

Aufgabenblatt 1

Kompetenzstufe 1

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Freitag, 01.11.2019 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.
- Erlaubt sind 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 `314.15e-2f`, `3.1f`, `0.000314159e4`, `3.14159`, `0x11`, `011`, `11L`, `'3'` 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 127 sowie eine short-Variable mit dem Wert 128 an.
- c) Erzeugen Sie durch Verwendung des Plus-Operators (+) einen String, der die Werte in den zuvor beschriebenen Variablen (unter Punkt a) und b) beschrieben) enthält, jeweils getrennt durch einen Beistrich (", "). Geben Sie den String mittels `System.out.println(...)` aus.
- d) Berechnen Sie das Produkt aller Werte in diesen Variablen. Geben Sie das Produkt mittels `System.out.println(...)` aus. Erwartetes Ergebnis: `1.3411140862906337E11`
- e) Berechnen Sie das ganzzahlige Produkt aller Werte in diesen Variablen, wobei die Nachkommastellen vor der Multiplikation abgeschnitten werden sollen. Geben Sie das Produkt mittels `System.out.println(...)` aus. Erwartetes Ergebnis: `113019401088`
- f) 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. Erwartetes Ergebnis: `355`
- g) 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. Erwartetes Ergebnis: `99`
- h) Erzeugen Sie zusätzlich eine Variable `s1` vom Typ `String` und initialisieren Sie diese mit `"Zeichenkette"`. Führen Sie danach `System.out.println(s1 + i1 + c1)` sowie `System.out.println(i1 + c1 + s1)` aus, wobei `i1` für eine Variable vom Typ `int`, und `c1` für eine Variable vom Typ `char` steht.

Zusatzfrage(n) zu Teilaufgabe A:

1. Warum ist der aus den Variablen erzeugte String in Punkt c) nicht gleich zu den vorgegebenen Literalen `314.15e-2f`, `3.1f`, `0.000314159e4`, `3.14159`, `0x11`, `011`, `11L`, `'3'`, `127`, `128`?
2. Welche Variablen verursachen die Unterschiede zwischen den beiden Summen der Punkte f) und g)?
3. Wann wird ein Wert automatisch in einen Wert eines anderen Typs umgewandelt und wann muss eine Umwandlung explizit durchgeführt werden (*Cast*)?
4. Warum sind die Ausgaben unter Punkt h) vor und nach dem String `"Zeichenkette"` unterschiedlich?

Erweitern Sie für Teilaufgabe B die Methode main:

- ❗ Für die folgenden Aufgabenstellungen sind Methoden der Klasse `String` zu verwenden. Jedes Ergebnis soll 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 16 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 "Es" anfängt.
 - e) Verwenden Sie eine Methode, die Ihnen die Möglichkeit gibt, Substrings zu extrahieren. Extrahieren Sie a) den Substring von Index 0 bis Index 25 und b) den Substring ab Index 52 und speichern Sie diese Substrings in Variablen ab.
 - f) Verwenden Sie eine Methode, die es Ihnen ermöglicht, Strings zu kombinieren. Verketteten Sie die beiden gespeicherten Strings aus Punkt e), ohne den Plus-Operator (+) zu verwenden.
 - g) Verwenden Sie eine Methode, die jedes 'i' innerhalb des Strings `sentence` durch ein 'I' ersetzt.
 - h) Verwenden Sie eine Methode, die die Zeichenkette "Stringmanipulation" durch die Zeichenkette "Bearbeitung" ersetzt.
 - i) Verwenden Sie eine Methode, die alle Buchstaben in der Variable `sentence` in Kleinbuchstaben umwandelt.
 - j) Verwenden Sie eine Methode, die alle Buchstaben in einem String in Großbuchstaben umwandelt. Verwenden Sie zur Demonstration den Substring von Index 33 bis 51.
- ❗ Der Link <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html> kann Ihnen dabei helfen, diese Aufgabe zu lösen.

Zusatzfrage(n) zu Teilaufgabe B:

1. Wie kann ein Substring extrahiert werden, 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 die Anzahl der Stellen einer gegebenen Zahl `int number` als Text auf der Konsole ausgibt. Die Texte, die in Abhängigkeit der Stellen ausgegeben werden sollen, finden Sie in Tabelle 1 aufgelistet.

Anzahl der Stellen	Ausgabestring
1	"einstellig"
2	"zweistellig"
3	"dreistellig"
4	"vierstellig"
Stellen ≥ 5	"fünf- oder mehrstellig"

Tabelle 1: Ausgabestrings in Abhängigkeit der Anzahl der Stellen einer Zahl.

- ⚠ Sie können den Wert für `number` optional mittels Scanner einlesen.
- ⚠ Zum Auslesen der Stellen einer Integer-Zahl dürfen Sie `String.valueOf(...).length()` verwenden. Sie können die Anzahl der Stellen auch mittels Schleife ermitteln.
- Realisieren Sie diese Ausgabenzuordnung mittels `if`-Anweisungen auf zwei verschiedene Arten:

1. Verschachtelung einfacher `if`-Anweisungen (ohne `else-if` und `else`)

```
if (...) {  
    if (...) {  
        ...  
    }  
}
```

2. Mehrfachverzweigung (mit `else-if`)

```
if (...) {  
    ...  
} else if (...) {  
    ...  
} else if (...) {  
    ...  
} ...
```

Zusatzfrage(n) zu Teilaufgabe A:

1. Darf eine `if`-Anweisung ohne `else`-Zweig verwendet werden?
2. Wie viele `else`-Zweige können Sie zu einem `if`-Zweig schreiben?
3. Darf in einem `if`-Zweig ein `break` verwendet werden?

Erweitern Sie für Teilaufgabe B die Methode `main`:

- Implementieren Sie eine UV-Index-Kategorisierung, die bei Angabe des UV-Indexes `int uvIndex` die korrekte Warnstufe ermittelt und einen Text ausgibt. In Tabelle 2 sind alle Warnstufen und die dazugehörigen Ausgabetexte aufgelistet. Verwenden Sie für die Implementierung eine `switch`-Anweisung.

UV-Index	Ausgabetext
1-2	"Die UV-Belastung ist --> niedrig"
3-5	"Die UV-Belastung ist --> mittel"
6-7	"Die UV-Belastung ist --> hoch"
8-10	"Die UV-Belastung ist --> sehr hoch"
>10	"Die UV-Belastung ist --> extrem"

Tabelle 2: Kategorisierung des UV-Indexes.

- ⚠ Sie können den Wert für `uvIndex` optional mittels Scanner einlesen

Zusatzfrage(n) zu Teilaufgabe B:

1. Darf generell das `break` in den einzelnen case-Zweigen weglassen werden?
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 Maßangaben vorhanden, die für die Realisierung des Bildes notwendig sind, aber nicht gezeichnet werden müssen!
- Setzen Sie die Größe des Ausgabefensters auf 400×400 Pixel. Der Punkt $P_0(x = 0, y = 0)$ befindet sich in der linken unteren Ecke.
- ❗ Die Figuren entlang der Diagonalen stellen Kreissegmente dar, bis auf die Figur in der Mitte, die ein Kreis ist. Für die Kreisbögen können Sie die Methode `StdDraw.arc(...)` verwenden. Realisieren Sie die Kreisbögen / den Kreis mittels Schleife. Beginnend bei Punkt P_3 mit einem Kreisbogen von 0-270 Grad wird bei jedem Kreisbogen die Öffnung um 90 Grad gegen den Uhrzeigersinn weitergedreht. Jeder Kreisbogen wird mit einer Linie (Kreissehne) abgeschlossen (siehe Abbildung 1). Zusätzlich soll das erste Kreissegment links unten rot eingefärbt werden, sowie danach jede dritte Figur auf dem Weg nach rechts oben. Beachten Sie, dass für das Einfärben der mittlere Kreis bei der Zählung berücksichtigt wird.
- ❗ Die orangen Gitternetzlinien bestehen in der Ecke links oben und rechts unten aus 9 Linien.
- ❗ 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.

Zusatzfrage(n):

1. Sind verschiedene Arten von Schleifen gegeneinander austauschbar?

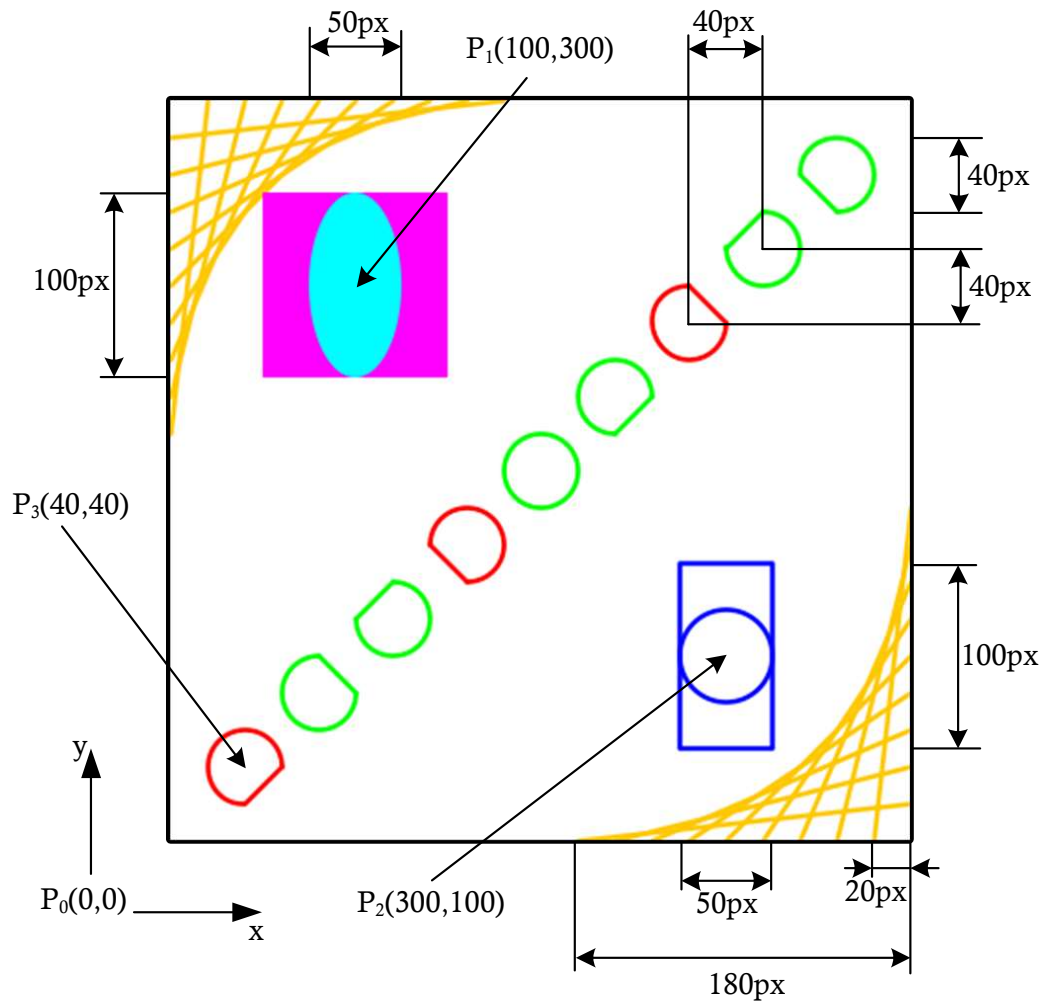


Abbildung 1: Ergebnisbild mit den entsprechenden Maßangaben.

Aufgabe 4

Erweitern Sie die Methode `main`:

- Zeichnen Sie eine rote Acht (Abbildung 2h) mittels eines gefüllten Kreises, den Sie bei jedem Zeichnen um ein Pixel weiter bewegen. Starten Sie am linken Rand in der Mitte (Abbildung 2a) und bewegen Sie den Kreis entlang des Randes gegen den Uhrzeigersinn (Abbildung 2b-2f), bis Sie wieder beim Ausgangspunkt angelangt sind. Danach bewegen Sie den Kreis noch zusätzlich zum rechten Rand, um die Acht zu komplettieren (Abbildung 2g-2h).
- Setzen Sie die Größe des Ausgabefensters auf 250×500 Pixel.
- Setzen Sie den Radius des Kreises auf 10 Pixel.
- Verwenden Sie als Wartezeit zwischen zwei Bildern und somit zwischen zwei Kreispositionen $5ms$. Dazu kann die Methode `StdDraw.pause(...)` verwendet werden. Der Kreis wird zwischen zwei Kreispositionen um ein Pixel weiterbewegt.

Zusatzfrage(n):

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

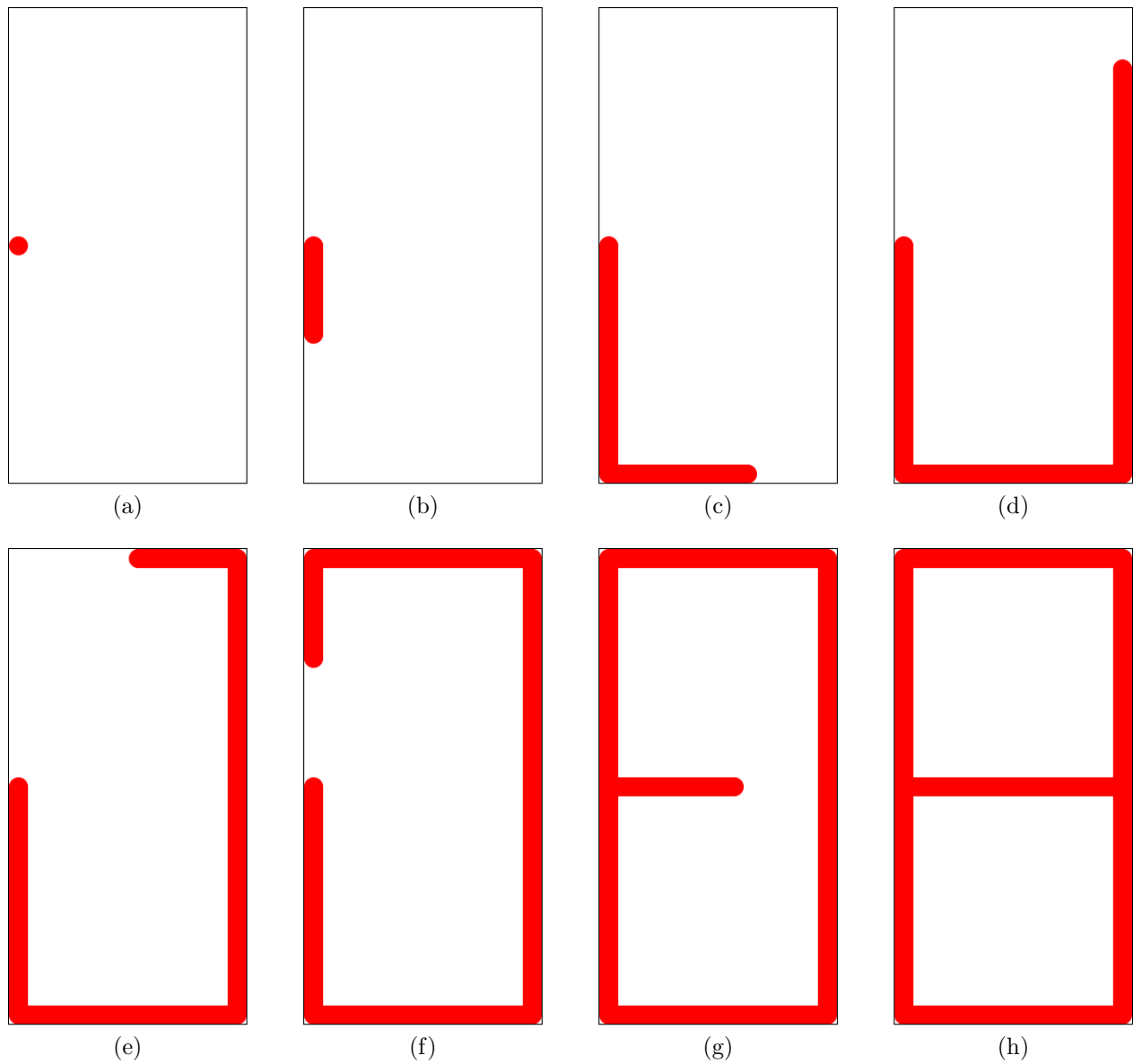


Abbildung 2: Bildausschnitte der Kreisbewegung (beginnend bei 2a) gegen den Uhrzeigersinn.