# Informatik Aufgaben 05.05.2020

## b) Schleifen in Java

Die while-Schleife ist folgendermaßen aufgebaut:

```
while(Bedingung){
    Anweisung 1;
    Anweisung 2;
    [...]
}
```

Wenn der Programmablauf die Schleife erreicht hat, wird zunächst geprüft, ob die Bedingung wahr ist. Wenn dem so ist, werden die Anweisungen in dem durch die geschweiften Klammern angegebenen Anweisungsblock der Reihe nach ausgeführt. Wenn das Ende des Anweisungsblocks erreicht ist, wird erneut geprüft, ob die Bedingung wahr ist. Ist sie wahr, läuft die Schleife erneut durch, solange, bis die Bedingung falsch ist. Daraufhin wird der Anweisungsblock nicht erneut ausgeführt und das Programm wird nach der Schleife weiter ausgeführt.

## c)Bearbeitung der Aufgaben

### Aufgabe 1

```
public Grafik() {
    View view = new View (400,400, "Grafik1");
    int zaehler = 0;
    while (zaehler < 3) {
     Rectangle rechteck = new Rectangle(100, 100, 200, 200, Color.RED);
        rechteck.turn(zaehler * 30);
        zaehler = zaehler + 1;
    }
}
```

### Aufgabe 2

a)

```
void zaehler(int obergrenze) {
    zahl = 1;
    Text zahlText = new Text(100, 100,"0");
    while (zahl <= obergrenze){
        zahlText.setText(""+zahl);
        zahl = zahl + 1;
        view.wait(1000);
    }
}
```

b)
Die Funktion zaehler erhält als Parameter eine Obergrenze. Dann zählt die Funktion von 1 bis zu dieser Obergrenze im 1-Sekunden-Takt, indem bei der Position x = 100, y = 100 ein Text mit immer der aktuellen Zahl steht, die jede Sekunde durch die nächste Zahl ersetzt wird, bis die Obergrenze erreicht ist.

# Aufgabe 5

## a)

Zuerst wird mit

```
view = new View(400,400,"Ballon platzt");
view.setBackgroundColor(Color.YELLOW);
```

ein 400 x 400 großes Fenster mit gelbem Hintergrund erstellt.
Dann wird mit

```
peng = new Text(150, 200, "PENG", Color.GREEN);
peng.setFontMonospaced(true,40);
peng.setHidden(true);
```

ein grüner Schriftzug erstellt, der zunächst unsichtbar gemacht wird.
Zum Ende des Konstruktors wird dann mit

```
ballon = new Circle(150, 150, 50, Color.RED);
```

ein roter Kreis erzeugt, der an der Position 150, 150 steht und einen Radius von 50.

## b)

Die erste while-Schleife in der Methode *fuehreAus()* wartet solange eine Sekunde, bis eine Taste gedrückt wurde.

**c)**

```
import sas.*;
import java.awt.Color;
class Luftballon {
    View view;
    Circle ballon;
    Text peng;

    Luftballon() {
        view = new View(400,400,"Ballon platzt");
        view.setBackgroundColor(Color.YELLOW);
        peng = new Text(150, 200, "PENG", Color.GREEN);
        peng.setFontMonospaced(true,40);
        peng.setHidden(true);
        ballon = new Circle(150, 150, 50, Color.RED);
    }

    void fuehreAus() {
        while (!view.keyPressed()){
            view.wait(1);
        }
        while (ballon.getShapeWidth() < 400){
            ballon.scale(1.05, 1.05);
            view.wait(100);
        }
        view.remove(ballon);
        peng.setHidden(false);
    }

    static void main(){
        Luftballon ballonSimulation = new Luftballon();
        ballonSimulation.fuehreAus();
    }
}
```