

Aufgabenblatt 1

Kompetenzstufe 1

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Freitag, 02.11.2018 13:00 Uhr** in TUWEL hoch.
- Zusätzlich müssen Sie in TUWEL ankreuzen, welche Aufgaben Sie gelöst haben und während der Übung präsentieren können.
- Ihre Programme müssen kompilierbar und ausführbar sein.
- Ändern Sie bitte nicht die Dateinamen und die vorhandene Ordnerstruktur.
- Bei manchen Aufgaben finden Sie Zusatzfragen. Diese Zusatzfragen beziehen sich thematisch auf das erstellte Programm. Sie müssen diese Zusatzfragen für gekreuzte Aufgaben in der Übung beantworten können. Sie können die Antworten dazu als Java-Kommentare in die Dateien schreiben.
- Verwenden Sie, falls nicht anders angegeben, für alle Ausgaben `System.out.println()` bzw. `System.out.print()`.
- Verwenden Sie für die Lösung der Aufgaben keine Aufrufe (Klassen) aus der Java-API, außer diese sind ausdrücklich erlaubt, wie z.B. die Klassen `StdDraw` und `Scanner` oder Klassen, die in den Hinweisen zu den einzelnen Aufgaben aufscheinen.

In diesem Aufgabenblatt werden folgende Themen behandelt:

- Deklaration von Variablen (verschiedene Datentypen)
- Verwendung von Operatoren
- Umgang mit den Klassen `String` und `StdDraw`
- Verzweigungen (if-Anweisung, switch-Anweisung)
- Einfache Schleifen

Aufgabe 1

Erweitern Sie für Teilaufgabe A die Methode main:

- a) Deklarieren und initialisieren Sie je eine Variable mit den Werten `1234e-3f`, `1.23f`, `1.234e3`, `43.21`, `1234L`, `0xa`, `014`, `'Z'`, `' '` und stellen Sie sicher, dass die Variablen dieselben Typen wie diese Literale haben.
- b) Legen Sie zusätzlich zu den bereits deklarierten Variablen noch eine byte-Variable mit dem Wert 123, sowie eine short-Variable mit dem Wert 4321 an.
- c) Erzeugen Sie durch Verwendung des `+` - Operators einen String, der die Werte in den zuvor beschriebenen Variablen enthält, jeweils getrennt durch einen Beistrich (`,` `"`). Geben Sie den String mittels `System.out.println(...)` aus.
- d) Berechnen Sie das ganzzahlige Produkt aller Werte in diesen Variablen, wobei die Nachkommastellen vor der Multiplikation abgeschnitten werden sollen. Vermeiden Sie einen Überlauf des Ergebnisses. Geben Sie das Produkt mittels `System.out.println(...)` aus.
- e) Berechnen Sie die Summe aller Werte in diesen Variablen und wandeln Sie das Ergebnis am Ende der Berechnung in eine ganze Zahl um. Geben Sie die ganzzahlige Summe mittels `System.out.println(...)` aus.
- f) Wandeln Sie jeden Wert in den Variablen in einen Wert vom Typ `byte` um und berechnen Sie die Summe der umgewandelten Werte. Geben Sie die Summe mittels `System.out.println(...)` aus.

Zusatzfrage(n) zu Teilaufgabe A:

1. Warum ist der aus den Variablen erzeugte String nicht gleich zu den vorgegebenen Literalen `1234e-3f`, `1.23f`, `1.234e3`, `43.21`, `1234L`, `0xa`, `014`, `'Z'`, `' '`?
2. Wodurch ergeben sich die Unterschiede zwischen den beiden Summen?
3. Wann wird ein Wert automatisch in einen Wert eines anderen Typs umgewandelt, und wann muss eine Umwandlung explizit durchgeführt werden (*Cast*)?
4. Erzeugen Sie zusätzlich eine Variable `st` vom Typ `String` und initialisieren Sie diese mit `"Lehrveranstaltung"`. Führen Sie danach `System.out.println(st + i + c)` sowie `System.out.println(i + c + st)` aus, wobei `i` für eine Variable vom Typ `int`, und `c` für eine Variable vom Typ `char` steht. Warum sind die Ausgaben vor und nach dem String `"Lehrveranstaltung"` unterschiedlich?

Erweitern Sie für Teilaufgabe B die Methode main:

- ❗ Für die folgenden Aufgabenstellungen sind Methoden der Klasse `String` zu verwenden. Zusätzlich soll jedes Ergebnis mittels `System.out.println(...)` auf der Konsole ausgegeben werden. Zeigen Sie alle Funktionalitäten mit der vorgegebenen String-Variable `sentence`. Diese String-Variable wird dabei nicht verändert.

- a) Verwenden Sie eine Methode, die Ihnen das Zeichen an der Stelle mit dem Index 10 zurückliefert.
 - b) Verwenden Sie eine Methode, die Ihnen die Länge (Anzahl der Zeichen) des Strings zurückgibt.
 - c) Verwenden Sie eine Methode, die prüft, ob `sentence` leer ist.
 - d) Verwenden Sie eine Methode, die prüft, ob `sentence` mit "Auf" anfängt.
 - e) Verwenden Sie eine Methode, die Ihnen die Möglichkeit gibt, Substrings zu extrahieren. Geben Sie a) den Substring ab Index 27 und b) den Substring von Index 0 bis Index 13 aus.
 - f) Verwenden Sie eine Methode, die es Ihnen ermöglicht, Strings zu kombinieren. Verketteten Sie den String in der Variable `sentence` und den zuvor extrahierten String von Index 0 bis 13.
 - g) Verwenden Sie eine Methode, die jedes 'e' innerhalb des Strings durch ein 'X' ersetzt.
 - h) Verwenden Sie eine Methode, die die Zeichenkette "aus" durch die Zeichenkette "AUS" ersetzt.
 - i) Verwenden Sie eine Methode, die alle Buchstaben in einem String in Großbuchstaben umwandelt. Verwenden Sie zur Demonstration den extrahierten String von Index 27 bis 37.
 - j) Verwenden Sie eine Methode, die alle Buchstaben in einem String in Kleinbuchstaben umwandelt. Verwenden Sie zur Demonstration den String `sentence`.
- ❗ Der Link <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html> kann Ihnen dabei helfen, diese Aufgabe zu lösen.

Zusatzfrage(n) zu Teilaufgabe B:

1. Wie kann man einen Substring extrahieren, wenn die Methode `substring(...)` nicht zur Verfügung steht? Wie würden Sie in diesem Fall vorgehen?

Aufgabe 2

Erweitern Sie für Teilaufgabe A die Methode main:

- Implementieren Sie ein Programm, das ein gegebenes Gewicht (`int weight` in kg) einer Kategorie zuordnet. Die Kategorisierung der Gewichte ist in Tabelle 1 aufgelistet. In der dritten Spalte finden Sie den String, der bei einem bestimmten Gewicht ausgegeben werden soll.
- Realisieren Sie diese Gewichtskategorisierung mittels if-Anweisungen auf zwei verschiedene Arten:
 1. Verschachtelung einfacher if-Anweisungen (ohne else-if)
 2. Mehrfachverzweigung (mit else-if)

Kategorie	Gewichtsintervall	Ausgabestring
Extra Leicht	< 5kg	"Gewicht --> Extra Leicht"
Leicht	5kg – 9kg	"Gewicht --> Leicht"
Mittelschwer	10kg – 24kg	"Gewicht --> Mittelschwer"
Schwer	25kg – 49kg	"Gewicht --> Schwer"
Extra Schwer	≥ 50kg	"Gewicht --> Extra Schwer"

Tabelle 1: Gewichtskategorisierung mit den dazugehörigen Ausgabestrings.

Zusatzfrage(n) zu Teilaufgabe A:

1. Darf eine if-Anweisung ohne else-Zweig verwendet werden?

Erweitern Sie für Teilaufgabe B die Methode main:

- Implementieren Sie einen Zahlenumrechner, der für eine hexadezimale Zahl `int hexNumber` die dazugehörige Binärzahl ausgibt.
- Verwenden Sie für die Implementierung eine switch-Anweisung.
- Es sollen die Zahlen von 0x0 bis 0xF auf 0000 bis 1111 abgebildet werden können.
- Bei einer ungültigen hexadezimalen Zahl soll ein entsprechender Hinweistext ausgegeben werden, ansonsten "<hexNumber> entspricht <binaryNumber>". Beispiel: 0xA entspricht 1010.

❗ Für die Formatierung der hexadezimalen Zahl können Sie den Aufruf `String.format("0x%X", hexNumber)` verwenden.

Zusatzfrage(n) zu Teilaufgabe B:

1. Darf man generell das `break` in den einzelnen case-Zweigen weglassen?
2. Muss bei der Verwendung der switch-Anweisung ein default-Zweig implementiert werden?
3. Sind if- und switch-Anweisungen gegeneinander austauschbar?

Aufgabe 3

Erweitern Sie die Methode `main`:

- Implementieren Sie ein Programm, welches das in Abbildung 1 gezeigte Bild ausgibt. Im Bild sind alle Maßangaben vorhanden, die notwendig sind, um dieses Bild zu erzeugen.
 - Setzen Sie die Größe des Ausgabefensters auf 400×400 Pixel. Der Punkt $P(x = 0, y = 0)$ befindet sich in der linken unteren Ecke.
 - Für die Realisierung der vertikalen Ringe mit der Teilung in Blau und Grün wurde die Methode `StdDraw.arc(...)` verwendet. Verwenden Sie für die Ringe und auch horizontalen Kreise geeignete Schleifen.
- ⚠ Die Linienbreite wurde für das gesamte Bild mit `StdDraw.setPenRadius(...)` auf 0.005 gesetzt.
- ⚠ `StdDraw.setPenColor(StdDraw.ORANGE)` setzt die aktuelle Zeichenfarbe auf Orange. Andere Farben können analog dazu gesetzt werden.
- ⚠ Die Dokumentation unter dem Link <https://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html> kann Ihnen dabei helfen, diese Aufgabe zu lösen.

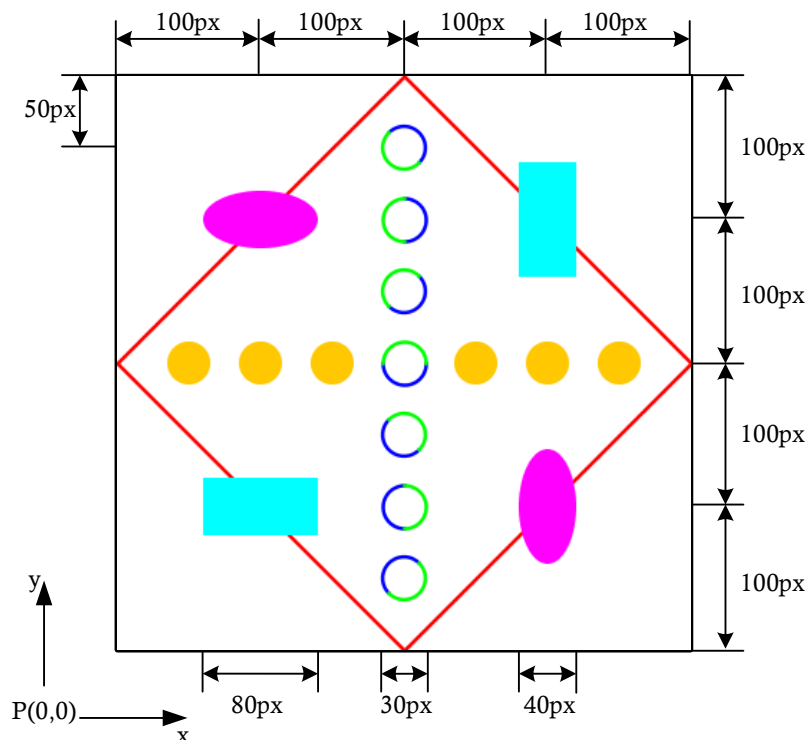


Abbildung 1: Ergebnisbild mit den entsprechenden Maßangaben

Zusatzfrage(n):

1. Sind verschiedene Arten von Schleifen gegeneinander austauschbar?

Aufgabe 4

Erweitern Sie die Methode `main`:

- Implementieren Sie Kreise wie in Abbildung 2a–2c gezeigt, die sich durch das Bild bewegen. Der rote Kreis startet links unten und bewegt sich nach rechts oben. Der blaue Kreis startet links oben und bewegt sich nach rechts unten.
 - Setzen Sie die Größe des Ausgabefensters auf 500×500 Pixel.
 - Setzen Sie den Radius der Kreise auf 10 Pixel.
 - Verwenden Sie als Wartezeit zwischen zwei Bildern und somit zwischen zwei Kreispositionen $5ms$. Dazu kann die Methode `StdDraw.pause(int t)` verwendet werden. Die Methode `StdDraw.clear()` hilft Ihnen dabei das Ausgabefenster zu leeren, damit Sie die Kreise neu zeichnen können.
 - Nachdem die Kreise am rechten Rand angekommen sind, sollen diese sich auf der selben Bahn wieder zurück bewegen, bis diese ihre ursprüngliche Startposition erreicht haben.
 - Für die Implementierung soll **eine einzige** Schleife verwendet werden.
- ❗ Für eine ruckelfreie Bewegung können Sie sogenanntes *DoubleBuffering* anwenden. Mehr Informationen dazu finden Sie unter <https://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html>.

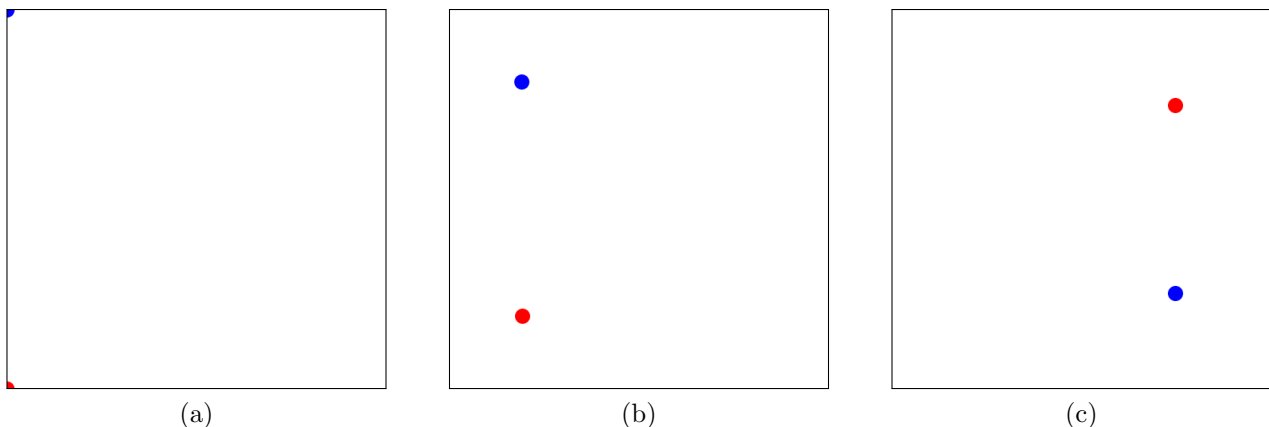


Abbildung 2: Bildausschnitte der Kreisbewegungen.

Zusatzfrage(n):

1. Ist bei diesem Beispiel die Verwendung einer `while`-Schleife von Vorteil, oder ist für diese Aufgabenstellung die `for`-Schleife zu bevorzugen?