

Glossaire :

I. Orienté objet :

Objet : Instance de classe.

Classe : Ce sont des prototypes (« des plans ») caractérisés par des méthodes et des attributs et à partir desquels on peut instancier des objets.

Méthode : C'est une fonction de classe.

Fonction : C'est un conteneur d'instructions.

Constructeur : C'est la méthode à partir de laquelle un objet est instancié.

Attribut : C'est une variable de classe.

Variable : C'est un conteneur de valeur.

Champ : Synonyme de variable.

Variable d'instance : Variable unique à chaque instance de classe.

Variable de classe : Idem sauf qu'elles sont uniques à la classe plutôt qu'à l'instance et définie avec static et optionnellement final.

Variable locale : Définie à l'intérieur de méthode et visible que là.

Modificateur d'accès : C'est une propriété qui permet de délimiter la portée d'un conteneur (classe, variable, fonction...).

Paramètre : Variables définies dans une déclaration de méthode.

Argument : C'est la valeur réelle insérée dans la méthode ou la fonction.

Surcharge (overload) : C'est la copie exacte d'une méthode (ou fonction) existante avec un ou plusieurs paramètres en plus.

Bloc d'initialisation : C'est un bloc d'instruction délimité par des accolades à l'intérieur d'une classe.

Ombrage de champ (field shadowing) : C'est lorsque le nom d'un argument de méthode est le même que celui d'une variable locale (this permet d'éviter ce problème).

Static : Mot clef qui permet de signifier que quelque chose est directement lié à une classe et non une instance de celle-ci.

Final : Mot clef qui va rendre constant ce qui y est associé.

Nested Class : Classe déclarée à l'intérieur d'une autre.

Local Class : Classe définie dans un bloc de code comme une méthode, un if, ou encore, une boucle.

Anonymous classe : Interface qui permet de déclarer et d'instancier une classe en même temps.

Héritage : Technique qui permet de définir une nouvelle classe qui va adopter les caractéristiques et les comportements d'une classe existante

Sous-classe (Subclasse) : La classe qui hérite d'une autre.

Extends : Mot clef utilisé pour l'héritage.

Overrinding : Fait pour une classe héritant de modifier des méthodes existantes de sa classe supérieur et sous certaine condition.

Super : Mot clef qui permet d'accéder aux méthodes, aux variables d'instance ou au constructeur de la classe supérieur et d'en récupérer le résultat.

Classe abstraite : Classe qui ne peut pas être instanciée et conçue principalement pour être héritée.

Abstract : Mot clef qui permet de définir une classe comme étant abstraite.

Interface : C'est un « type » de référence. Ce n'est pas une classe.

Default : Mot clef qui permet de définir une méthode dans une interface et qui a une implémentation réelle. Utile si on veut modifier une interface sans forcer des changements en cascade.

Interface fonctionnelle (functional interface) : Interface qui contient une seule méthode abstraite (mais peut avoir de multiples default ou static).

Polymorphisme : Capacité d'une abstraction (interface ou classe de base) à prendre de nombreuses formes au moment de l'exécution.

Lambda : Moyen de représenter une interface fonctionnelle en utilisant une expression.