## Attribut (*attribute*)

Un attribut d'un objet est une information propre à cet objet. Un objet peut avoir un nombre arbitraire d'attributs. Les valeurs de ses attributs constituent son *état*.

**Synonymes :**

- champs (*field*)
- donnée membre (*data member*)
- propriété (*property*) : attribut auquel est associé un mutateur (*setter*) et un accesseur (*getter*)
- variable d'instance (*instance variable*)

## Bloc (*block*)

Un bloc est une portion de programme délimité par une paire d'accolades (`{` et `}`), permettant de regrouper soit des déclarations, soit des instructions ensemble. On trouve des blocs :

- lors de la déclaration d'une classe, le bloc contient alors les déclarations de méthodes, d'attributs et de constructeurs.
- lors de la déclaration d'une méthode et d'un constructeur, le bloc contient les instructions de la méthode ou du constructeur.
- lors de l'utilisation d'une structure de contrôle (`for`, `if`, `while`,...), le bloc contient les instructions associées.

## Classe (*class*)

Une classe est un plan permettant de créer des objets. La classe définit quelles seront les attributs et les méthodes des objets créés à partir d'elle.

## Compiler (*compile*)

Le programme écrit en Java n'est pas directement exécutable par le processeur de l'ordinateur. Il doit être transformé dans un langage adapté à l'ordinateur. Ce processus est appelé *compilation* et est effectué par un programme appelé `compilateur`, pour Java, c'est le programme `javac`.

## Déclaration (*declaration*)

Une déclaration permet de définir une entité dans le programme. En Java, toute variable que l'on souhaite utiliser doit d'abord être déclarée. La déclaration d'une variable est toujours de la forme *type de la variable* suivi de l'*identifiant de la variable*. L'avantage (par rapport à un langage comme Python qui ne demande pas de déclaration) est que si on fait une erreur typographique en écrivant le nom d'une variable, le compilateur détecte immédiatement l'erreur puisque l'identifiant faux ne correspond probablement pas à une variable déclarée.

```
1 int toto = 42;
2 System.out.println("toto = " + tito); // erreur en Java, tito non déclaré.
```

En Java, on doit aussi déclarer les classes, et dans les classes on doit déclarer les méthodes, les attributs et les constructeurs.

**Synonymes :**

— définition (*definition*)

## Instancier (*instantiate*)

Instancier est l'action de créer un objet à partir d'une classe, en utilisant l'instruction `new`.

**Synonymes :**

— construire (*construct*)
— créer (*create*)

## Méthode (*method*)

Une méthode est une fonction définie dans une classe. Il existe deux variantes de méthodes :

— méthode d'instance (*instance method*) : méthode appelée à partir d'un objet (et donc d'une instance d'une classe) qui permet d'adresser des requêtes à l'objet.
— méthodes de classe ou méthode statique (*class method* ou *static method*) : méthode appelée à partir d'une classe qui peut affecter la classe en entier.

## Objet (*object*)

Un objet est une entité, constitué d'un *état* (les valeurs des *attributs*), et de divers point d'accès (les *méthodes*) permettant de lui adresser des requêtes. Son état peut évoluer au cours de l'exécution du programme. On peut le voir comme un petit composant indépendant du programme.

**Synonymes :**

— instance (*instance*) : une instance est un objet d'une certaine *classe*.

## Point d'entrée (*entry point*)

Le point d'entrée du programme correspond aux instructions que le programme doit exécuter, et est formée d'une *méthode de classe* dont le format est : *

```java
public static void main(String[] args) {
// instructions du programme
}
```

En général, cette méthode crée un ou plusieurs objets, puis appelle des méthodes de ces objets. Elle ne contient donc pas tout le programme, mais juste le nécessaire pour l'initier.

## Retourner (*return*)

Une méthode est un point d'accès permettant de faire une requête à un objet. La requête peut nécessiter une réponse. Les instructions de la méthode doivent donc fournir une réponse, on dit : retourner une réponse. L'instruction pour retourner une réponse est l'instruction `return`.

**Synonymes :**

— renvoyer

## Source (*source*)

Les fichiers d'extensions `.java` constituant le programme sont appelés les sources du programme. Ces sources peuvent être *compilés* en un *programme exécutable*, compréhensibles par le processeur. De manière générale, les sources sont les fichiers de programmes tels qu'écrits par les développeurs, alors que la plupart des programmes sont distribués sous leur forme compilée, et donc sans les sources (ce qui empêche leur modification ou l'analyse de ce qu'ils font). Les logiciels `open source` sont des logiciels dont les sources sont rendus disponibles au public, dans un souci de transparence et/ou de travail collaboratif (chacun pouvant alors proposer des modifications au logiciel).

## Type (*type*)

Les données manipulées par le programme ont une nature un nombre (entier ou à décimale), un texte, une fiche d'identité, un relevé bancaire, ... Dans chaque cas, les opérations pouvant s'appliquer sur ces données ne sont pas les mêmes : on peut additionner des entiers, mais pas des fiches d'identité. En programmation on parle de *type* : des données de même type supportent les mêmes opérations. En Java, chaque variable, chaque attribut possède un type, les données stockées dans cette variable ou cette attribut doivent respecter ce type. Ainsi une variable de type `int` ne peut pas contenir $\pi$, ni un relevé bancaire. Voici quelques types de Java : `int` (nombre entier), `double` (nombre décimal), `String` (texte), `boolean` (valeur de vérité). Toute classe est aussi un type, par exemple `Book`.