



- 1 Définissez par écrit ce que vous voulez faire : ce qui se conçoit bien s'énonce clairement !
- 2 Découpez le plus possible la logique de votre code en sous-logiques.
- 3 Écrivez chaque bout de code l'un après l'autre en prenant soin de bien les tester à chaque fois.
- 4 Commentez votre code au fur et à mesure pour qu'il reste facilement maintenable et évolutif.

### Opérateurs logiques

- &&**  
"ET" : le résultat n'est vrai que si toutes les parties participantes sont vraies.
- ||**  
"OU" : le résultat est vrai si au moins une des parties participantes est vraie.
- !**  
"NON" : il inverse l'expression donnée.

### Opérateurs de comparaison

- |  |  |
|--|--|
| <b>==</b><br>"Égal à ..." (exactement le même).                          | <b>&lt;=</b><br>"Inférieur ou égal à ...". |
| <b>!=</b><br>"Non égal à ..." (différent, de quelque façon que ce soit). | <b>&gt;</b><br>"Supérieur à ...".          |
| <b>&lt;</b><br>"Inférieur à ...".  | <b>&gt;=</b><br>"Supérieur ou égal à ...". |

### Définitions

#### Variable

Association d'un nom à une valeur. Elle est déclarée via un mot-clé et peut être de type texte, nombre, booléen, etc.

#### Boucle

Répétition d'un bloc de code tant que la condition spécifiée est valide.

#### Fonction

Ensemble d'instructions qui effectuent une tâche. Une fonction peut être appelée plusieurs fois dans un programme.

#### Tableau (array)

Bloc de mémoire où des données de même type sont rangées côte à côte.

#### Programmation Orientée Objet

Modèle qui consiste à définir et faire interagir des éléments appelés "objets".

#### Objet

Représentation qui se rattache au monde physique : un livre, une page de livre, une lettre.

```
1 public class OC {
2
3     public static void main(String[] args) {
4         Personne personne = new Personne();
5
6         int nombreDeMarchesAMonter = 2348;
7
8         while(nombreDeMarchesAMonter > 0){
9             nombreDeMarchesAMonter--;
10            personne.monteLaMarche(nombreDeMarchesAMonter);
11        }
12    }
13 }
14
15 class Personne {
16     public void monteLaMarche(int nombre){
17         System.out.println("Je monte la marche " + nombre +
18             " !");
19     }
16 }
```

### Bonnes pratiques

- ✓ Nommer les variables, les classes et les fonctions de manière explicite.
- ✓ Indenter son code pour ne pas se perdre.
- ✓ Contrôler l'accès aux variables et aux fonctions.
- ✓ Initialiser une variable au moment de la déclaration, si sa valeur est connue.

### Erreurs classiques

- ✗ Faire des fonctions trop longues qui font trop de choses.
- ✗ Utiliser le mauvais type de variable.
- ✗ Créer une boucle infinie : si la condition est toujours vraie, le programme ne s'arrêtera jamais !
- ✗ Ne pas gérer les exceptions qui peuvent survenir lors de l'exécution du programme.