

3.2 Vererbung

Das Spiel Snake macht nur dann Sinn, wenn die Schlange sich von alleine bewegt und nicht jedes mal durch Ausführen der Methode „Bewegen()“ per Hand bewegt wird. Außerdem sollte dem User ermöglicht werden ins Spielgeschehen einzugreifen (Steuerung).

Dieses Prinzip gilt quasi für jedes Spiel: Ohne Bewegung und Steuerung ist es witzlos.

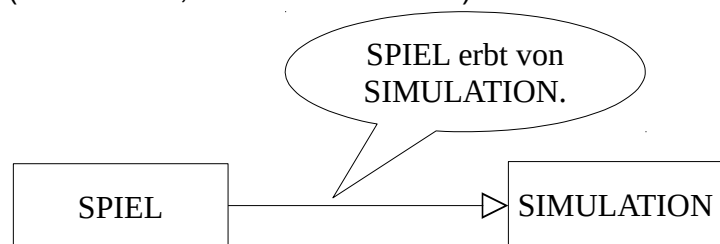
Im Game Design ist es üblich, sogenannte „Game Engines“ zu verwenden, die immer wiederkehrende Aufgaben der Spiele übernehmen. In unserem Fall haben wir die Klasse SIMULATION, die einen Taktgeber enthält und Tastatureingaben abfragen kann. Rechts sehen wir die Methoden der Klasse SIMULATION.

SIMULATION
Starten ()
Anhalten ()
TaktdauerSetzen (int dauer)
TaktImpulsAusfuehren ()
TasteGedrueckt (char welche)

Um diese Funktionalitäten in unserer Klasse SPIEL zu nutzen können wir die Fähigkeiten und Eigenschaften an unsere Klasse weitergeben. Man nennt dieses Prinzip Vererbung.

Bei der Vererbung übernimmt die erbende Klasse (Unterklasse, hier: SPIEL) alle Attribute und Methoden der vererbenden Klasse (Oberklasse, hier: SIMULATION).

Im **Klassendiagramm** zeichnet man das so:



Die Vererbung wird in Java mit dem Schlüsselwort `extends` in der Klassendefinition angegeben. Im Konstruktor muss zuerst der Konstruktor der Oberklasse mit dem Befehl `super()` aufgerufen werden, damit die Attribute und Methoden der Oberklasse geladen sind.

Bei den Methoden gibt es die Möglichkeit bestehende Methoden zu überschreiben, indem man eine Methode mit dem gleichen Bezeichner und gleichen Parametern wie in der Oberklasse deklariert.

Somit kann eine Unterklasse die Fähigkeiten der Oberklasse verändern und Eigenschaften hinzufügen.

In Java sieht das dann so aus:

„extends“ zeigt die Vererbung an.

Code:

```
public class SPIEL extends SIMULATION
{
    //zusätzliche Attribute

    public SPIEL()
    {
        super();
        //Initialisierung der Attribute der Klasse SPIEL
    }

    //... jetzt kommen die Methoden
    void TaktImpulsAusfuehren()
    {
        Bewegen();
    }
}
```

super() ruft den Konstruktor der Oberklasse auf

Überschreibt die Methode mit gleichem Bezeichner und Parametern in der Oberklasse

Aufgaben:

1. Die Klasse SPIEL

- a) Lege die Klasse SPIEL an, die ein Attribut schlange für die Referenz auf die Schlange vereinbart und dieses Attribut im Konstruktor dann initialisiert.
- b) Implementiere die Methode TaktImpulsAusfuehren, welche die Methode Bewegen der Schlange ausführt.
- c) Lege ein Objekt der Klasse SPIEL an und teste das Bewegen der Schlange durch den Aufruf der Methode Start.

2. Die Methode TasteGedrückt

- a) Implementiere die Methode TasteGedrückt mit Hilfe der Mehrfachauswahl. Beschränke dich vorerst auf die Tasten ‚S‘ und ‚P‘ und teste das Starten und Pausieren der Bewegung.
- b) Ergänze nun auch die Funktionalität dass man die SCHLANGE den Kopf drehen lassen kann, indem man eine Taste drückt. Teste wieder.

Tipp: Pfeiltasten oder WASD Steuerung!

Taste	Wert
→	KeyEvent.VK_RIGHT
←	KeyEvent.VK_LEFT
↑	KeyEvent.VK_UP
↓	KeyEvent.VK_DOWN