

# **Orienté Objet :**

## **Les classes abstraites en C#**

# Déclarer des classes abstraites

```
abstract class Token
{
    ...
}
class Test
{
    static void Main( )
    {
        new Token( );
    }
}
```

***Mot clé  
abstract***

Une classe abstraite ne peut  
jamais être instanciée

# Ce qu'est une classe abstraite

- Les classes abstraites ne sont pas instanciables, on ne peut instancier qu'une sous-classe concrète.
- Cette sous-classe concrète doit donner un corps à toutes les méthodes abstraites.
- Les classes abstraites sont déclarées par le modificateur abstract.
- Une classe abstraite peut:
  - Contenir ou hériter de méthodes abstraites(des méthodes sans corps)
  - Contenir des constantes
  - Avoir des méthodes normales (avec corps)

# Ce qu'est une classe abstraite

Classe abstraite abstractChien

```
public abstract class abstractChien
{
```

```
    protected int age;
    protected String couleur;
```

//On donne l'implémentation qui est commune à tous les chiens

```
public void vieillir()
{
    age = age +1;
}
```

//Cette méthode n'est définie que par les classes filles. Elle est donc laissée abstract

```
public abstract void aboyer();
```

```
}
```

## Exemple concret

- On sait que la méthode vieillir sera implémentée de la même manière, quelle que soit l'implémentation de Chien.
- Plutôt que d'implémenter cette méthode à chaque fois, on va factoriser le code dans une classe abstraite et étendre cette classe quand le besoin s'en fait sentir.
- On crée donc une classe AbstractChien qui n'implémente que la méthode vieillir, les autres étant laissées abstract.

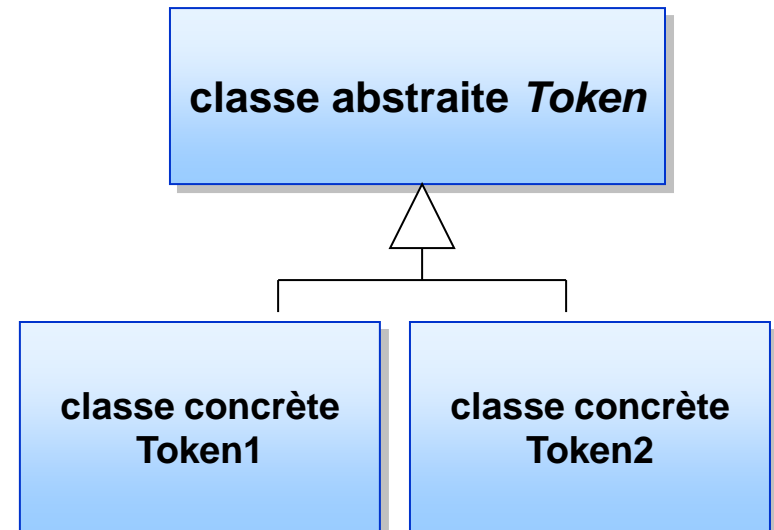
# Utiliser une classe abstraite dans une hiérarchie de classes

## Exemple

```
abstract class Token
{
    public virtual string Name( )
    {
        ...
    }
    ...
}

class Token1: Token
{
    ...
}

class Token2: Token
{
    ...
}
```



# Implémenter des méthodes abstraites

```
abstract class Token
{
    public virtual string Name( ) { ... }
    public abstract int Length( );
}
class CommentToken: Token
{
    public override string Name( ) { ... }
    public override int Length( ) { ... }
}
```

- Seule une classe abstraite peut déclarer des méthodes abstraites
- Les méthodes abstraites **ne doivent jamais** avoir de corps

## Où sont les bugs ?

```
class First
{
    public abstract void Method( );
}
```

```
abstract class Second
{
    public abstract void Method( ) { };
}
```