SoSe 2018

Prof. Dr. Margarita Esponda

Objektorientierte Programmierung

1. Übungsblatt

Ziel: Auseinandersetzung mit logischen Ausdrücken, Schleifen und Funktionen in Python.

1. Aufgabe (4 Punkte)
Programmieren Sie eine Python-Funktion, die mit Hilfe einer **for**-Schleife nach Eingabe einer Liste mit Zahlen das Produkt aller Zahlen der Liste als Ergebnis der Funktion zurückgibt.

2. Aufgabe (6 Punkte)
Schreiben Sie eine Python-Funktion **echtTeiler**, die bei Eingabe einer natürlichen Zahl **n** die Liste aller Teiler von **n** berechnet. Die Zahl **n** ist kein echter Teiler von **n** und soll deswegen nicht in der Ergebnisliste vorkommen.

Anwendungsbeispiel:

```
>>> echtTeiler(250)
>>> [1, 2, 5, 10, 25, 50, 125]
```

3. Aufgabe (4 Punkte)
Zwei natürliche Zahlen (m, n) werden als "Befreundetes Zahlenpaar" bezeichnet, wenn jede Zahl gleich der Summe der echten Teiler der anderen Zahl ist. Schreiben Sie eine Funktion in Python, die bei Eingabe zweier natürlicher Zahlen entscheidet, ob die Zahlen befreundet sind oder nicht.

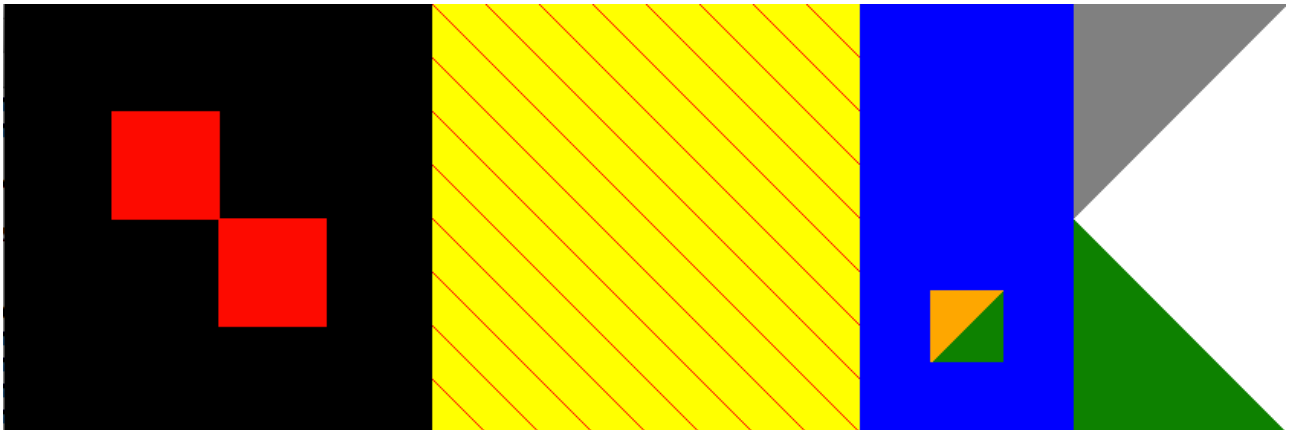
4. Aufgabe (15 Punkte)

In dieser Aufgabe soll der Inhalt von 5 Funktionen programmiert werden, die 5 der unten stehenden Bilder produzieren.

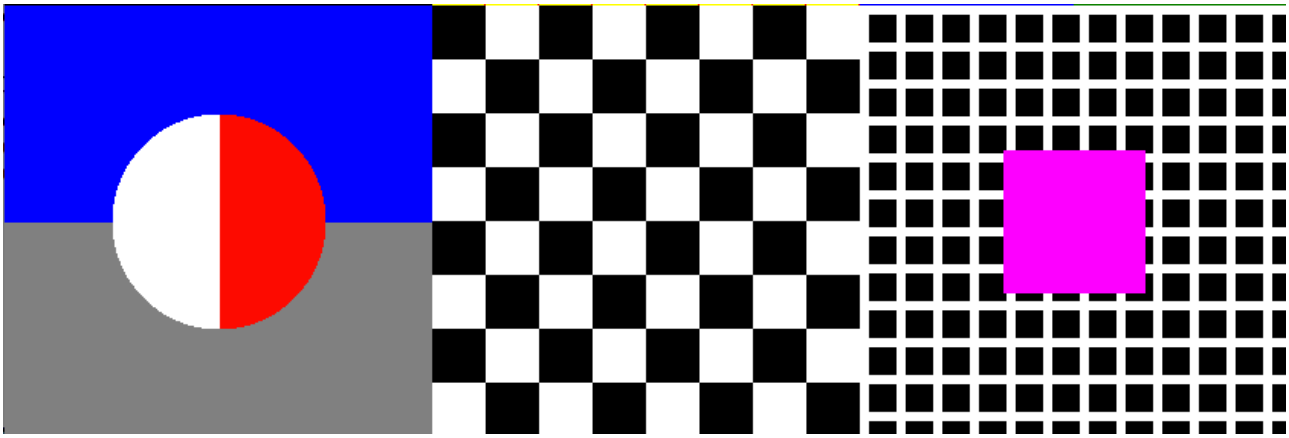
Die Funktionsrahmen sind in der Datei **MosaicFunctions.py** vorgegeben, und nur der Inhalt der Funktionen darf verändert werden.

Ein **MosaicFrames.py** Programm-Modul wird vorgegeben. Dieses Modul darf nicht verändert werden. Innerhalb des **MosaicFrames**-Moduls wird das Modul **MosaicFunctions.py** importiert und die verschiedenen Bilder in einem graphischen Fenster visualisiert.

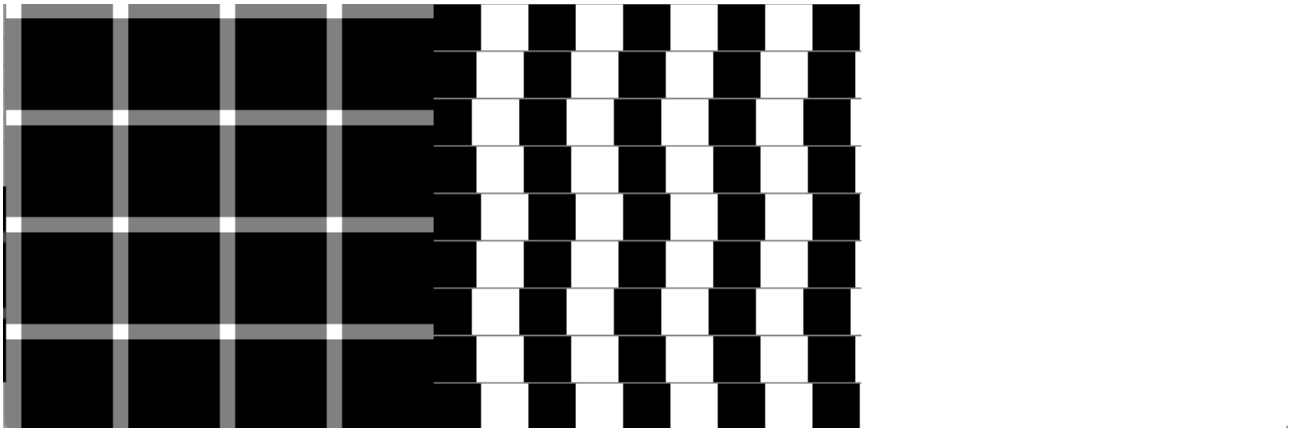
Innerhalb der zu implementierenden Funktionen dürfen keine Schleifen verwendet werden. Die Lösung benötigt nur **if-else**-Anweisungen, logische Ausdrücke und arithmetische Operationen.



decide_color_squares decide_color_diags decide_color_triangles



decide_color_circle decide_color_chess decide_color_illusion_1



decide_color_illusion_2 decide_color_illusion_3 decide_color_own_picture

Das Programm kann zum Testen der Funktionen mit dem Python-Modul **MosaicFrames.py** wie folgt gestartet werden:

```
Jython MosaicFrames.py
```

Für jedes zusätzliches Bild werden 2 Bonuspunkte vergeben, sodass insgesamt 8 Bonuspunkte möglich sind.

Wichtige Hinweise:

1) Verwenden Sie geeignete Namen für Ihre Variablen und Funktionen, die den semantischen Inhalt der Variablen oder die Funktionalität der Funktionen darstellen.

2) Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.

3) Kommentieren Sie Ihre Programme.

4) Verwenden Sie geeignete Hilfsvariablen und Hilfsfunktionen in Ihren Programmen.

5) Löschen Sie alle Programmzeilen und Variablen, die nicht verwendet werden.

6) Schreiben Sie getrennte Test-Funktionen für alle 6 Aufgaben für Ihren Tutor.

7) Die Lösungen sollen in Papierform und elektronisch (KVV-Upload) abgegeben werden.

Installationsanleitung für Python und Jython

Um die Übungszettel bearbeiten zu können, wird **Java 8**, **Jython 2.7.0** und **Python 3.6.5** benötigt.

Python kann von (https://www.python.org/downloads/) heruntergeladen werden. **Java 8** (JDK) kann von der Oracle-Seite heruntergeladen werden http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

Nachdem das JDK installiert ist, muss nur noch **Jython** heruntergeladen (http://www.jython.org/downloads.html) und mit folgendem Befehl direkt am Terminal installiert werden:

java -jar jython_installer-2.7.0.jar

Damit die Übungsaufgaben auf den Linux Rechnern ausgeführt werden können, muss die Jython Environment-Variable **"JAVA"** auf das JDK-10 gesetzt werden. Dies kann mit folgendem Befehl getan werden:

export JAVA = /usr/lib/jvm/java-8-openjdk-amd64/bin/java