QUALITÉ DE DÉVELOPPEMENT

UML VERS JAVA

🎓 2A - Bachelor Universitaire de Technologie
🏛 IUT d'Orsay - Université Paris-Saclay - 2024/2025

Idir AIT SADOUNE
idir.ait-sadoune@universite-paris-saclay.fr

PLAN

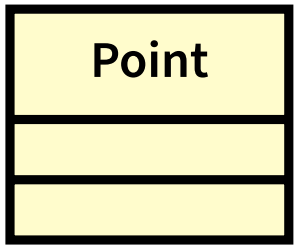
Retour au plan - Retour à l'accueil

PLAN

Retour au plan - Retour à l'accueil

TRANSCRIPTION DES CLASSES

A chaque **classe UML** correspondra une **classe Java**.

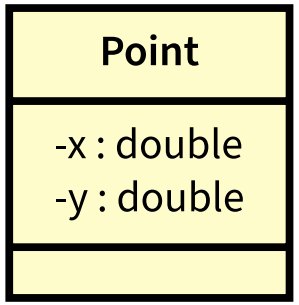



```java
class Point{
    ...
}
```

TRANSCRIPTION DES CLASSES

A chaque **attribut** d'une **classe UML** correspondra un **attribut** d'une **classe Java**.

- cardinalité $\leq 1 \rightarrow$ `Type nomAttribut;`
- cardinalité $> 1 \rightarrow$ `Collection<Type> nomAttribut;`



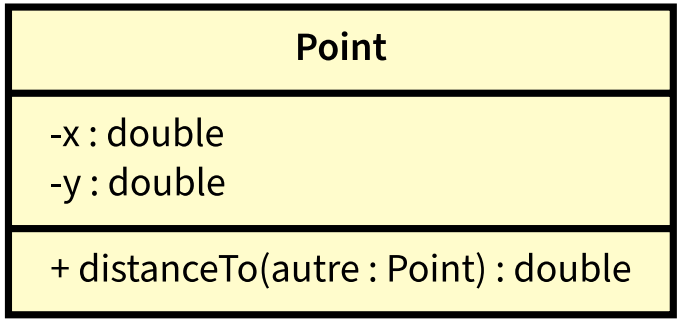

```java
class Point{
    private double x;
    private double y;
}
```

REMARQUE

Les visibilités des éléments de modélisation **UML** seront les mêmes dans leurs correspondants en **Java**.

TRANSCRIPTION DES CLASSES

A chaque **operation** d'une **classe UML** correspondra une **méthode** d'une **classe Java**.

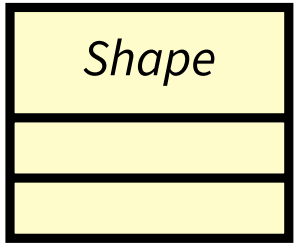



```java
class Point{
    private double x;
    private double y;

    public double distanceTo(Point autre){
        // TO DO
    }
}
```

TRANSCRIPTION DES CLASSES

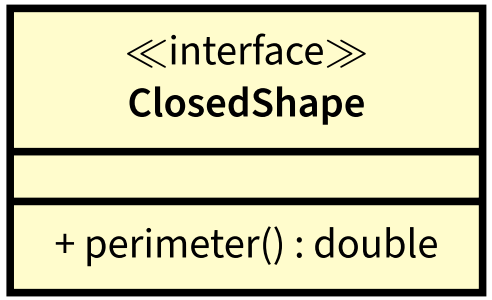
A chaque **classe UML abstraite** correspondra
une **classe Java abstraite**.




```java
abstract class Shape{
  ...
}
```

TRANSCRIPTION DES CLASSES

A chaque **interface UML** correspondra une **interface Java**.




```java
interface ClosedShape{
  public abstract double perimeter();
}
```

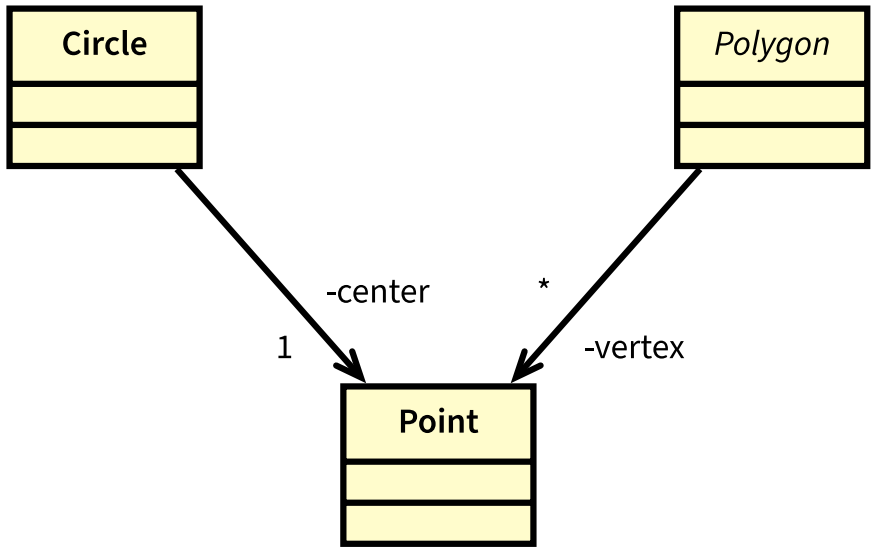
PLAN

Retour au plan - Retour à l'accueil

TRANSCRIPTION DES ASSOCIATIONS

- A chaque **association UML** correspondra **un attribut** dans les **classes Java** participantes à l'association.
- L'**attribut** est ajouté uniquement si l'association est navigable.
- Le **nom de l'attribut** est le nom de l'extrémité navigable.
  - cardinalité $\leq 1 \rightarrow$ `Type nomExtremite;`
  - cardinalité $> 1 \rightarrow$ `Collection<Type> nomExtremite;`

TRANSCRIPTION DES ASSOCIATIONS



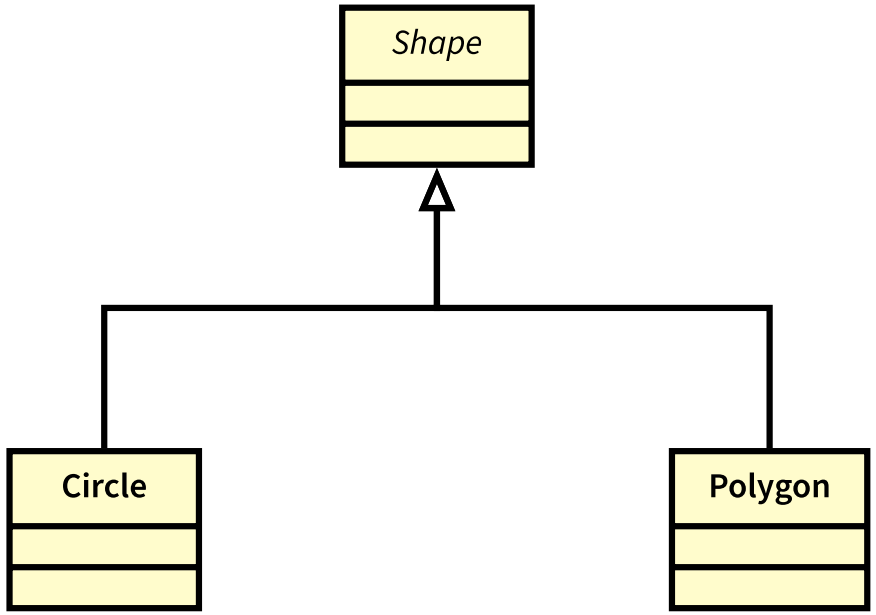

```java
class Point{
  ...
}
```

```java
class Circle{
  private Point center ;
}
```

```java
abstract class Polygon{
  private Collection<Point> vertex;
}
```

TRANSCRIPTION DES ASSOCIATIONS

La relation de **généralisation en UML** est traduite
par une relation d'**héritage en Java**.



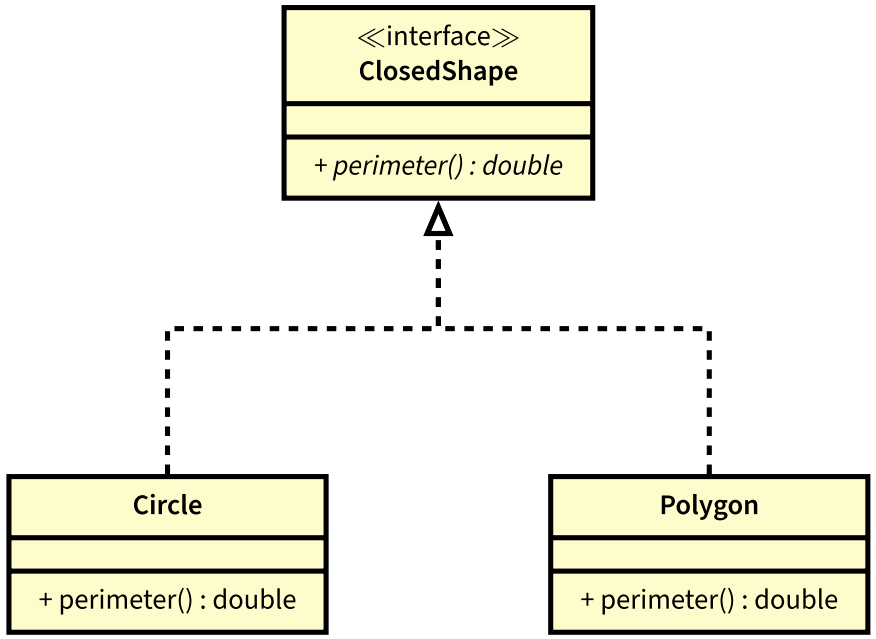

```java
abstract class Shape{
  ...
}
```

```java
class Circle extends Shape{
  ...
}
```

```java
class Polygon extends Shape{
  ...
}
```

TRANSCRIPTION DES ASSOCIATIONS

La relation de **réalisation en UML** est traduite
par une relation d'**implémentation en Java**.




```java
interface ClosedShape{
  public abstract double perimeter();
}
```

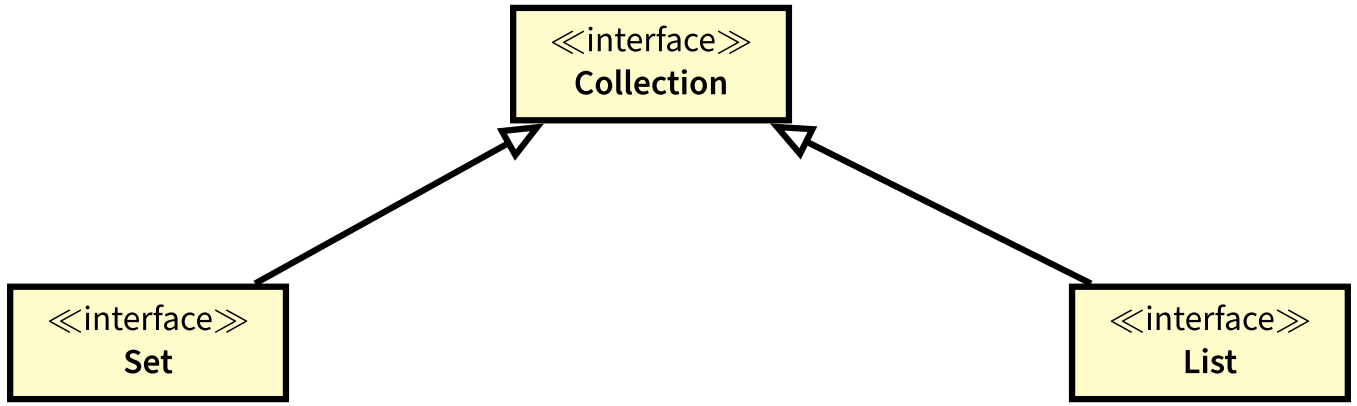
```java
class Circle implements ClosedShape{
  public double perimeter(){
    // TO DO
  }
}
```

```java
class Polygon implements ClosedShape{
  public double perimeter(){
    // TO DO
  }
}
```

PLAN

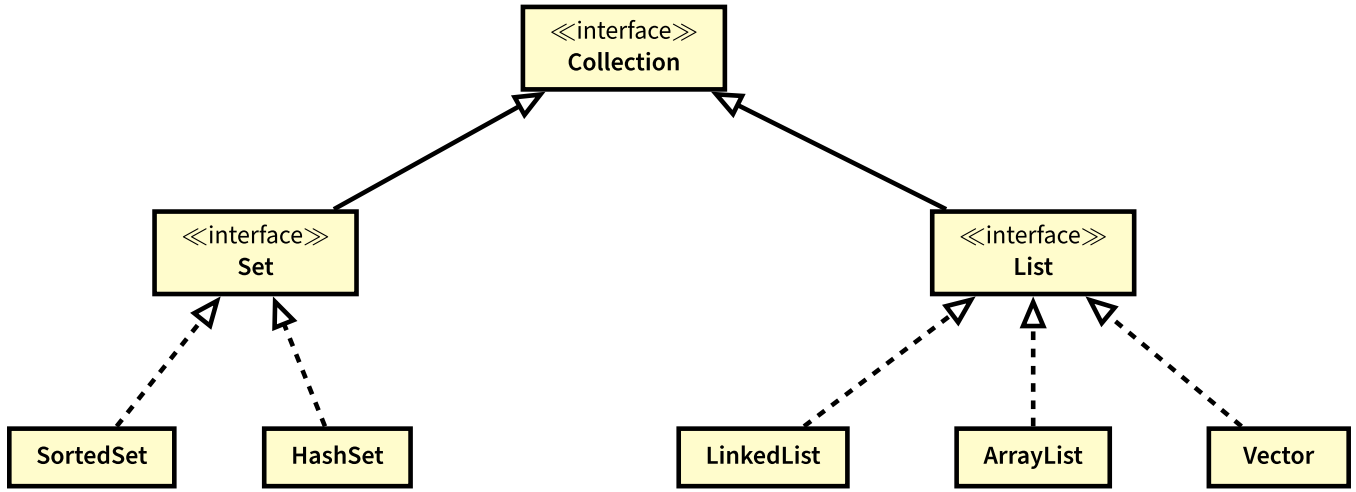
Retour au plan - Retour à l'accueil

LES COLLECTIONS EN JAVA



- `Collection` → interface qui gèrent toutes les collections dynamiques.
- `Set` → interface pour des collections qui n'autorisent pas de doublons.
- `List` → interface pour des collections qui autorisent des doublons et un accès direct à un élément.
- ...

LES COLLECTIONS EN JAVA



- `Vector` → implémente l'interface `List` (synchronisé)
- `ArrayList` → implémente l'interface `List` (non synchronisé)
- `LinkedList` → implémente l'interface `List` (doublement chaînée)
- `HashSet` → implémente l'interface `Set`
- `SortedSet` → implémente l'interface `Set` (ordonné)

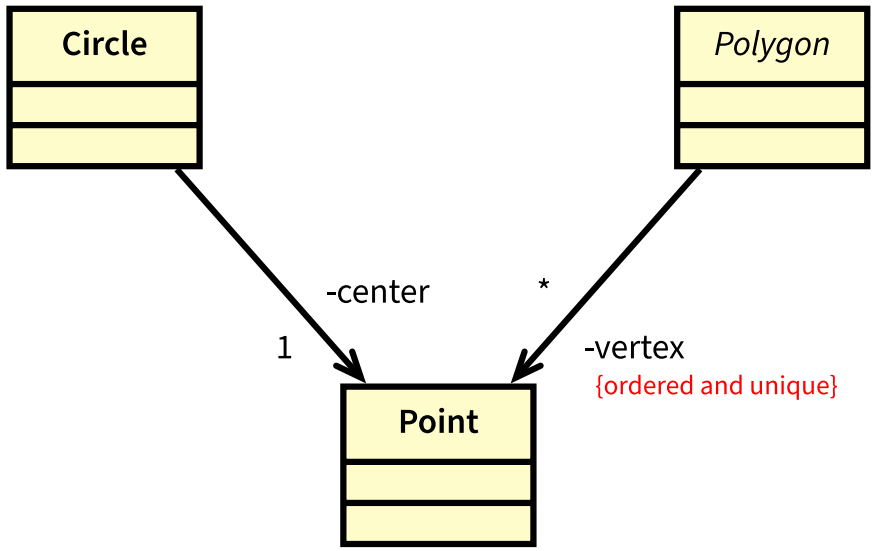
TRANSCRIPTION DES COLLECTIONS

Le type de la `Collection` affecté à un attribut en **Java**, dépend des propriétés de l'attribut/association **UML**:

- aucune propriété $\longrightarrow$ à configurer manuellement.
- ordered $\longrightarrow$ traduit en `ArrayList` ou `LinkedList`.
- unique $\longrightarrow$ traduit en `HashSet`.
- ordered et unique $\longrightarrow$ traduit en `SortedSet`.

TRANSCRIPTION DES COLLECTIONS

EXEMPLE




```
1 class Point{
2   ...
3 }
```

```
1 class Circle{
2   private Point center ;
3 }
```

```
1 abstract class Polygon{
2   private SortedSet<Point> vertex;
3 }
```

MERCI

Version PDF des slides

Retour à l'accueil - Retour au plan