



# CLASSES ET OBJETS

Les **classes** sont définies comme un ensemble d'**attributs** (variables) et de **méthodes** liées à un domaine spécialisé.

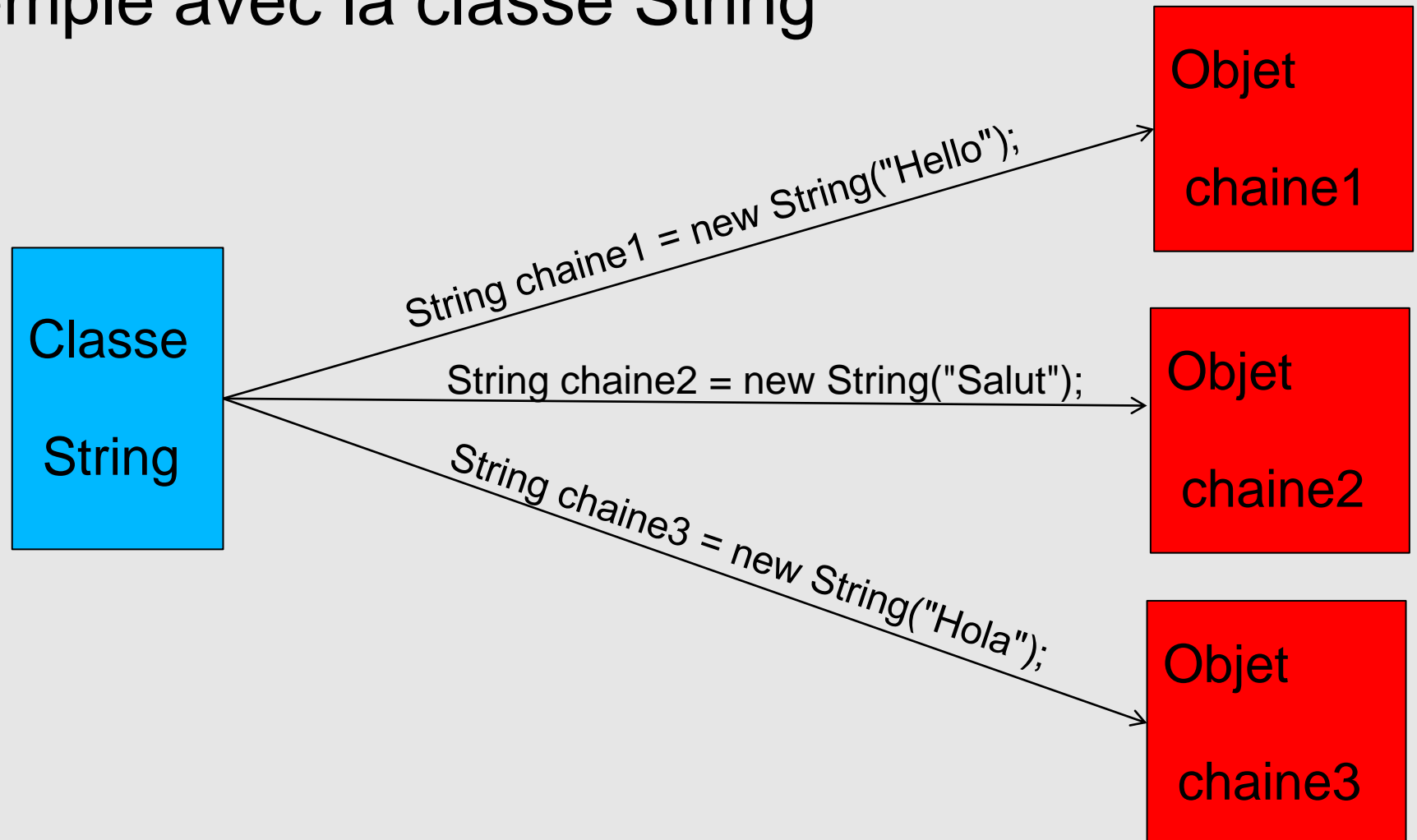
Elles peuvent être considérées comme des modèles permettant de créer des **objets**.

Chaque objet est construit avec la même structure et évolue différemment.



# CLASSES ET OBJETS

Exemple avec la classe String





# CLASSES ET OBJETS

Rappels sur la création d'objet:

L'opérateur **new**

- Réserve de la mémoire pour contenir l'objet.
- Créé l'objet dans cette zone mémoire selon le modèle de la classe.
- Associe une référence à cet objet
- Copie cette référence dans la variable associée à l'objet



# CLASSES ET OBJETS

La construction de classes permet de :

- Créer des ressources réutilisables dans plusieurs programmes sans réécriture.
- Créer et manipuler des objets dans le programme



# CLASSES ET OBJETS

## Exemple de création de Classe

### Programme de calcul de périmètre

```
public class CalculsGeometriques{  
    public static void main (String args[]){  
        double rayon;  
        double perimetre;  
        rayon = In.readDouble();  
        perimetre = calculerPerimetre(rayon);  
        System.out.println("perimetre = " +perimetre);  
    }  
  
    public static double calculerPerimetre (double leRayon){  
        double perimetreCalcule;  
  
        perimetreCalcule =2 * leRayon * Math.PI;  
        return perimetreCalcule;  
    }  
}
```



# CLASSES ET OBJETS

## Exemple de création de Classe

### Création de la classe Cercle:

**Attributs :**

**Le rayon :**

double leRayon

**Méthodes :**

**Calcul du périmètre :**

double calculerPerimetre ()



# CLASSES ET OBJETS

## Exemple de création de Classe

Creation de la classe Cercle:

```
public class Cercle{  
    public double leRayon;  
  
    public double calculerPerimetre (){  
        double perimetre;  
        perimetre = 2 * Math.PI * leRayon;  
        return perimetre;  
    }  
}
```

Diagram annotations:

- Modificateurs d'accès (points to `public` in `public class Cercle{`)
- Attribut de la classe (points to `double leRayon;`)
- Méthode (points to `calculerPerimetre ()`)

la classe Cercle est créée dans un nouveau fichier Cercle.java



# CLASSES ET OBJETS

## Exemple de création de Classe

### Modification de la classe CalculGeometrique:

```
public class CalculGeometrique {  
    public static void main(String[] args) {  
        double rayon;  
        double perimetre;  
        Cercle monCercle = new Cercle();  
        System.out.print("\nRayon du cercle : ");  
        rayon = In.readDouble();  
  
        monCercle.leRayon = rayon;  
        perimetre = monCercle.calculerPerimetre();  
  
        System.out.println("le Perimetre du cercle de rayon : " + monCercle.leRayon +  
            " est : " + perimetre);  
    }  
}
```

Création de l'objet

Accès à l'attribut leRayon de l'objet monCercle avec l'opérateur .

Accès à la méthode de l'objet monCercle avec l'opérateur .





# CLASSES ET OBJETS

## Exercice d'application

- Construire la classe Cercle
- Construire la classe principale , créer un objet et faire des essais.
- Rajouter un deuxième objet de type Cercle, afficher les caractéristiques des deux objets.
- Rajouter une méthode pour calculer la surface, faire des essais.
- Changer le modificateur d'accès de l'attribut **Lerayon** de **public** à **private**. Que constatez vous?
- Rajouter une méthode pour comparer les rayons de deux objets de type Cercle: