

Snake03

Ein Schlangen-Objekt lässt sich nun wunderbar über die Tastatur steuern. Für unser besonderes Snake-Spiel benötigen wir gegnerische Schlangen-Objekte, die sich zufällig bewegen und natürlich nicht gegen die Wand laufen. Diagonale Bewegungen sind nicht zulässig und die gegnerische Schlange darf sich nicht rückwärts bewegen. Eine Kollision mit einer anderen Schlange ist noch nicht abzufangen. Erweitern Sie hierzu die Schlangen-Klasse um die Methode

```
public void setzeNeuesKopfElementZufaellig() { ...
```

Snake04

Schreibe eine Methode, die eine Kollision mit einer anderen Schlange überprüft.

```
public boolean kollision(Schlange s) {
```

Die Methode liefert „true“ bei einer Kollision und sonst „false“

Snake05

Schreibe eine Methode, die eine Kollision mit mehreren Schlangen überprüft.

```
public boolean kollision(Schlange[] schlangen) {
```

Die Methode liefert „true“ bei einer Kollision und sonst „false“.

Erzeuge im SnakeApplet ein SchlangenArray mit 8 AutoSchlangen und positioniere diese zufällig.

Überprüfe in der run()-Methode ob für die tastaturgesteuerte Schlange eine Kollision mit einer der 8 Autoschlangen vorliegt und beende ggf. das Spiel.

Snake06

Verhindere, dass die Schlange nicht in die Richtung laufen kann aus der sie kommt.

Snake07

Platziere zufällig 10 Herzen auf dem Spielfeld. Immer wenn die Schlange ein Herz erreicht verlängert sie sich um ein SnakeElement (Hinweis: Einfach letztes SnakeElement nicht löschen). Dies gilt auch für AutoSchlangen.

Snake08

Erweitere dein Applet um einen HighScore (Rang-Name-Punkte-Liste). Speichere die Daten in einer Textdatei. Verwende hierzu ein Servlet.