

Programmation orienté objets en C#

Maher REBAI: Enseignant en informatique à
Ecole supérieure d'ingénieurs Léonard de Vinci
Paris la défense(ESILV)

ECE de Paris

Année Universitaire : 2018-2019

Chapitres

- Révision
- Concept de classe
- Collections
- Héritage/polymorphisme
- Classe abstraite et interface
- La délégation
- Suivi de projet

Classe abstraite et interface

Exercice

- On souhaite réaliser un programme qui gère différentes Formes géométriques. Chaque objet Forme est caractérisé par une couleur. Il peut également se décrire à travers des propriétés (tel que la spécification du rayon pour un cercle ou la spécification du côté pour un carré etc.). En particulier, on peut enregistrer dans ce programme des cercles, des carrés, des rectangles et des losanges.

Tracer le diagramme de classe pour ce programme

Implémenter le en C#.

Comment faire ?

- Déclarer une classe Forme qui contient l'attribut couleur.
- Créer la méthode SeDécrire.
- Créer les classes filles Carre, Cercle, triangle, losange qui hérite de Forme.

Remarques:

- Dans le programme, on n'a pas besoin d'objet Forme. Donc inutile de créer une classe Forme instanciable.
- Les corps de la méthode SeDécrire est à définir dans les classes filles et non dans la classe mère.
- Solution: la classe Animal devrait être définie **abstraite**.

Les classes abstraites

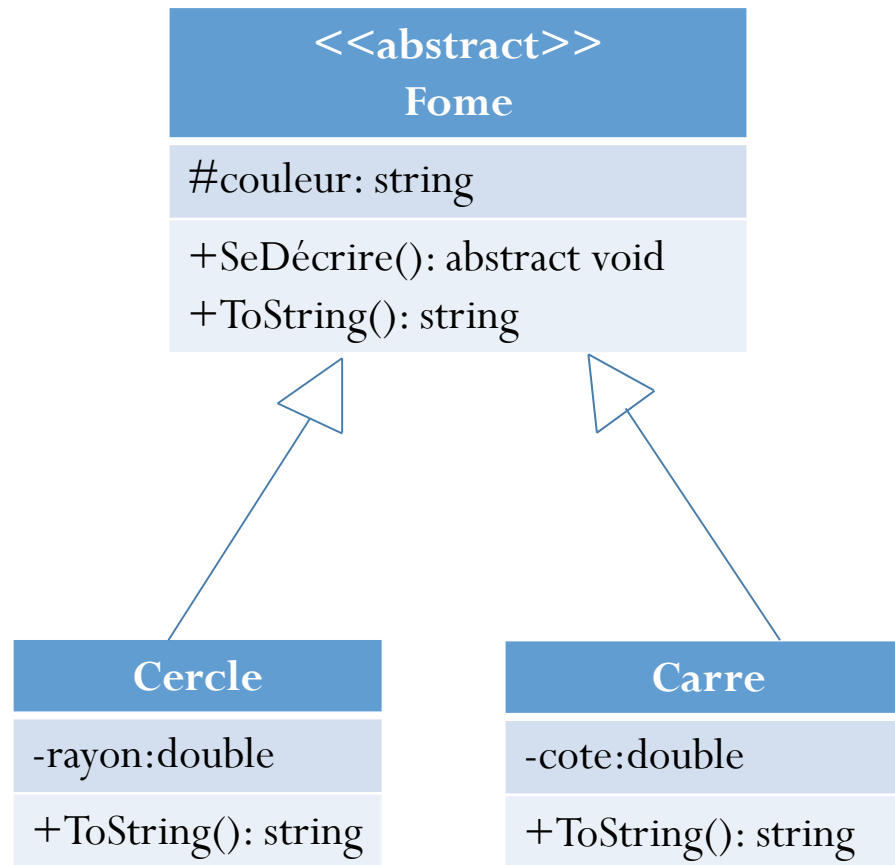
- Une classe abstraite est une classe qui ne permet pas d'instancier des objets.
- Elle ne peut servir que de classe de base pour une dérivation.
- Elle peut contenir des méthodes et des attributs dont hériteront les classes filles.
- Elle peut contenir des méthodes abstraites.

Les classes abstraites

- Une méthode abstraite est définie dans la classe mère uniquement par sa signature.
- Le corps de la méthode abstraite s'écrit différemment d'une classe fille à une autre.
- Les méthodes abstraites sont toujours publiques
- L'implémentation d'une méthode abstraite dans chaque classe fille est obligatoire.

Les interfaces

❑ Diagramme des classes



Les classes abstraites

❑ Implémentation en C#

```
abstract class Forme
{
    //Attributs
    protected string couleur;
    //Constructeur
    public Forme(string couleur)
    { this.couleur = couleur;}
    //Méthodes
    abstract public void SeDecrire();
    public override string ToString()
    {
        return "La Forme est de couleur :"+couleur;
    }
}
```

Les classes abstraites

❑ Implémentation en C#

```
class Cercle:Forme
{
    //Attributs
    private double rayon;
    //Constructeur
    public Cercle(string couleur, double rayon):base(couleur)
    { this.rayon = rayon; }
    //Méthodes
    public override void SeDecrire()
    {
        Console.WriteLine("Je suis un cercle: \nPérimètre = " +
            "2*PI*rayon\nAire = PI *rayon^2\n");
    }
    public override string ToString()
    {
        return base.ToString()+". Il s'agit d'un cercle de " +
            "rayon :" + rayon;
    }
}
```

Les classes abstraites

❑ Implémentation en C#

```
class Carre:Forme
{ //Attributs
    private double cote;
    //Constructeur
    public Carre(string couleur, double cote) : base(couleur)
    { this.cote = cote; }
    //Méthodes
    public override void SeDecrire()
    {
        Console.WriteLine("Je suis un carre: \nPérimètre = " +
            "4*cote\nAire = cote*cote\n");
    }
    public override string ToString()
    {
        return base.ToString() + ". Il s'agit d'un carré de " +
            "coté :" + cote;
    }
}
```

Les classes abstraites

❑ Implémentation en C#

```
static void Main(string[] args)
{
    Forme f1 = new Forme("Bleu");
    Console.ReadKey();
}
```

⚠ `Forme.Forme(string couleur)`

Impossible de créer une instance de la classe abstraite ou de l'interface 'Forme'

```
static void Main(string[] args)
{
    Forme f1 = new Cercle("Bleu", 15);
    Carre f2 = new Carre("Rouge", 12);
    f1.SeDecrire();
    f2.SeDecrire();
    Console.ReadKey();
}
```

Sélection E:\Enseignement_20

```
Je suis un cercle:
Périmètre = 2*PI*rayon
Aire = PI *rayon^2
```

```
Je suis un carre:
Périmètre = 4*cote
Aire = cote*cote
```

Exercice

- On souhaite réaliser un programme qui gère différents types d'animaux. Chaque objet animal est caractérisé par une couleur, un poids et un nom. Il peut également décrire sa façon de se déplacer de manger et de crier. En particulier, on peut enregistrer dans ce programme des chiens, des canards et des chats.

Tracer le diagramme de classe pour ce programme

Implémenter le en C#.