

---

**Programmierzertifikat Objekt-Orientierung mit Java, Blatt Nr. 8**  
<http://proglang.informatik.uni-freiburg.de/teaching/java/2009/>

---

### 8.1 Aufgabe, Ein Malprogramm

```
package malen.proglang.informatik.unifreiburg.de;

import java.awt.event.KeyEvent;
import javax.swing.JFrame;

public final class World extends AWorld {

    public static void main(String args[]) {
        new World();
    }

    /**
     * This function is called by the World class.
     * Please create your main window here.
     */
    @Override
    protected void createAndShowGUI() {
        // Erzeugt ein neues Frame mit Titel: Tetris.
        this.frame = new JFrame("Tetris");

        // Erzeugt das Feld für das Tetrisspiel
        this.field = new Field();

        // Was passiert wenn das Fenster geschlossen wird?
        // -> Die Anwendung soll beendet werden
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Fügt das Feld dem Fenster hinzu.
        frame.add(this.field);

        // Packt die Elemente des Fensters passend
        frame.pack();

        // Zeigt das Fenster an
        frame.setVisible(true);
    }

    @Override
    public void keyPressed(KeyEvent e) {
        // TODO: Bitte hier auf Tasten reagieren
    }

    @Override
    public void mousePressed(MouseEvent e) {
        // TODO
    }

    @Override
    public void mouseReleased(MouseEvent e) {
        // TODO
    }
}
```

Diese Vorlage wurde Ihnen in der Vorlesung gezeigt. Sie finden den Code auf der Homepage des Lehrstuhls. Um sich mit Ihm vertraut zu machen erstellen Sie ein Program, dass folgendes Verhalten hat:

- Sie können mit der Maus Grafiken auf dem Fenster malen. Hierbei sollen Sie die linke Maustaste drücken, diese gedrückt halten, und wenn Sie die Maustaste loslassen, soll eine entsprechende Figur auf dem Fenster zu sehen sein.
- Falls Sie die Taste **k** drücken, können Sie mit der Maus eine Kreis zeichnen

- Falls Sie die Taste `r` drücken, können Sie mit der Maus ein Rechteck zeichnen
- Falls Sie die Taste `l` drücken, können Sie mit der Maus eine Linie zeichnen
- Falls Sie die Taste `n` drücken, sollen Sie eine neue Zeichfläche erhalten, d.h. es sollen alle Elemente gelöscht werden.

Achten Sie darauf, dass sie alle graphischen Objekte in einer Liste speichern, damit diese nicht verloren gehen, wenn das Fenster verdeckt wird.

## 8.2 Aufgabe, Tetris

Ziel: Ein Tetrispiel

1. Wir beginnen mit einer einfachen Version, in der alle Steine nur quadratisch sind, und sich nach unten bewegen. Hier soll es noch nicht möglich sein, die Steine mit den Tasten zu verschieben. Für diesen ersten Schritt beginnen wir mit folgender Vorlage:

Erstellen Sie Klassen, die Ihre Tetriswelt und die Informationen, die Sie benötigen werden, speichern können. Zeichnen Sie diese als Klassendiagramm auf. Fragen Sie nun einen Tutor, ob dieses Klassendiagramm sinnvoll ist, bevor Sie fortfahren.

Folgende Klassen und Interface sollten Sie erstellen (eventuell nicht vollständige Liste):

- a) Tetris (das gesamte Spiel)
  - b) Fild (das Spielfeld)
  - c) IBlock, Block (ein Block)
  - d) IStone, Stone (ein Spielstein, zusammengesetzt aus 4 Blöcken)
  - e) IDropBlock, DropBlock (Blöcke, die sich nach unten bewegen)
  - f) BlockList (die Liste der Blöcke des Feldes)
2. Schreiben Sie die Klasse `Block`, die einen Block darstellt, und diesen auf eine Canvas zeichnen kann. Testen Sie den Code, indem Sie eine `Block` in der Welt zeichnen.
  3. Schreiben Sie eine Klasse `DropBlock`, die von der Klasse `Block` abgeleitet ist, und die sich pro Zeiteinheit eine feste Anzahl Pixel nach unten bewegt. Hierzu sollte die Klasse ein `onTick` Methode erhalten. Binden Sie nun einen `Block` in Ihr Spiel ein, und rufen Sie die `onTick` Methode immer aus der `onTick` Methode der Weltklasse aus auf. Nun sollte Sie eine erste Version besitzen, in der eine `DropBox` nach unten fällt, bis sie nicht mehr zu sehen ist.
  4. Schreiben Sie einen Test, mit dem Sie sicherstellen, dass die Boxen nicht aus der Zeichenfläche fallen, und dass sie nicht übereinander zu liegen kommen. Fügen Sie hier zuerst der Klasse `Block` eine Methode hinzu, die prüft, ob sich zwei Blöcke schneiden. Schreiben Sie anschließend eine Methode, die für alle in dem Spielfeld vorkommenden Blöcke ein übereinanderliegen feststellt, d.h. `true` liefert, falls eine Überlagerung vorhanden ist, und `false`, falls es keine Überlagerung gibt.
  5. Falls eine Box den Boden oder eine andere Box erreicht hat, soll diese an der Stelle verweilen, und eine neue Box soll oben in der Mitte des Spielfelds erscheinen. Vergessen Sie nicht die Box, die nun fest ist, geeignet zu speichern. Sonst können Sie bei den folgenden Boxen keine Kollision abfragen oder testen, ob eine Reihe mit Boxen angefüllt ist.
  6. Fragen Sie ab, ob der Benutzer die Taste `rechts` oder `links` gedrückt hat, und verschieben Sie dann die Box, die sich nach unten bewegt, in die entsprechende Richtung.
  7. Falls eine Reihe voll ist, löschen Sie diese, und bewegen Sie alle Boxen eine Reihe nach unten.
  8. Vergeben Sie Punkte für gelöschte Reihen
  9. Erstellen Sie nun Steine, die andere Formen haben
  10. Schreiben Sie die Methode `rotate` für die Steineklasse.
  11. Zeigen Sie eine Vorschau an, damit der Spieler den nächsten Stein sehen kann
  12. ...