Classes abtraites

Concept

Une **classe abstraite** est une classe qui ne peut pas être instanciée directement, mais qui sert plutôt de **modèle** ou de **base** pour d'autres classes. Elle est conçue pour **être héritée** et **étendue** par d'autres classes qui peuvent en revanche être instanciées.

Dans une classe abstraite, on pourra trouver :

- Des **méthodes et attributs classiques**, comme on en a déjà l'habitude en POO ;
- Des **méthodes dites « abstraites »** qui fournissent uniquement leur **signature**. Charge aux classes qui en héritent de définir le code de ces méthodes.

Exemple :

A.java

```
// une classe abstraite A
public abstract class A
{
    // une méthode non abstraite (=implémentée) dans A
    public void maMethode() {
        ...
    }

    // une méthode abstraite de A
    public abstract void autreMethode (int nombre) ;
}
```

B.java

```
// une classe qui dérive la classe abstraite A
public class B extends A
{

    // implémentation de la méthode abstraite de A : obligatoire,
    // sinon B reste abstraite
    public void autreMethode (int param) {
        ...
    }
}

// instanciation d'une classe qui dérive une classe abstraite : OK
B b = new B (...) ;
```

Bénéfices

- On peut placer dans une **classe abstraite** toutes les **fonctionnalités** dont on souhaite disposer dans ses **classes dérivées**, sans pour autant définir leur **fonctionnement interne**;
- On peut utiliser des **classes abstraites** pour simplement empêcher qu'une classe ne soit **instanciée** parce que son existence en tant que classe est **uniquement technique** et ne repose pas sur le besoin de manipuler l'objet en tant que tel.

Règles de construction

- Dès qu'une classe comporte une ou plusieurs méthodes abstraites, elle est abstraite, même si on ne l'indique pas avec le mot-clé « abstract » dans sa définition.
- Une classe qui hérite d'une classe abstraite n'est pas tenue de redéfinir toutes les méthodes abstraites (elle peut n'en définir que certaines). Dans ce cas elle demeure elle-même abstraite.

Exemples d'utilisation

Définir une classe abstraite

```
public abstract class Classe_A {
    ...
}
```

Définir des méthodes abstraites

```
public  class Classe_A {
    // Bien observer ici qu'il s'agit d'une signature de méthode :
    //    + Pas de corps défini => pas de {}
    //    + Un point-virgule en fin de ligne
    public abstract void uneMethode (liste_de_paramètres);
    ...
}
```

Hériter de la classe abstraite (rien de nouveau ...)

```
public class Classe_B extends Classe_A {
    ...
}
```

Implémenter les méthodes abstraites

```
public class UneClasse extends Classe_A {

    @Override
    public void uneMethode (liste_de_paramètres) {
        // Implémentation de la méthode ... qui n'est plus abstraite
    }
}
```

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

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc2:prog:poo:classesabstraites&rev=1680873979

Last update: **2023/04/07 15:26**