IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CLASSES ABSTRAITES EN PROGRAMMATION OBJET

Thomas Ledoux

QU'EST-CE QU'UNE CLASSE ABSTRAITE ?

Définition

En programmation objet, une classe abstraite est une classe qui ne peut pas être instanciée

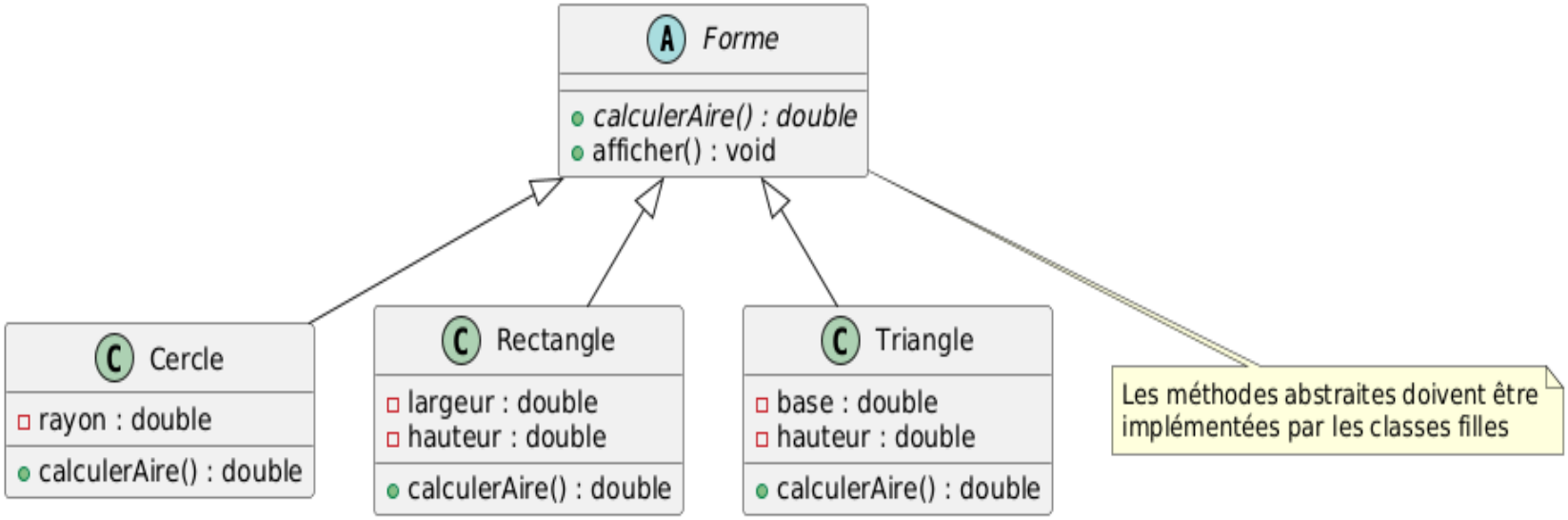
Caractéristiques

Elle sert de modèle de base pour d'autres classes dérivées dans le graphe d'héritage

Elle peut contenir des

- méthodes abstraites (sans implémentation, à redéfinir par les classes filles)
- méthodes concrètes (avec implémentation)
- attributs

MÉTHODES ABSTRAITES ET CLASSES CONCRÈTES



Mot-clef en Java

```java
abstract class Forme {
    public abstract double calculerAire(); // Abstraite
    public void afficher() { // Concrète
        System.out.println("Je suis une forme.");
    }
}

class Cercle extends Forme {
    private double rayon;
    public double calculerAire() {
        return Math.PI * rayon * rayon;
    }
}
```

POURQUOI UTILISER UNE CLASSE ABSTRAITE ?

Modélisation de concepts génériques

Représenter des idées abstraites (ex. : Forme, Véhicule, Animal)

Réutilisabilité

Factoriser du code commun à plusieurs classes

Eviter la duplication de code

Organisation du code

Imposer une structure aux classes dérivées (via les méthodes abstraites)

Polymorphisme

Permet de traiter différents types d'objets de manière uniforme

EXEMPLE DE POLYMORPHISME


```java
// Création d'un tableau de Forme (type abstrait)
// Chaque élément peut être une instance de n'importe quelle classe fille
Forme[] formes = new Forme[3];

formes[0] = new Cercle(rayon: 5.0);
formes[1] = new Rectangle(largeur: 4.0, hauteur: 6.0);
formes[2] = new Triangle(base: 3.0, hauteur: 8.0);

// CLEF DU POLYMORPHISME :
// On traite tous les objets de manière uniforme via la référence Forme
// La bonne méthode calculerAire() est appelée automatiquement selon le type dynamique
double aireTotale = 0;
for (Forme forme : formes) {
    forme.afficher();  // Appelle la méthode commune
    aireTotale += forme.calculerAire();  // Liaison dynamique !
}

System.out.printf(format: "Aire totale de toutes les formes : %.2f unités²\n", aireTotale);
```

DIFFÉRENCES ENTRE INTERFACES ET CLASSES ABSTRAITES EN JAVA

	Interface	Classe Abstraite
But principal	Contrat	Base de code commune
Héritage multiple	☑ Oui	✗ Non
Méthodes abstraites	☑ Oui	☑ Oui
Méthodes concrètes	☑ Oui (depuis Java 8)	☑ Oui
Attributs	✗ Non	☑ Oui
Constructeur	✗ Non	☑ Oui