CHAPITRE 3

Caractériser l’abstraction

1. **Principes**
2. Classes abstraites
3. Méthodes abstraites

Principe de l'abstraction

- L'abstraction est un principe qui consiste à **ignorer certains aspects** qui ne sont pas importants pour le problème dans le but de **se concentrer sur ceux qui le sont**
- La POO permet, ainsi, de se focaliser **sur les caractéristiques importantes** d'un objet.
- Son objectif principal est de **gérer la complexité** en masquant les détails inutiles à l'utilisateur
    - Cela permet à l'utilisateur d'implémenter une logique plus complexe sans comprendre ni même penser à toute la complexité cachée

CHAPITRE 3

Caractériser l’abstraction

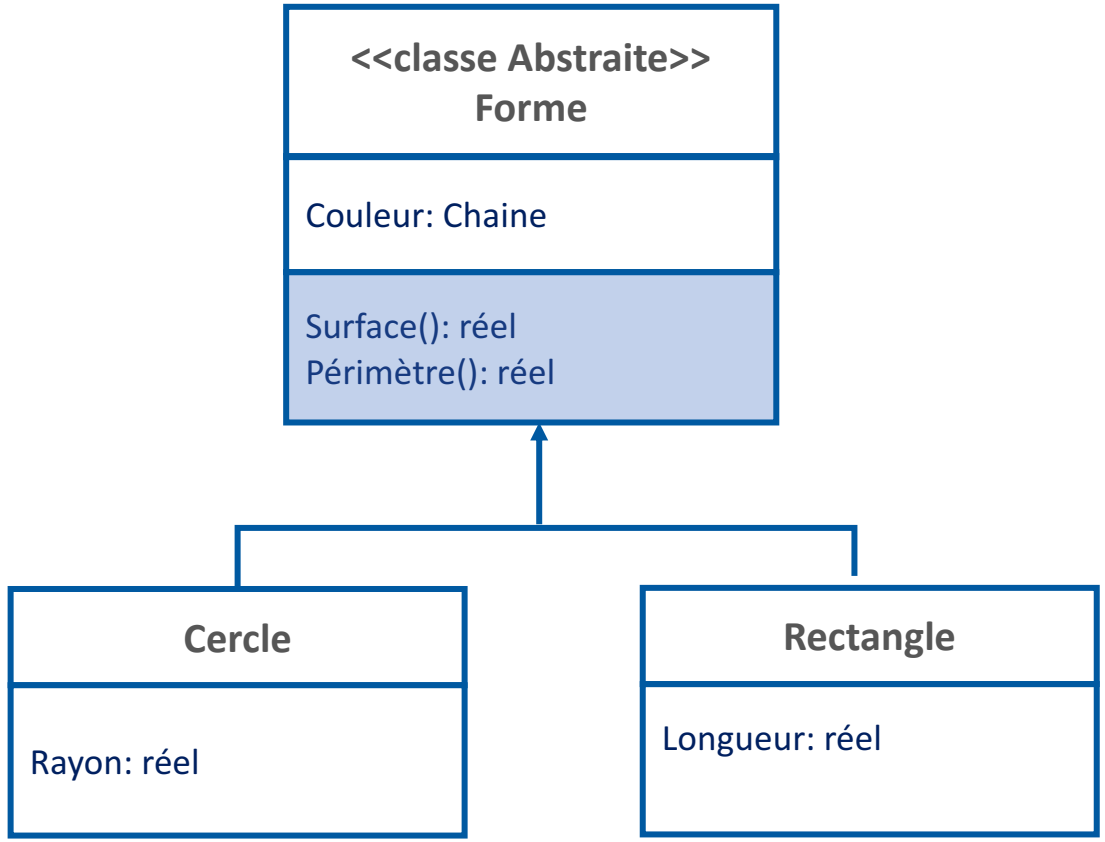
1. Principes
2. **Classes abstraites**
3. Méthodes abstraites

- Contrairement à une classe "normale", **on ne peut créer des objets à partir d'une classe abstraite.**
- L'utilité d'une classe abstraite est essentiellement en tant que classe mère.
- En ce sens, une classe abstraite peut servir :
  - **Comme racine d'un héritage en cascade :**
    - Dans ce cas, chaque sous-classe spécifiera ses propres propriétés et méthodes.
    - Favorise le polymorphisme

- Les classes abstraites permettent de définir des fonctionnalités (des méthodes) que **les sous-classes** devront **impérativement implémenter**
- Les utilisateurs des sous-classes d'une classe abstraite sont donc assurés de trouver toutes les méthodes définies dans la classe abstraite dans chacune des sous-classes concrètes
- Les classes abstraites constituent donc **une sorte de contrat** (spécification contraignante) qui garantit que certaines méthodes seront disponibles dans les sous-classes et qui oblige les programmeurs à les implémenter dans toutes les sous-classes concrètes

Exemple :

- Sachant que toutes les formes possèdent les propriétés périmètre et surface, il est judicieux de placer les méthodes périmètre()et surface() dans la classe qui est à la racine de l'arborescence (Forme)
- Les méthodes surface() et périmètre() définies dans la classe Forme ne présentent pas de code et doivent impérativement être implémentées dans les classes filles
- Les méthodes surface() et périmètre() sont des méthodes abstraites



CHAPITRE 3

Caractériser l’abstraction

1. Principes
2. Classes abstraites
3. **Méthodes abstraites**

- **Une méthode abstraite** est une méthode qui **ne contient pas de corps**. Elle possède simplement une signature de définition (pas de bloc d'instructions)
- Une méthode abstraite est déclarée dans **une classe abstraite**
- Une classe possédant une ou plusieurs méthodes abstraites devient obligatoirement une classe abstraite
- Il n'est pas indispensable d'avoir des méthodes abstraites dans une classe abstraite.

Les règles suivantes s'appliquent aux classes abstraites :

- Une sous-classe d'une classe abstraite ne peut être instanciée que si elle redéfinit chaque méthode abstraite de sa classe parente et qu'elle fournit une implémentation (un corps) pour chacune des méthodes abstraites
- Si une sous-classe d'une classe abstraite n'implémente pas toutes les méthodes abstraites dont elle hérite, cette sous-classe est elle-même abstraite (et ne peut donc pas être instanciée)