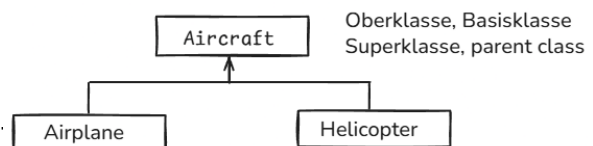




Vererbung

Eines der Kernprinzipien der objektorientierten Programmierung, die Vererbung (inheritance), erlaubt es existierenden Code wiederzuverwenden oder eine bestehende Klasse zu erweitern.

Klassen können von anderen Klassen abgeleitet werden und erben damit alle Eigenschaften und Methoden der Oberklasse/Basisklasse/Superklasse/parent class.



In Java kann eine Klasse nur von einer Oberklasse aberben. (einfach Vererbung)
in Java:

```
public class Airplane extends Aircraft {}
```

super :)

Zum Ausführen der ursprünglichen Methode in der abgeleiteten Klasse wird in der überschreibenden Methode das Schlüsselwort "super" verwendet.

```
class BaseClass {
    public BaseClass() {
        System.out.println("Code
in BaseClass#BaseClass");
    }
    public void method() {
        System.out.println("Code i
```

```
class SubClass extends BaseClass {
    public SubClass() {
        super();
        // zusätzliches Verhalten;
        System.out.println("Code
in SubClass#SubClass");
    }
```

```

n BaseClass#method"):
    }
    public String otherMethod() {
    }
}

```

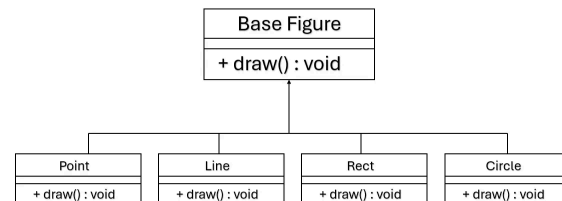
```

@Override
public void method(){
    super.method();
    //zusätzliches Verhalten:
    System.out.println("Code
in SubClass#method");
}
public String additionalMetho
d(){
    //zusätzliches Verhalten
    SubClass sb = new SubC
lass();
    sb.method();
}
}

```

Vererbung - abstrakte Methoden

Problem: Was soll eine draw-Methode bei einer nicht konkreten geometrischen Figur (Base Figure) zeichnen?



Man könnte eine leere Methode einfügen, dies führt zu der Problematik, dass die Klasse instanzierbar ist, obwohl es nicht sinnvoll ist, da die geometrische Form nicht festgelegt ist.

Genau in solche Fällen eignet sich abstrakte Basisklassen. Diese erlauben es, Methoden vorzugeben, die von Subklassen zu implementieren sind.

Dazu nutzt man Methoden ohne
Implementierung

Diese besitzen deswegen auch keine Block mit Programmzeilen, allerdings eine spezielle Kennzeichnung durch das Schlüsselwort `abstract`. Man spricht auch von abstrakten Methoden.

```
public abstract class BaseFigure{
    public abstract void draw();
    public void otherMethod(){
        ...
    }
}
public class Rect extends BaseFigure {
    public void draw() {
        // Code zum Zeichnen des Rechtecks
    }
}
public class Circle extends BaseFigure {
    public void draw() {
        // Code zum Zeichnen eines Kreises
    }
}
```