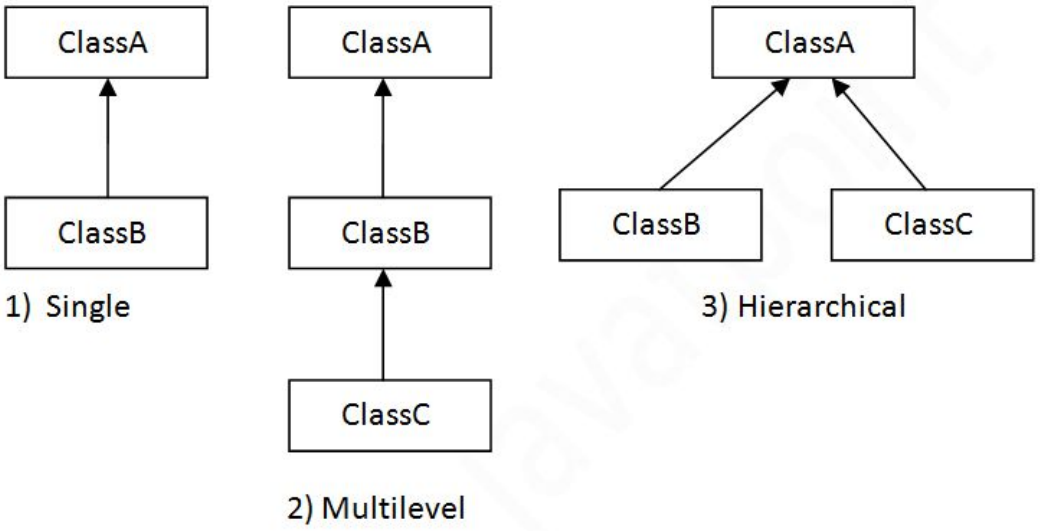
PAO - Laborator 4

Mostenire

- Mecanismul prin care o clasa preia structura şi comportamentul unei alte clase la care se adaugă elementele specifice.
- **Clasa părinte** - clasa de la care se mosteneste structura şi comportamentul.
- **Clasa copil** - clasa care preia structura şi comportamentul clasei părinte.



- Fata de alte limbaje, Java nu suporta moştenirea multiplă.

```
class nume_subclasa extends nume_superclasa {
        //conținut specific subclasei
}
```

Cuvantul cheie **super**:

- Este folosit pentru a accesa variabile şi metode ascunse în clasa părinte (o mai numim şi clasa de baza)
- o clasa copil (sau derivata) foloseste un constructor al clasei de baza pentru a iniţializa toate datele moştenite din clasa de baza

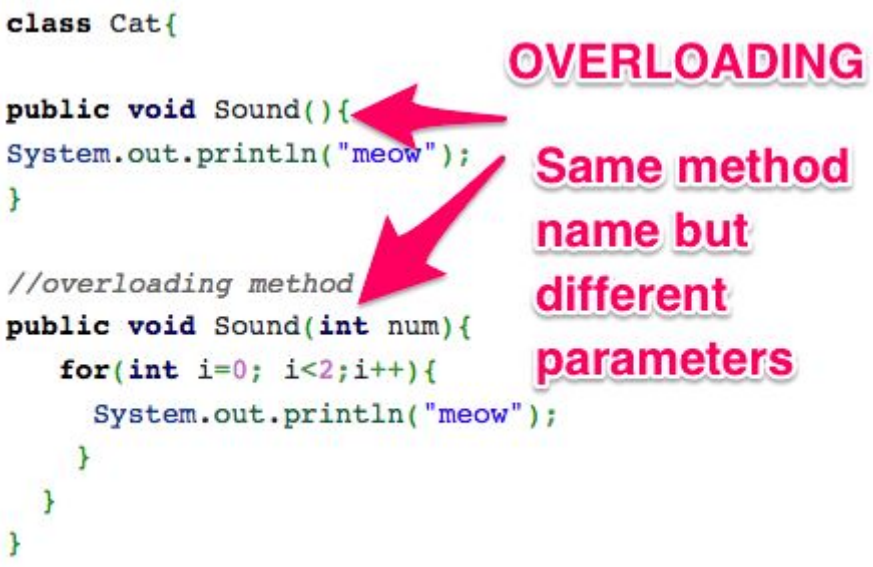
- Cand invocăm, în clasa copil, un constructor al clasei părinte folosind cuvantul cheie **super**, acest apel trebuie sa fie pe prima linie din constructorul clasei copil

```java
public class Shape {

    private String color;

    public Shape(String color) {
        this.color = color;
    }
}
```

```java
public class Triangle extends Shape {

    public Triangle(String color) {
        super(color);
    }
}
```

Polimorfism

- Se referă la proprietatea obiectelor de a avea mai multe forme.
- Limbajul java permite definirea a doua tipuri de polimorfism:
    - **tipul de supraîncărcare/parametric** - mecanismul prin care putem defini o metodă cu acelaşi nume în aceeaşi clasă, funcţii care trebuie să difere prin numărul şi/sau tipul parametrilor.

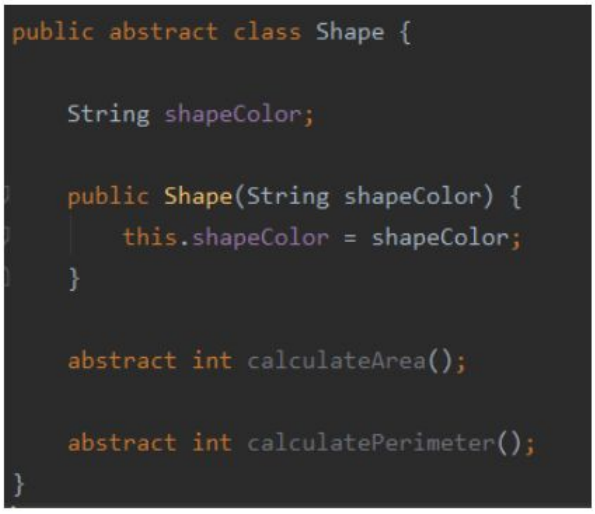


    - **tipul de supradefinire** - mecanismul prin care o metodă din clasa de bază este definită cu aceiaşi parametri în clasele derivate.

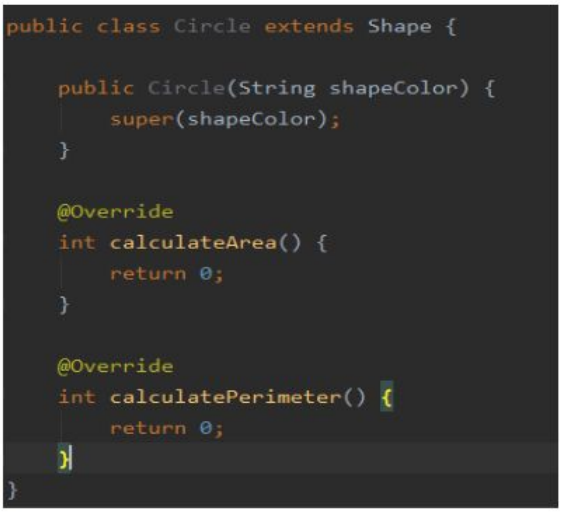
Supraîncărcare	Suprascriere
Trebuie să aibă cel puţin două metode cu acelaşi nume	Trebuie să aibă cel puţin o metoda cu acelaşi nume în clasa părinte şi clasa copil
Semnătura metodei trebuie sa fie diferită	Semnătura metodei trebuie sa fie aceiasi
Dacă numărul parametrilor este acelaşi, trebuie sa fie de tipuri diferite	Tipul parametrilor trebuie sa fie acelaşi

Clase abstracte

- O metoda sau o clasa abstracta se declara folosind cuvantul cheie **abstract**
- O clasa care contine cel puţin o metoda abstractă trebuie sa fie abstractă
- Dintr-o clasa abstracta nu se poate instantia nici un obiect
- Fiecare subclasă a unei clase abstracte care va fi folosită pentru a instantia obiecte trebuie sa ofere implementări pentru toate metodele abstracte din superclasa



```java
public abstract class Shape {

    String shapeColor;

    public Shape(String shapeColor) {
        this.shapeColor = shapeColor;
    }

    abstract int calculateArea();

    abstract int calculatePerimeter();
}
```



```java
public class Circle extends Shape {

    public Circle(String shapeColor) {
        super(shapeColor);
    }

    @Override
    int calculateArea() {
        return 0;
    }

    @Override
    int calculatePerimeter() {
        return 0;
    }
}
```

Clasa Object

- toString
- getClass
- hashCode
- equals