

Übungsblatt 3- EIPNF WS 15/16

Besprechung des Übungsblatts erfolgt am 18.11. & 21.11.

Aufgabe 1:

Hier sehen wir den in der Vorlesung bereits besprochenen Code. Aufgabe ist es nun vier verschiedene drawAnimation Methoden zu schreiben, vier Objekte der Klasse Animation zu erzeugen, mit unterschiedlichen Charactern zu initialisieren und so auf vier Tasten vier verschiedene Animationen abrufen zu können.

```

class Animation {
    int animationCounter;
    char animationKey;

    public Animation (char c) {
        animationCounter = 0;
        animationKey = c;
    }

    void handleInput () {
        if (keyPressed && animationKey == key) {
            animationCounter++;
        } else {
            animationCounter = 0;
        }
    }

    void drawAnimation1 () {
        if (animationCounter>0) {
            // hier steht deine Animation
        }
    }

    void drawAnimation2 () {...}

    void drawAnimation3 () {...}

    void drawAnimation4 () {...}
}

// -----

Animation a;
// deklariere weitere Objekte

void setup () {
    size(600, 400);
    a = new Animation('1');
    // initialisiere alle Objekte
}

void draw () {
    background(0);
    a.handleInput();
    a.drawAnimation1();
    // führe für jedes Objekt die entsprechenden Animationen aus
}

```

Aufgabe 2:

Schreibe eine Klasse `Person` die die folgenden Felder und Methoden enthält. Alle Felder sollen im Konstruktor durch Parameter gesetzt werden können!

1. Felder: `name`, `geburtsJahr`
2. Methoden: `sagName`, `sagAlter`, `berechneAlter`

Die Methoden `sagName` und `sagAlter` geben beide Text in der Konsole aus. Dieser soll in etwa so erscheinen:

„name: Hallo ich heiÙe name.“

„name: Ich bin ... Jahre alt.“

Das Alter wird in der Methode `berechneAlter` ausgerechnet und zurückgegeben (Wir verwenden einfach immer 2016 als aktuelles Jahr).

Beachte: In einer Klasse definierte Methoden können innerhalb anderer Methoden der Klasse aufgerufen werden. Dies erfolgt einfach per Aufruf (ohne Punkt oder ähnliches).

Hinweis: Rufe in `sagAlter` die Methode `berechneAlter` auf um das Alter aus dem `geburtsJahr` zu ermitteln.

Man

```
class Person {
  // Felder

  // Konstruktor

  // Methoden
}

Person p;

void setup () {
  size(100,100);
  // initialisiere p mit dem Aufruf des Konstruktors
}

void draw () {
  // rufe auf dem Objekt p sagName() und sagAlter auf
  delay(1000);
}
```