Structures conditionnelles

Principes

Les structures conditionnelles sont des éléments fondamentaux de la programmation qui permettent de contrôler le flux d'exécution d'un programme en fonction de certaines conditions. Elles permettent de prendre des décisions et d'exécuter différentes actions selon que ces conditions sont vraies ou fausses.

Condition

La condition est l'élément central d'une structure conditionnelle. Elle est évaluée et doit retourner une valeur booléenne (vrai ou faux) pour déterminer le chemin d'exécution à suivre.

Exemple : x > 5, le résultat est vrai si la valeur de x est supérieure à 5, sinon elle est fausse

Type booléen

Le type booléen est un type de donnée qui ne peut prendre que deux valeurs possibles : true (vrai) ou false (faux). Il est utilisé pour représenter les résultats de conditions.

```
if (estConnecte == True) {
   System.out.println("Vous êtes connecté.");
} else {
   System.out.println("Vous n'êtes pas connecté.");
}
```

Opérateurs logiques

Les opérateurs logiques permettent de combiner plusieurs conditions pour construire des expressions conditionnelles plus complexes. Les principaux opérateurs logiques incluent :

- ET (&& ou and) : renvoie true si toutes les conditions sont vraies.
- OU (|| ou or) : renvoie true si au moins une des conditions est vraie.
- NON (! ou not) : inverse la valeur d'une condition (si la condition est true, le résultat sera false).

Forme simple (Si-Alors)

La forme la plus élémentaire des structures conditionnelles est la structure “Si-Alors”, qui exécute un bloc de code seulement si la condition est vraie.

```
habitants = 150;
if (habitants >= 100) {
    System.out.println("Réussi");
}
```

Forme complète (Si-Alors-Sinon)

La forme complète, permet de spécifier une action alternative à exécuter lorsque la condition initiale est fausse.

```
habitants = 150;
if (habitants >= 100) {
    System.out.println("Réussi");
} else {
    System.out.println("Echec");
}
```

Forme riche (Selon-cas)

La forme riche, permet de tester plusieurs conditions successives et d'exécuter un bloc de code spécifique en fonction de la première condition vraie rencontrée. C'est une alternative à l'emboîtement de structures conditionnelles simples.

switch(choix){

```
  case 1:
     System.out.println("Bonjour");
     break;
  case 2:
     System.out.println("Hello");
     break;
  case 3:
     System.out.println("Buenos dias");
     break;
  default:
     System.out.println("Choix incorrect");
     break;
```

}

String choix; if(choix == 1){

```
  System.out.println("Bonjour");
```

} else if(choix == 2){

```
  System.out.println("Hello");
```

} else if(choix == 3){

```
  System.out.println("Buenos dias");
```

} else {

```
  System.out.println("Choix incorrect");
```

}

Imbrication

L'imbrication consiste à placer une structure conditionnelle à l'intérieur d'une autre. Cela permet de créer des conditions complexes nécessitant plusieurs niveaux de décision.

```
if (score > 50) {
   System.out.println("Vous avez réussi.");

    if (score >= 80) {
        System.out.println("Excellent travail !");
    } else {
        System.out.println("Bon travail.");
} else {
   System.out.println("Échec, vous devrez réessayer.");
}
```

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc1:prog:conditionnelles&rev=1732003771

Last update: **2024/11/19 09:09**