

Aufgabensammlung zu Aufgabenblatt 1

Kompetenzstufe 1

Allgemeine Informationen zu den Aufgaben:

- Die Sammlung beinhaltet Aufgaben, die in den Übungen bzw. auch zu Hause gelöst werden können.
- Fragen dazu können in den Übungen bzw. den Programmiertutorien beantwortet werden.
- Erstellen Sie ein eigenes Projekt für die folgenden Aufgaben.

CharacterSearch

Erstellen Sie eine Klasse CharacterSearch und erweitern Sie die Methode main:

Implementieren Sie eine Verzweigungsstruktur, die bei einem gegebenen String überprüft, ob ein gegebenes Zeichen `character` vorkommt. Wenn das Zeichen `character` nicht im gegebenen String vorkommt, dann wird der String "Kein Zeichen gefunden" ausgegeben. Wenn das Zeichen `character` einmal im gegebenen String vorkommt, dann wird "Genau ein Zeichen gefunden" ausgegeben und wenn `character` mehr als einmal vorkommt, dann wird der String "Zwei oder mehr Zeichen gefunden" ausgegeben.

Beispiele:

"Schneeeulen" und `character = 'a'` liefert "Kein Zeichen gefunden"

"Schneeeulen" und `character = 'u'` liefert "Genau ein Zeichen gefunden"

"Schneeeulen" und `character = 'e'` liefert "Zwei oder mehr Zeichen gefunden"

Loops

Erstellen Sie eine Klasse **Loops** und erweitern Sie die Methode **main**:

- a) Schreiben Sie eine for-Schleife, die alle durch 4 oder 7 teilbaren Zahlen im Intervall [13, 68] aufsummiert und das Ergebnis auf der Konsole ausgibt.

Erwartetes Ergebnis: 812

- b) Schreiben Sie eine for-Schleife, die alle Zahlen im Intervall [10, 20] ausgibt. Nach jeder Zahl wird ein '=' ausgegeben. Beginnend bei 10 wird nach jeder dritten Zahl zusätzlich noch ein '*' ausgegeben.

Erwartetes Ergebnis: 10=*11=12=13=*14=15=16=*17=18=19=*20=

- c) Schreiben Sie eine for-Schleife, die alle Vorkommen des Zeichens '#' aus dem Satz **#Wichtige #Nachricht erhalten##** entfernt und dabei einen neuen String erzeugt, der auch ausgegeben wird.

Erwartetes Ergebnis: Wichtige Nachricht erhalten

Calculation

Erstellen Sie eine Klasse Calculation und erweitern Sie die Methode main:

Sie haben folgende Berechnungsvorschrift gegeben:

- Sie schreiben pro Zeile zwei Zahlen nebeneinander.
- Die erste Zahl wird pro Zeile halbiert (Reste abgerundet) und die Ergebnisse untereinander geschrieben, bis man zur 1 gelangt.
- Die zweite Zahl wird pro Zeile verdoppelt und die Ergebnisse untereinander geschrieben.
- Die zweite Zahl pro Zeile wird gestrichen, falls die erste Zahl gerade ist. Diese Information ist in der nachfolgenden Tabelle in der dritten Spalte vorhanden.
- Die Summe der nicht gestrichenen Zahlen ergibt das gesuchte Ergebnis.

Dazu haben Sie folgendes Beispiel mit den Zahlen 25 und 33 gegeben:

Erste Zahl	Zweite Zahl	Was wird addiert?
25	33	33
12	66	-
6	132	-
3	264	264
1	528	528

Daher werden 33, 264 und 528 addiert und das Ergebnis lautet: 825

Implementieren Sie ein Programm, das diese Berechnung für beliebige positive Zahlen (nicht nur 25 und 33) durchführt und das korrekte Ergebnis erzeugt und ausgibt.

Hinweis: Das obige Verfahren realisiert eine Multiplikation der beiden Zahlen.

Pattern

Erstellen Sie eine Klasse **Pattern** und erweitern Sie die Methode **main**:

- Erstellen Sie ein Ausgabefenster mit der Größe 400×400 Pixel.
- Deklarieren Sie eine Variable **n**, die die Anzahl an Rechtecken bestimmt.
- Zeichnen Sie von unten nach oben abwechselnd blaue, grüne und rote Rechtecke.
- Die Rechtecke werden nach oben hin gleichmäßig kürzer. Das unterste Rechteck hat die Breite 400. Die Höhe ergibt sich aus der Anzahl der Rechtecke.
- Bei jedem Rechteck wird genau im Mittelpunkt ein oranger Kreis gezeichnet, dessen Radius der halben Höhe eines Rechtecks entspricht. Beim obersten Rechteck entspricht der Radius des Kreises auch der halben Breite des Rechtecks.

Beispiel für Ausgaben für unterschiedliche Werte von **n** sind in der folgenden Abbildung ersichtlich:

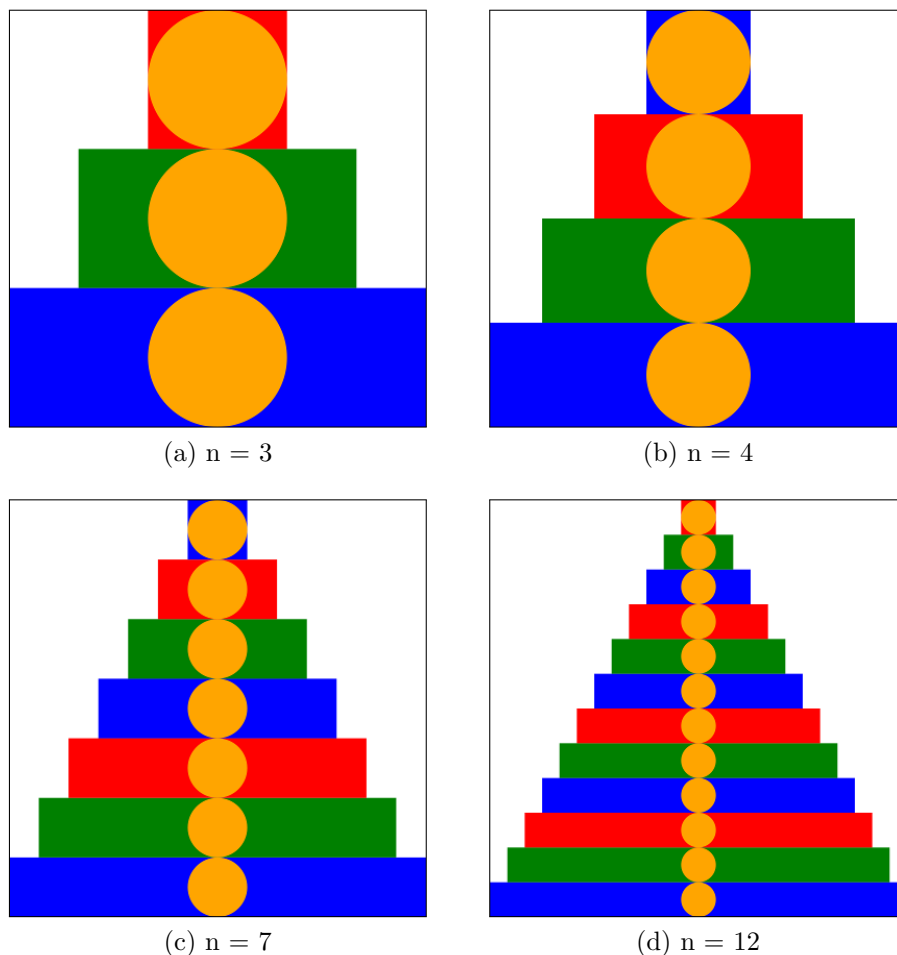


Abbildung 1: Verschiedene Ausgaben für unterschiedliche Werte von **n**.