Déclaration d'une variable

```
type nom = valeur ;
```

Exemples:

```
String message = "Hello!";
int compteur = 0;
String réponse = System.console().readLine();
boolean estPositif = compteur > 0;
double moitié = x / 2;
```

Types importants

`String`	Texte. Ex: `"abc"`
`int`	Nombre entier. Ex: `10`
`double`	Nombre décimal. Ex: `3.14`
`boolean`	`true` ou `false`

Signature d'une méthode

```
static typeDeRetour nom ( typeParamètre1 nomParamètre1 , ...) { ... }
```

Exemples:

```
static void main(String[] args) { ... }
static void ditHello() { ... }
static boolean demandeOuiOuNon(String question) { ... }
static double puissance(double nombre, double exposant) { ... }
```

Méthodes Java utiles

```
void System.out.println(String message)
String System.console().readLine()
int Integer.parseInt(String nombreEnTexte)
double Math.random()
double Math.round(double nombre)
double Math.ceil(double nombre)
double Math.cos(double angleEnRadian)
double Math.sin(double angleEnRadian)
```

Opérateurs importants

Avec nombres, retourne un nombre:

```
3 + 4
3 - 4
3 * 4
3 / 4
```

Avec textes, retourne un booléen:

```
"abc".equals("abc")
```

Avec textes, retourne un texte:

```
"abc" + "def"
```

Avec nombres, retourne un booléen:

```
3 == 4
3 != 4
3 > 4
3 >= 4
3 < 4
3 <= 4
```

Avec booléens, retourne un booléen:

```
true && false
true || false
```

Conditions

```
if ( valeurBooléenne ) {
  …
} else if ( valeurBooléenne ) {
  …
} else {
  …
}
```

Note: on est pas obligé d'avoir un `else if` ou un `else`

Exemple:

```
if (nombre == 0) {
  return "zero";
} else if (nombre > 0) {
  return "positif";
} else {
  return "négatif";
}
```

Boucles

```
while ( valeurBooléenne ) {
  …
}
```

Exemple:

```
int compteur = 1;
while (compteur <= 10) {
  System.out.println(compteur);
  compteur = compteur + 1;
}
```

Travailler avec les tableaux

Créer un tableau, avec valeurs:
```
String[] fruits = {"Pomme", "Orange", "Banane"};
```
Créer un tableau vide:
```
int[] nombres = new int[10]; // Taille de 10 éléments
```
Récupérer un élément:
```
String pomme = fruits[0];    // indice 0 = 1er élément
String banane = fruits[2];   // indice 2 = 3ème élément
```
Remplacer un élément:
```
fruits[1] = "Citron";
```
Récupérer la taille du tableau:
```
int taille = fruits.length;  // taille = 3
```
Boucle à travers les éléments:
```
for(String fruit : fruits) {
  System.out.println(fruit);
}
```

Classes

```
public class NomClasse {
  private typeChamp nomChamp = valeur;
  // Autres champs

  public NomClasse(typeParam nomParam, …) {
    // Code du constructeur
  }
}
```

Instantiation (appel du constructeur):

```
NomClasse nomVariable =
  new NomClasse(param1, …);
```

Référence à un champ dans la classe:

```
this.nomMembre
```

Une classe bien construite

- Tous les champs ont une valeur après l'exécution du constructeur
- Tous les champs sont **`private`**
- Tous les champs qui restent constants sont **`final`**
- Toutes les méthodes non utiles à l'utilisateur de la classe sont **`private`**

Packages

```
package nom1.nom2;

public class NomClasse { … }

NomClasse.java va dans le dossier
src/main/java/nom1/nom2
```

Utilisation (en haut du fichier):

```
import nom1.nom2.NomClasse;
```

Conventions de code en Java

Nom des classes: `MajusculeMajusculeMajuscule`. Pas de - ou _.

Nom des variables, paramètres, méthodes, packages: `minusculeMajusculeMajuscule`. Pas de - ou _.

Nom des constantes: `MAJUSCULES_MAJUSCULES_MAJUSCULES`. Pas de -.

`if` , **`else`**, **`else if`**, **`while`**, **`for`:** toujours utiliser les accolades { et }.

Accolades: { à la fin de la ligne, et } au même niveau que la ligne qui ouvre l'accolade. Exemple:

```
if (x > 0) {
  if (y > 0) {
    // Code
  }
}
```

Indentation: 4 espaces de plus à chaque niveau. Exemple:

```
public class Jaseur {
□□□□public void ditHello() {
□□□□□□□□System.out.println("Hello");
□□□□}

□□□□public void ditBye() {
□□□□□□□□System.out.println("Bye");
□□□□}
}
```