

# Die Klasse Spiel

- Die Klasse SPIEL stellt uns ein **Programm-Fenster** zur Verfügung in dem wir beliebige Grafiken anzeigen und bewegen können.
- Die Klasse SPIEL stellt uns außerdem einen **Taktgeber** (Ticker – Methode tick()) zur Verfügung. Der Ticker ist so eingestellt, dass in der Methode tick() angegebene Befehls-Abläufe automatisch immer wieder in einem frei wählbaren Zeitintervall ausgeführt werden. So kommt Bewegung ins Spiel.
- Die Klasse SPIEL stellt uns natürlich auch Möglichkeiten zur Verfügung um auf **Tastatur- oder Maus-Ereignisse** reagieren zu können. So kommt Interaktion ins Spiel.

## Konstruktoren

Du kannst entweder das Standard-Fenster verwenden oder dir selbst eines gestalten. Hierfür gibt es mehrere Konstruktoren.

### Standard-Konstruktor

```
SPIEL ( )
```

erzeugt ein Fenster der Breite 800 Pixel und der Höhe 600 Pixel. Das Fenster ist leer und es gibt keinen Mauszeiger. Jede gedrückte Taste auf der **Tastatur** wird im Konsolen-Fenster quittiert.

### Komplexer Konstruktor

Es gibt noch einen zweiten Konstruktor, mit dem du dir einiges einstellen kannst:

```
SPIEL( int breite, int hoehe, boolean punkteLinks, boolean punkteRechts, boolean maus )
```

Mit den Parametern **breite** und **hoehe** kannst du die Größe des Spielfelds definieren.

Die Wahrheitswerte **punkteLinks** und **punkteRechts** ermöglichen dir das Einblenden einer Punkteanzeige mit dem Wert **true**. Die Anzeige wird mit dem Wert **false** ausgeblendet.

Mit **true** beim Wahrheitswert **maus** wird der Mauszeiger im Fenster gefangen und angezeigt. Man muss dann das Fenster mit der Escape-Taste der Tastatur schließen.

**Alle anderen Konstruktoren sind Mischformen und erwarten nur einen Teil der Parameter des komplexen Konstruktors.**

**Die Klasse SPIEL wird verwendet, indem man davon erbt.  
Lasse deine „Haupt-Klasse“ davon erben.**

## Das Ticker-System

Deine Klasse erbt von SPIEL die Methoden

`tick()`

wird regelmäßig automatisch aufgerufen.

Soll etwas anderes geschehen als „tick tack“, so musst du diese Methode in deiner Klasse überschreiben:

```
@Override
public void tick() {
    // dein Code
}
```

`tickerIntervallSetzen( int zeitabstand )`

setzt das Zeitintervall (in Millisekunden) zwischen den automatischen Aufrufen von `tick()`.

`tickerStoppen()`

stoppt den automatischen Aufruf von `tick()`.

`tickerNeuStarten( int zeitabstand )`

startet den automatischen Aufruf von `tick()` wieder.

Der Übergabe-Parameter gibt das neue Zeitintervall (in Millisekunden) an.

## Auf Tastatur-Ereignisse reagieren

Deine Klasse erbt von SPIEL die Methode

`tasteReagieren( int taste )`

Immer wenn du auf eine Taste der Tastatur drückst, wird automatisch diese Methode aufgerufen. Ihr wird dabei automatisch ein Tastenkürzel übergeben, an dem du unterscheiden kannst, welche Taste gedrückt wurde. Soll etwas anderes geschehen als „Taste ... wurde gedrückt“, so musst du diese Methode in deiner Klasse überschreiben:

```
@Override
public void tasteReagieren( int taste ) {
    // dein Code
}
```



...

