Chapitres

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

Héritage et polymorphisme

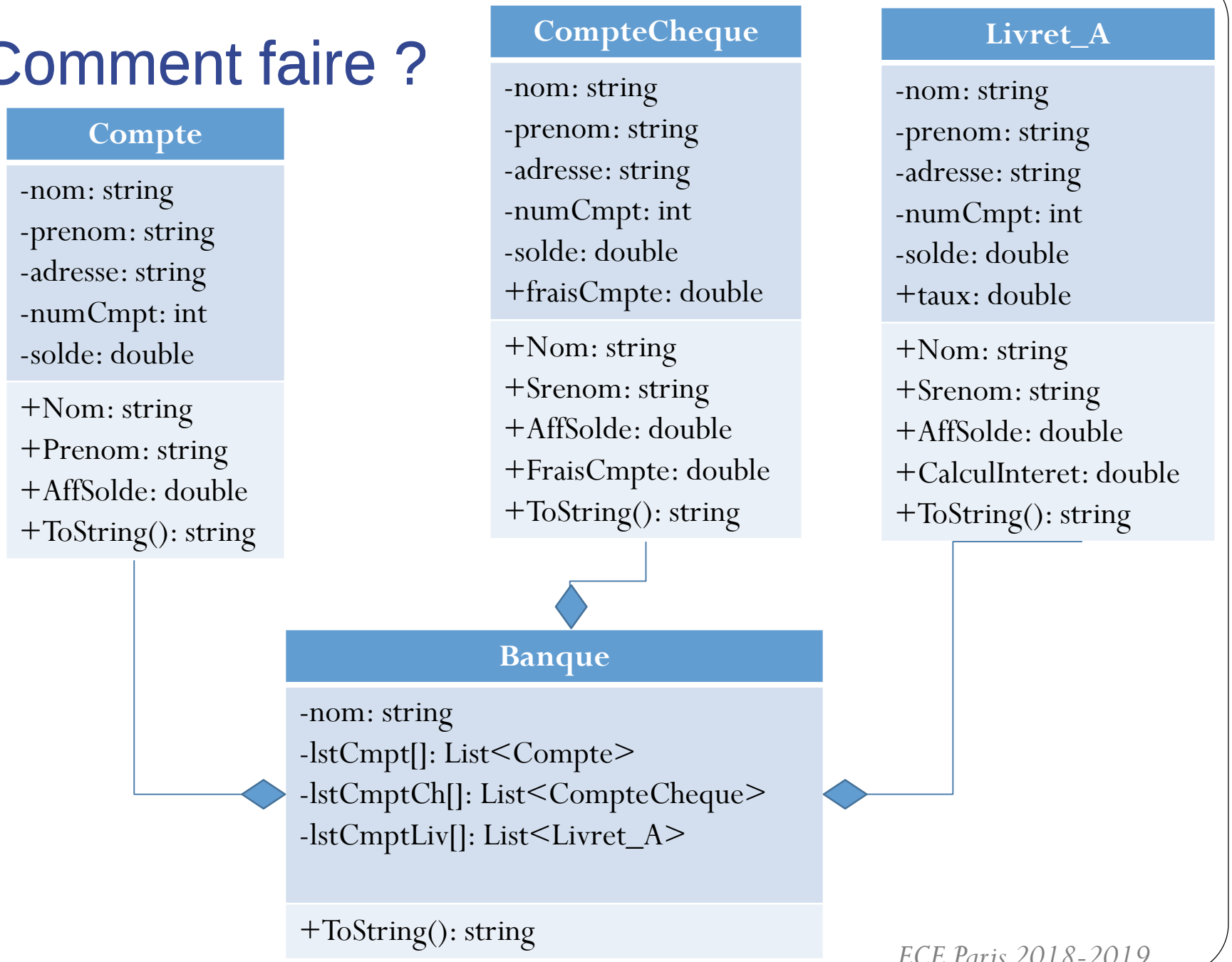
Exercice

- On se propose de créer un POO pour la gestion des comptes d'une banque. Un compte est caractérisé par un numeroCompte, un nomClient, un prénomClient, une adresseClient et un Solde. La banque propose à ses clients deux types de compte. Le premier type de compte, « CompteChèque », ne rapporte pas d'intérêt. Cependant, ce type de compte nécessite des frais de tenu de compte. Le deuxième type de compte, « Livret_A », génère des intérêts chaque mois selon un taux prédéfini.

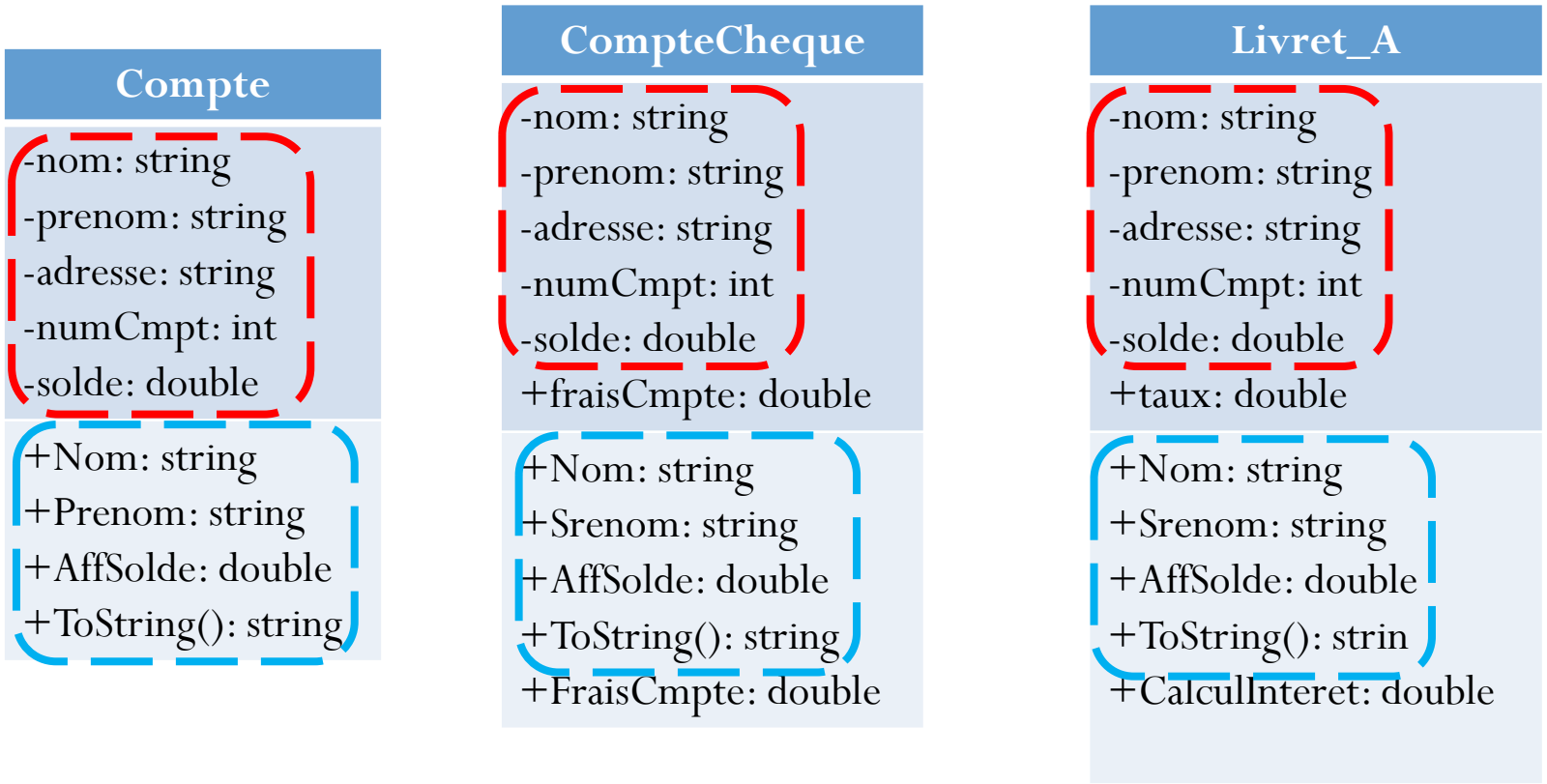
Exercice

- Créer le diagramme de classe pour ce programme sachant que chaque compte devrait être capable d'afficher le nom et le prénom du client ainsi que le solde du compte. Pour les compteChèque, on veut avoir la possibilité de modifier les frais de tenu de compte. Enfin, il faut créer pour chaque type de compte une méthode ToString() qui permet de décrire le compte.
- Traduisez le diagramme de classe en langage C#.

Comment faire ?

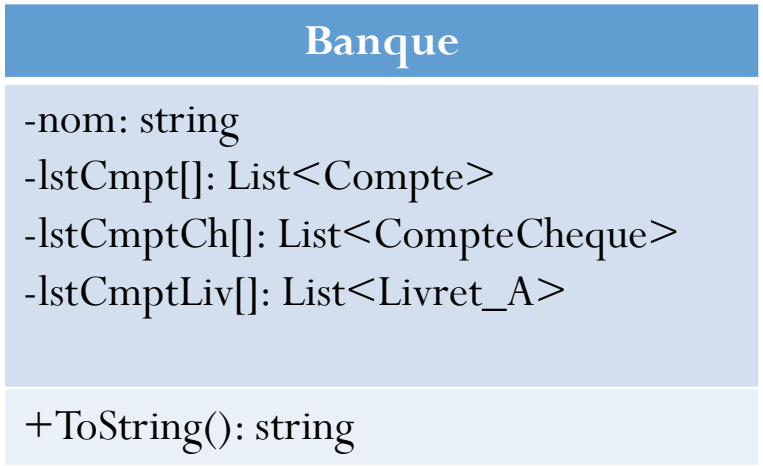


Limites de la méthode



- Répétition de blocs d'attributs et de méthodes (une redondance de code lié au recouvrement des deux concepts).

Limites de la méthode



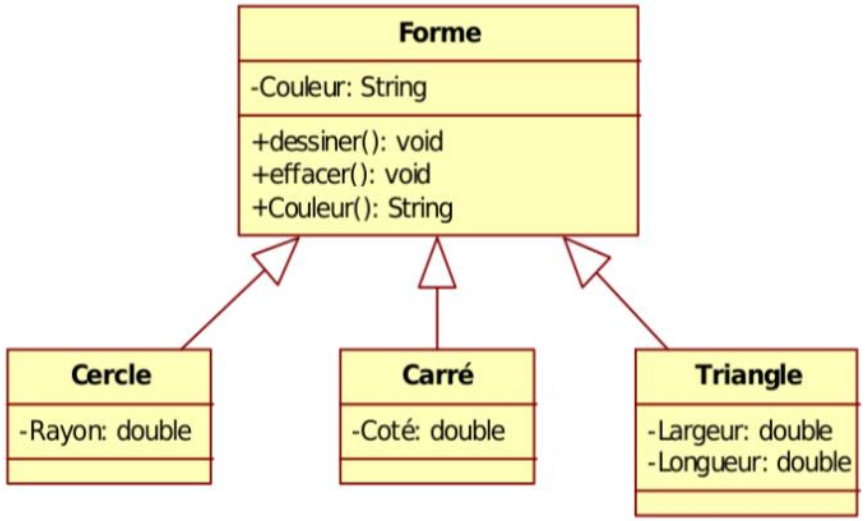
- Répétition de blocs d'attributs et de méthodes.
- Plusieurs listes de compte au lieu d'avoir une seule liste*
- On peut dire que les CompteChèque sont des cas particuli-ers des Compte (spécialisation).
- On traduit la notion d'héritage par « est un ». Un CompteCheque est un Compte avec certaines spécifications.

L’héritage

- L'héritage est un mécanisme permettant de créer une nouvelle c lasse à partir d'une classe existante en lui proférant ses pro-priétés et ses méthodes.
- La classe qui hérite est dite classe fille ou sous-classe ou classe dérivée.
- La classe dont on hérite est dite classe mère.
- La classe fille possède au moins tous les attributs et méthodes de la classe mère.
- La classe mère **ne** possède **PAS** tous les attributs et méthodes de la classe fille.
- En C#, une classe ne peut hériter que d’une seule classe mère

L'héritage

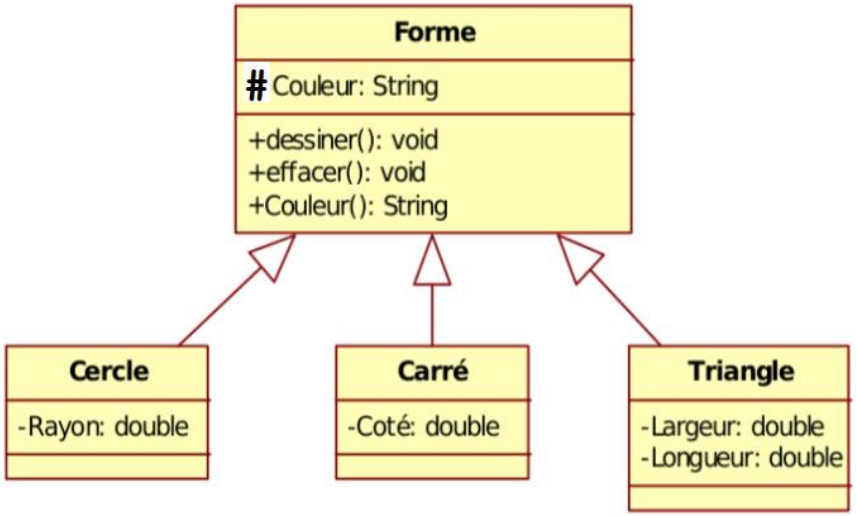
- **Représentation UML**



- Les classes Cercle, Carré et Triangle héritent de la classe Forme.
- Elles ont des attributs propres.
- La classe Cercle accède à toutes les méthodes (et les attributs) publiques, mais pas privées de la classe Forme.
- Toutefois, les attributs privés de Forme ont bien été créés pour Cercle, Carré et Triangle. Ils restent accessibles par des propriétés.

L'héritage

- **Représentation UML**
- **Visibilité des attributs**



- Pour rendre un attribut d'une classe accessible par ses classes filles, on utilise la portée « protected » (protégé) noté en # dans le diagramme des classes.
- Dans l'exemple, l'attribut protected Couleur de Forme est bien créé pour Cercle, Carré et Triangle. Ces classes filles accèdent à l'attribut Couleur sans l'intermédiaire d'une propriété.

L'héritage

- **Représentation UML**
- **Visibilité des attributs**
- **Implémentation en C#**

```
class Forme
{
    //Attributs
    protected string couleur;
    //Constructeur
    public Forme(string couleur)
    {
        this.couleur = couleur;
    }
}
```

```
class Cercle:Forme
{
    //Attribut
    private double rayon;
    public Cercle(string couleur, double rayon):base(couleur)
    {
        this.rayon = rayon;
    }
}
```

- Forme est la classe mère et Cercle est la classe fille.
- Pour constuire un objet de typeCercle, il faut définir la valeur du rayon mais aussi définir la couleur de la forme.

L'héritage

- **Représentation UML**
- **Visibilité des attributs**
- **Implémentation en C#**

```
class Forme
{
    //Attributs
    protected string couleur;
    //Constructeur
    public Forme(string couleur)
    {
        this.couleur = couleur;
    }
}
```

```
class Cercle:Forme
{
    //Attribut
    private double rayon;
    public Cercle(string couleur, double rayon):base(couleur)
    {
        this.rayon = rayon;
    }
}
```

- Si un autre constructeur de la classe mère possède des arguments, il faut explicitement l'appeler avec l'instruction «base » tout en ajoutant les paramètres dans l'ordre.

L'héritage

- **Représentation UML**
- **Visibilité des attributs**
- **Implémentation en C#**

```
class Forme
{
    //Attributs
    protected string couleur;
    //Constructeur
    public Forme()
    {
        this.couleur = "rouge";
    }
}
```

```
class Cercle:Forme
{
    //Attribut
    private double rayon;
    public Cercle(double rayon)
    {
        this.rayon = rayon;
    }
}
```

- Si la classe mère ne possède pas de constructeur, c'est le constructeur par défaut (sans paramètres) qui est appelé.

L'héritage

- **Représentation UML**
- **Visibilité des attributs**
- **Implémentation en C#**

- Une classe fille peut fournir une nouvelle définition d'une méthode d'une classe mère. Il s'agira non seulement de méthodes de même nom, mais aussi de même signature et de même type de valeur de retour.

L'héritage

- **Représentation UML**
- **Visibilité des attributs**
- **Implémentation en C#**

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

```
class Cercle:Forme
{
    //Attribut
    private double rayon;
    public Cercle(string couleur, double rayon):base(couleur)
    {
        this.rayon=rayon;
    }
    //Méthodes
    override
    public string ToString()
    {
        return base.ToString()+ "Il s'agit d'un cercle de rayon: " +rayon+"\n";
    }
}
```

```
static void Main(string[] args)
{
    Cercle c1 = new Cercle("blue", 15);
    Console.WriteLine(c1.ToString());
    Console.ReadKey();
}
```

```
E:\Enseignement_2018_2019\ECE\TDs\Heritage_Cours\
La forme est de couleur: blue
Il s'agit d'un cercle de rayon: 15
```

Exercice

- Créer le diagramme de classe pour l'exercice de gestion de compte en utilisant les notions d'héritages.
- Traduisez le diagramme de classe en langage C#.