations à la console

Étape 1 : Installer Eclipse

Lancer Eclipse en choisissant le répertoire par défaut dans lequel vous voulez travailler

Étape 2 : Créer un projet

Créez un nouveau projet appelé PremierProjet.

File ⇒ New ⇒ Choisir un projet de type Java

Donnez un nom à votre projet. Le projet contient un répertoire appelé src qui va contenir le code source du programme.

Étape 3 : Créer un package

En vue de pouvoir réutiliser les classes que l'on écrit, il faut toujours créer un package pour les y placer et donc éviter de travailler dans le package par défaut (default package).

Attention, le nom d'un package commence par une minuscule.

Créez un package appelé premierPackage.

Clic droit sur src ⇒ New ⇒ Package

Étape 4 : Créer une classe avec la méthode main

Dans le package premierPackage, créez une classe appelée Principale.

Clic droit sur le package ⇒ New ⇒ Class

En tant que classe principale, elle doit contenir une méthode `public static void main(String[] args)` qui sera exécutée automatiquement au lancement du

projet (cette méthode joue un rôle identique à celui de la fonction `main` dans un programme C). Dans cette méthode, vous pouvez ajouter du code à la manière d'un langage impératif. Pour que cette méthode soit automatiquement créée, cochez l'option correspondante.

Étape 5 : Variables

Dans la méthode `main`, déclarez deux variables entières `x` et `y` ainsi qu'une variable de type chaîne de caractères (String) appelée `mot`.

Camélisation

En java, les noms de variables ou de méthodes commencent par une minuscule et respectent la notation appelée "camel case" (aussi stylisé en camelCase). Le "camel case" fait par défaut référence au "lower camel case", c'est-à-dire que la première lettre du nom de la variable ou de la méthode est en minuscule. Il convient de respecter la notation "Camel Case" en Java car c'est un des pragmas (bonnes pratiques) du langage.

Ajoutez les instructions permettant d'affecter respectivement à ces variables les valeurs : 7, 3 et "Java".

Étape 6 : Affichage à la console

Ajoutez une instruction permettant d'afficher à la console la phrase ci-dessous (pour afficher un message, utilisez `System.out.println`).

Note

En toute généralité, dans les exemples de sorties, les parties soulignées et entre parenthèses sont des parties qui doivent être remplacées par les valeurs des variables correspondantes.

Le produit vaut (produit des deux entiers) et le mot est (contenu de la variable mot).

Pour rappel, l'opérateur `+` appliqué à deux variables de type numérique exécute l'addition. L'opérateur `+` appliqué à deux objets de type String est l'opérateur de concaténation.

Exécutez ensuite le projet :

Première exécution : Clic droit sur la classe contenant la méthode `main` ⇒ Run as ⇒ `application java`

Exécutions suivantes : icône « Flèche blanche dans cercle vert ».

Exercice 2 : Classe Rectangle

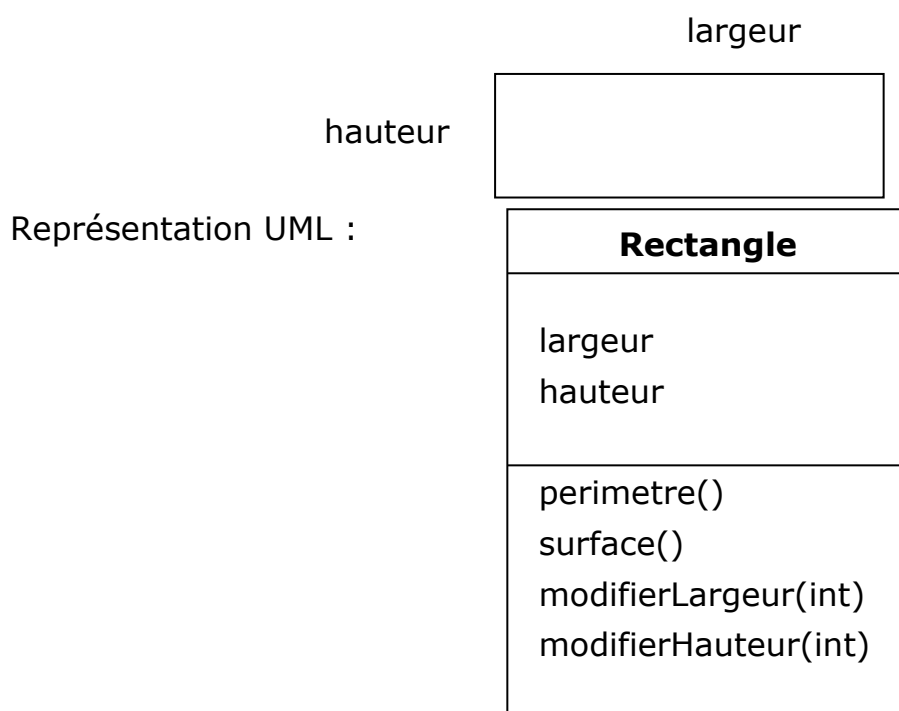
Objectifs spécifiques

- Créer des objets à partir d'une classe
- Appeler des méthodes sur des objets

Étape 1 : Créer une classe

Toujours dans `PremierProjet` et dans le même package, créez une nouvelle classe (un fichier par classe Java) appelée `Rectangle` permettant de gérer des rectangles : *Clic droit sur premierPackage* ⇒ *New* ⇒ *Class*

Tout rectangle est décrit par sa largeur et sa hauteur.



Pour le code de la classe `Rectangle`, vous pouvez vous inspirer du syllabus.

Étape 2 : Déclarer les variables d'instance (ou attributs)

La classe comportera les variables d'instance suivantes de type entier (avec la protection par défaut : visibilité package) :

- `largeur`
- `hauteur`

Étape 3 : Constructeur

Créez un constructeur pour la classe `Rectangle` ; ce constructeur reçoit en arguments la largeur et la hauteur du rectangle.

Étape 4 : Création d'un objet de la classe

Pour tester votre code créé jusqu'ici, dans la classe `Principale`, remplacez le contenu de la méthode `main` par :

- La déclaration d'un objet appelé `premierRectangle` de type `Rectangle` ;

- L'instanciation de `premierRectangle` en utilisant le constructeur. Les valeurs à transmettre lors de cette instanciation sont largeur = 10, hauteur = 4.

Étape 5 : Accès aux variables d'instance d'un objet

Dans la méthode `main` :

- Affichez (via `System.out.println`) la largeur et la hauteur de l'objet `premierRectangle` ;
- Modifiez les variables d'instance de l'objet `premierRectangle` : largeur = 5, hauteur = 6 ;
- Affichez de nouveau la largeur et la hauteur de l'objet `premierRectangle`.

Étape 6 : Définition de méthodes

Ajoutez dans la classe `Rectangle` :

- Une méthode appelée `modifierLargeur` qui reçoit en argument un entier (positif ou négatif) et qui ajoute cette valeur à la largeur du rectangle (*méthode sans type de retour*)
- Une méthode appelée `modifierHauteur` qui reçoit en argument un entier (positif ou négatif) et qui ajoute cette valeur à la hauteur du rectangle (*méthode sans type de retour*)
- Une méthode appelée `perimetre` (*sans arguments*) qui retourne le périmètre calculé du rectangle
- Une méthode appelée `surface` (*sans arguments*) qui retourne la surface calculée du rectangle

Étape 7 : Appel de méthode sur des objets

Dans la classe `Principale`, appelez ces différentes méthodes sur l'objet `premierRectangle` :

- Affichez à la console le périmètre et la surface de l'objet `premierRectangle` ;
- Elargissez et agrandissez `premierRectangle` en appelant les méthodes adéquates ;
- Affichez ensuite de nouveau la largeur et la hauteur de `premierRectangle`.