Aufgabenblatt 1

Kompetenzstufe 1

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Freitag, 03.11.2017 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 in der Übung beantworten können.
- Verwenden Sie, falls nicht anders angegeben, für alle Ausgaben `System.out.println()`.
- Verwenden Sie für die Lösung der Aufgaben keine Aufrufe (Klassen) aus der Java-API, außer diese sind ausdrücklich erlaubt.

Aufgabe 1

Erweitern Sie die main-Methode um folgende Funktionalität für Teilaufgabe A:

- Deklarieren und initialisieren Sie je eine Variable mit den Werten `'A', 0xa, 065, 65L, 55e-1f, 6.6f, 7.77e1, 88.8` und stellen Sie sicher, dass die Variablen dieselben Typen wie diese Literale haben.
- Legen Sie zusätzlich zu den bereits deklarierten Variablen noch eine byte-Variable mit dem Wert 65, sowie eine short-Variable mit dem Wert 650 an.
- Erzeugen Sie durch Verwendung des Operators + einen String, der die Werte in den Variablen in der oben gegebenen Reihenfolge enthält, jeweils getrennt durch einen Beistrich (`", "`). Geben Sie den String mittels `System.out.println(...)` aus.
- 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.
- 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.
- 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.

Zusatzfragen zu Teilaufgabe A:

1. Wodurch erklären Sie sich die Unterschiede zwischen den beiden Summen? Welche Variable ist hauptsächlich für diesen Unterschied verantwortlich?
2. Warum ist der aus den Variablen erzeugte String nicht gleich zu den vorgegebenen Literalen `'A', 0xa, 065, 65L, 55e-1f, 6.6f, 7.77e1, 88.8`?
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 `str` vom Typ `String` und initialisieren Sie diese mit `"Hallo"`. Führen Sie danach `System.out.println(str + i + c)` sowie `System.out.println(i + c + str)` aus, wobei `i` für eine Variable vom Typ `int`, und `c` für eine Variable vom Typ `char` steht. Warum sind die Ausgaben nach und vor dem String `"Hallo"` unterschiedlich?

Erweitern Sie die main-Methode um folgende Funktionalität für Teilaufgabe B:

- Verwenden Sie für die folgenden Aufgabenstellungen immer `System.out.println(...)` um das Ergebnis auf der Konsole auszugeben. Zeigen Sie alle Funktionalitäten mit der vorgegebenen Variable `workString`.

- Verwenden Sie eine Methode der String-Klasse, die Ihnen das Zeichen an der Stelle mit dem Index 10 zurückliefert.
- Verwenden Sie eine Methode der String-Klasse, die Ihnen die Länge (Anzahl der Zeichen) des Strings zurückgibt.
- Verwenden Sie eine Methode der String-Klasse, die prüft, ob `workString` leer ist.
- Verwenden Sie eine Methode der String-Klasse, die prüft, ob `workString` mit `"In"` anfängt.
- Verwenden Sie eine Methode der String-Klasse die Ihnen die Möglichkeit gibt, Substrings zu extrahieren. Geben Sie a) den Substring ab Index 17 und b) den Substring von Index 3 bis Index 12 aus.
- Verwenden Sie eine Methode der String-Klasse, die Ihnen ermöglicht, Strings zu kombinieren. Verketten Sie den String in der Variable `workString` und den zuvor extrahierten String von Index 3 bis 12.
- Verwenden Sie eine Methode der String-Klasse, die Ihnen es ermöglicht, alle Buchstaben in einem String in Kleinbuchstaben umzuwandeln. Verwenden Sie zur Demonstration den extrahierten String von Index 3 bis 12.
- Verwenden Sie eine Methode der String-Klasse, die Ihnen es ermöglicht, alle Buchstaben in einem String in Großbuchstaben umzuwandeln. Verwenden Sie zur Demonstration den extrahierten String ab Index 17.
- Der Link `https://docs.oracle.com/javase/9/docs/api/java/lang/String.html` kann Ihnen dabei helfen, diese Aufgabe zu lösen.

Zusatzfrage zu Teilaufgabe B:

1. Wie kann man einen Substring extrahieren, wenn keine vorgefertigte Methode zur Verfügung steht? Wie würden Sie in diesem Fall vorgehen?

Aufgabe 2

Erweitern Sie für Teilaufgabe A die main-Methode um folgende Funktionalität:

- Implementieren Sie einen Bewertungsschlüssel, der eine Punktezahl (`int points`) einer Note zuordnet.
- Der Bewertungsschlüssel ist wie in Tabelle 1 aufgebaut. In der dritten Spalte finden Sie den String, der bei einer bestimmten Anzahl von Punkten (Note) ausgegeben werden soll.
- Implementieren Sie diesen Bewertungsschlüssel mittels if-Anweisungen auf zwei verschiedene Arten:
  1. Verschachtelung einfacher if-Anweisungen
  2. Kaskadieren von if-Anweisungen (Mehrfachverzweigung)

Note	Punkteintervall	Ausgabestring
1	200 – 180	`"Das ist ein Sehr gut!"`
2	179 – 160	`"Das ist ein Gut!"`
3	159 – 130	`"Das ist ein Befriedigend!"`
4	129 – 110	`"Das ist ein Genügend!"`
5	109 – 0	`"Das ist ein Nicht genügend!"`

Tabelle 1: Bewertungsschlüssel mit den dazugehörigen Ausgabestrings.

Zusatzfragen zu Teilaufgabe A:

1. Darf eine if-Anweisung ohne else-Zweig verwendet werden?
2. Gibt es ein Limit bei der Verschachtelung von if-Anweisungen?

Erweitern Sie für Teilaufgabe B die main-Methode um folgende Funktionalität:

- Implementieren Sie einen Kalenderrechner, der bei einer Angabe von Monat `int month` und Jahr `int year` die korrekte Anzahl an Tagen für diesen Monat ausgibt.
- Hinweis: Für den Monat Februar müssen Sie die Schaltjahre berücksichtigen. Beim *Gregorianischen Kalender* kommen die folgenden drei Regeln zum Einsatz:
  1. Ein Jahr ist **ein** Schaltjahr, wenn es sich restlos durch 4 teilen lässt.
  2. Ein Jahr ist **kein** Schaltjahr, wenn es sich restlos durch 100 teilen lässt.
  3. Ein Jahr ist **ein** Schaltjahr, wenn es sich restlos durch 400 teilen lässt.

  Beispiele: Februar 2000 hatte 29 Tage aufgrund eines Schaltjahres, August 1998 hatte 31 Tage, Februar 1900 hatte 28 Tage (kein Schaltjahr).

- Verwenden Sie für die Implementierung eine switch-Anweisung. Bei einer case-Marke dürfen if-Anweisungen verwendet werden.
- Bei einem falschen Datum soll der String `"Ungültiges Datum!"` ausgegeben werden, ansonsten die Anzahl der Tage des angegebenen Monats.

Zusatzfragen 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 main-Methode um folgende Funktionalität:

- Implementieren Sie ein Programm, welches das in Abbildung 1 gezeigte Bild mit allen geometrischen Formen und Farben als Ausgabe hat.
- Im Bild sind alle Maßangaben vorhanden, die notwendig sind, um dieses Bild zu erzeugen.
- Das gesamte Bild hat eine Größe von 500×500 Pixel und der Punkt $P(x = 0, y = 0)$ befindet sich in der unteren linken Ecke.
- Mit `StdDraw.setPenRadius(0.01)` wird die Linienbreite für das gesamte Bild auf 0.01 gesetzt. Sie können diese Linienbreite anpassen, falls das für die Generierung des Bildes notwendig sein sollte.
- `StdDraw.setPenColor(StdDraw.BLUE)` setzt die aktuelle Zeichenfarbe auf blau. Andere Farben können entsprechend gesetzt werden.
- Verwenden Sie für die diagonalen Linien eine geeignete Schleife.
- Der Link `http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html` kann Ihnen dabei helfen, diese Aufgabe zu lösen.

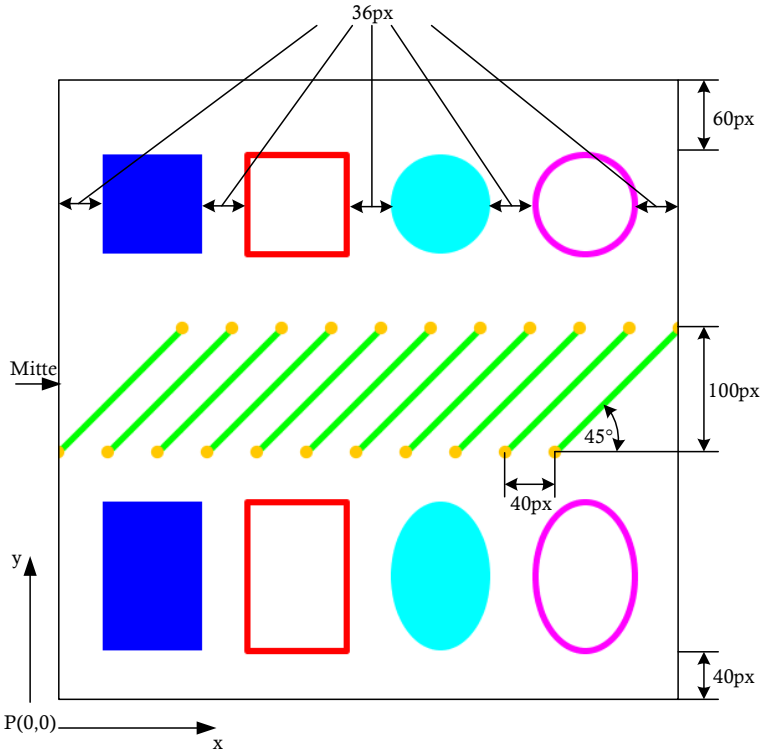


Abbildung 1: Ergebnisbild mit den entsprechenden Maßangaben

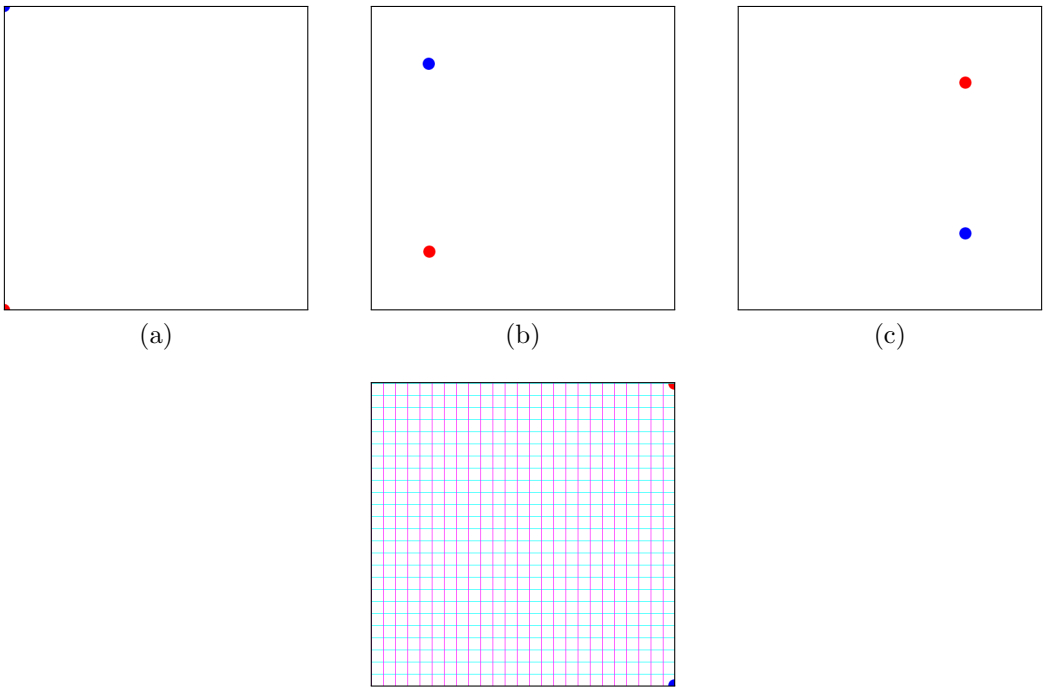
Zusatzfrage:

1. Sind verschiedene Arten von Schleifen gegeneinander austauschbar?

Aufgabe 4

Erweitern Sie die main-Methode um folgende Funktionalität:

- Implementieren Sie fliegende Kreise wie in Abbildung 2a – 2c gezeigt. Der rote Kreis startet links unten und bewegt sich nach rechts oben. Der blaue Kreis startet links oben und fliegt nach rechts unten.
- Das gesamte Bild hat eine Größe von 500×500 Pixel und der Punkt $(0, 0)$ befindet sich in der unteren linken Ecke.
- Der Radius der Kreise wird auf den Wert 10 gesetzt.
- Bitte verwenden Sie als Wartezeit zwischen zwei Bildern und somit Kreispositionen den Wert von $10ms$. Dazu kann die Methode `StdDraw.pause(int t)` nützlich sein.
- Hinweis: Für ein ruckelfreies Fliegen können Sie sogenanntes *DoubleBuffering* anwenden. Mehr Informationen dazu finden Sie unter `http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html`.
- Nachdem die Kreise ihre Endposition erreicht haben, soll ein Gitter (Abbildung 2d) gezeichnet werden. Der Linienabstand beträgt in beiden Dimensionen 20 Pixel.



(a) (b) (c)

(d)

Abbildung 2: Bildausschnitte des Ergebnisses. Bilder 2a – 2c zeigen die fliegenden Kreise und Bild 2d das Gitter mit den horizontalen und vertikalen Linien.

Zusatzfrage:

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