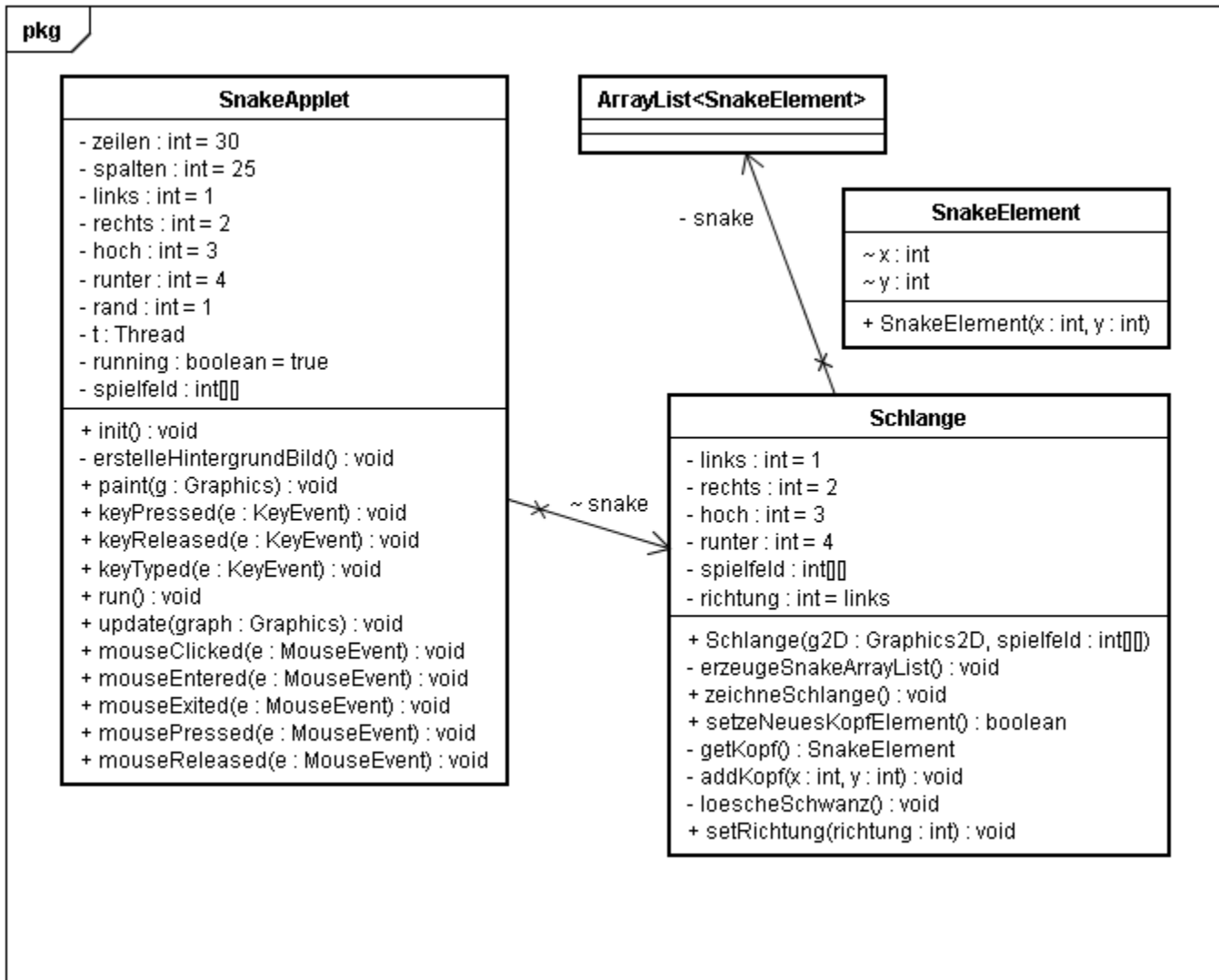


Snake 02

1. Ändern Sie die Klasse SnakeApplet wie im Quelltext unten gezeigt.
2. Führen Sie eine neue Klasse Schlange ein (siehe Klassendiagramm). Programmieren Sie die im Klassendiagramm gegebenen Methoden und testen Sie anschließend ihr Programm.

Hinweise zu den Methoden der Klasse Schlange:

1. Der Konstruktor speichert die Übergabewerte (Referenzen) in globalen Variablen vom Typ Graphics2D und int[][] ab. Außerdem ruft er die Methode `erzeugeSnakeArrayList()` auf.
2. Die Methode `erzeugeSnakeArrayList()` erzeugt die ArrayList und fügt dieser drei SnakeElemente hinzu. Positionieren Sie ihre Schlange in der Mitte des Applets.
3. `zeichneSchlange` nutzt das übergebene Graphics2D-Objekt zum Zeichnen. Sie können die Methode weitgehend aus unserem ersten Snake-Beispiel übernehmen.
4. Die Methode `setzeNeuesKopfelement()` setzt ein neues Kopfelement, wobei Kollisionen mit dem Rand zum Rückgabewert „false“ führen. Sie müssen hier den `switch(richtung)`-Teil aus Snake01 umsetzen/anpassen.



```

package snake02;

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.util.ArrayList;

public class SnakeApplet extends Applet implements KeyListener, MouseListener, Runnable{
    //Konstanten
    private final int zeilen = 30;
    private final int spalten = 25;

    private final int links = 1;
    private final int rechts = 2;
    private final int hoch = 3;
    private final int runter = 4;

    //private int richtung = links;
    private final int rand =1;

    //globale Variablen
    private Graphics2D g2D, g2DHintergrundBild;
    private Thread t;
    private boolean running = true;
    private Image buffer, hintergrundbild;
    private int[][] spielfeld; //Hier Rand,Kollisionselemente usw. eintragen
    Schlange snake;

    //Ähnlich wie ein Konstruktor, diese Methode wird zu Beginn des Applets aufgerufen
    public void init(){
        //Die Größe des Applets wird in der HTML-Seite eingestellt
        this.setSize(spalten*20,zeilen*20);
        //Listener registrieren
        this.addKeyListener(this);
        this.addMouseListener(this);

        //Beachte: Im 2D-Feld wurden spalten und dann Zeilen definiert!
        this.spielfeld = new int[spalten][zeilen];
        erstelleHintergrundBild();

        //Im Hintergrund auf ein Bild zeichnen,
        //welches nach Fertigstellung in den Vordergrund geschaltet wird
        if(buffer==null){
            buffer = createImage(this.getSize().width, this.getSize().height);
            g2D = (Graphics2D) buffer.getGraphics();
            snake = new Schlange(g2D, spielfeld);
        }
        running = true;
        t=new Thread(this); //Für neues Spiel neuen Thread starten
        t.start();
    }
}

```



```

public void keyReleased(KeyEvent e) {    }
@Override
public void keyTyped(KeyEvent e) {    }

//----- Spiel-Schleife -----
//Die Methode run() wird nach der sleep-Zeit
//(hier 200ms) ständig neu aufgerufen
@Override
public void run() {
    while(running){
        //Versucht ein neues Kopfelement zu setzen
        //liefert false, falls neues Element eine Kollision verursacht
        if(snake.setzeNeuesKopfElement()==false){
            System.out.println("Game Over");
            running=false;
        }

        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        //repaint() ruft update() auf!!!
        repaint();
    } //end while
} //end run

//-----
public void update(Graphics graph){ //wird von repaint() aufgerufen.
    paint(graph);                //Muss überschrieben werden!!
}                                // ruft paint( ) auf

//----- Maus - Ereignisse -----
@Override
public void mouseClicked(MouseEvent e) {
    //Mit einem Mausklick ins Applet wird ein neues Spiel gestartet
    running = false; //alten Thread beenden
    try {
        Thread.sleep(500);
    } catch (InterruptedException e1) {}
    snake = new Schlange(g2D, spielfeld);
    init();
}
//Diese Ereignisse spielen in unserem Programm keine Rolle und müssen deshalb
//nicht ausprogrammiert werden
@Override
public void mouseEntered(MouseEvent e) {}
@Override
public void mouseExited(MouseEvent e) { }
@Override
public void mousePressed(MouseEvent e) {}
@Override
public void mouseReleased(MouseEvent e) {}
//-----

```

```

} //End Class

```