

# Informatik Übungsblatt 4

## Ziel der Stunde:

**Du solltest in der Lage sein, Fallunterscheidungen im Programmcode zu realisieren und Zustandsdiagramme zeichnen zu können.**

## Übungen zum Verständnis

- 1 M Verdeutliche am Beispiel des Vorgangs des Geldabhebens 3 verschiedene Situationen, in denen eine Fallunterscheidung vorgenommen werden muss.
- 2 S Welchen Vorteil bieten switch-case Blöcke im Gegensatz zu if-then-else Blöcken?

## Praktische Übungsmöglichkeiten

- 3 L Erweitere des Zustandsdiagramm der Fußgängerampel zu einem Zustandsdiagramm für eine übliche Straßenampel.

*Hinweis: Es ist keine Schande im Internet zu ersuchen, wann genau eine Ampel „rotgelb“ wird!*

Füge in der Klasse Rechteck im Projekt „verbessertes Zeichenfenster“ eine Funktion `boolean gleichesRechteck (Rechteck anderesRechteck)`

- 4 L hinzu, die prüft, ob das übergebene Rechteck gleich dem Objekt ist, das die Funktion ausführt und dann wahr zurückgibt. (2 Rechtecke sind bei gleichen x, y, länge und breite Werten gleich).

*Hinweis: schaue dir ggf. das 2. Video noch einmal an um zu wiederholen, wie man auf Attribute zugreift! Ergänze eine gleichartige Methode zur Übung ggf. auch beim Kreis und dem Dreieck*

Erstelle eine Klasse Ampel und lasse diese in ein Zeichenfenster zeichnen.

- 5 M Die Ampel soll eine Methode „umschalten“ haben, die in den nächsten Zustand der Ampel wechselt und die Farbe automatisch umschaltet.

Profis programmieren natürlich eine Ampel mit 3 Leuchten.

*Hinweis: Benutze hierfür das verbesserte Zeichenfenster, dort ist das Ändern einer Farbe einfacher.*

- 6 M Erstelle eine Funktion, die den Absolutbetrag einer Zahl berechnet.

- 7 S 

```
static double function (double d){
    return d % 1 == 0 ? d % 2 >= 0 ? d % 2 : - d % 2 : (int) d % 2 >= 0 ? (int) d % 2 : (int) - d % 2;
}
```

Obiger Code stellt eine gültige Java-Funktion dar. Erkläre, was sie berechnet und wie sie dies tut.

*Hinweise zum Lösen: „Ternärer Operator“, „Modulo-Operator“*

**Zeige die Funktionsfähigkeit deines Codes durch geeignete main-Methoden!**