Aufgabenblatt 1

Kompetenzstufe 1

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Dienstag, 31.03.2020 13:00 Uhr** in TUWEL hoch.
- Zusätzlich müssen Sie in TUWEL ankreuzen, welche Aufgaben Sie gelöst haben.
- Ihre Programme müssen kompilierbar und ausführbar sein.
- Ändern Sie bitte **nicht** die **Dateinamen** und die **vorhandene Ordnerstruktur**.
- Bei den Aufgaben finden Sie Zusatzfragen. Diese Zusatzfragen beziehen sich thematisch auf das erstellte Programm. Sie müssen bei diesem Aufgabenblatt die Antworten als Java-Kommentare in die Dateien schreiben (unter den Lösungen).
- 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 `String` oder Klassen, die in den Hinweisen zu den einzelnen Aufgaben aufscheinen.

In diesem Aufgabenblatt werden folgende Themen behandelt:

- Verzweigungen (if-Anweisung, switch-Anweisung)
- Einfache Schleifen (for-Schleife, while-Schleife)
- Umgang mit den Klassen `String` und `StdDraw`

Aufgabe 1

Erweitern Sie die Methode `main`:

a) Implementieren Sie eine Verzweigungsstruktur, die bei einem gegebenen String überprüft, ob ein Punkt innerhalb des Strings vorkommt. Wenn ein Punkt enthalten ist, dann wird zuerst alles vom Punkt bis zum Ende des Strings ausgegeben und danach alles vom Beginn des Strings bis zum Punkt. Der Punkt ist in der Ausgabe nicht vorhanden. Sollte es mehrere Punkte geben, dann wird nur der erste Punkt (von links gelesen) betrachtet und die anderen Punkte bleiben in der Ausgabe erhalten. Wenn kein Punkt enthalten ist, dann wird in der Mitte des Strings ein Punkt eingefügt. Bei Strings mit ungerader Anzahl an Zeichen wird dieser beim kleineren Index (ganzzahliges Ergebnis bei der Division durch 2) eingefügt. Für die Implementierung dürfen keine Schleifen verwendet werden und die Änderung des Strings wird direkt auf der Konsole ausgegeben, ohne einen neuen String zu erzeugen. Verwenden Sie für die Aufgabe Methoden aus der Klasse `String`.

Beispiele:

`"Der erste Satz. Der zweite Teil."` liefert `" Der zweite Teil.Der erste Satz"`
`"Ein Punkt am Ende."` liefert `"Ein Punkt am Ende"`
`".Ein Punkt am Anfang!"` liefert `"Ein Punkt am Anfang!"`
`"Stringteilung"` liefert `"String.teilung"`
`"Stringteilung!"` liefert `"Stringt.eilung!"`

b) 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 `"Nur ein Zeichen gefunden"` ausgegeben und wenn `character` mehr als einmal vorkommt, dann wird der String `"Zwei oder mehr Zeichen gefunden"` ausgegeben. Für die Implementierung dürfen keine Schleifen verwendet werden.

Beispiele:

`"Erdbeermarmelade"` und `character = 'g'` liefert `"Kein Zeichen gefunden"`
`"Erdbeermarmelade"` und `character = 'l'` liefert `"Nur ein Zeichen gefunden"`
`"Erdbeermarmelade"` und `character = 'e'` liefert `"Zwei oder mehr Zeichen gefunden"`

c) 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. Für den Monat Februar müssen Sie die Schaltjahre[1] berücksichtigen. Beim *Gregorianischen Kalender* kommen die folgenden drei Kriterien zur Anwendung, nach denen geprüft wird, ob es sich um ein Schaltjahr handelt:

1. Schaltjahre müssen durch 4 teilbar sein.
2. Ist das Jahr auch durch 100 teilbar, ist es kein Schaltjahr, es sei denn...
3. ...das Jahr ist ebenfalls durch 400 teilbar – dann ist es ein Schaltjahr.

[1] Quelle: `https://www.timeanddate.de`

Beispiele: Die Jahre 2000 und 2400 sind Schaltjahre (infolge Regel 3); Jahre 1800, 1900, 2100, 2200, 2300 und 2500 sind hingegen keine Schaltjahre (infolge Regel 2, aber Regel 3 greift nicht).

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 im angegebenen Jahr.

Zusatzfrage(n):

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?
4. Darf generell das `break` in den einzelnen case-Zweigen einer switch-Anweisung weggelassen werden?
5. Muss bei der Verwendung der switch-Anweisung ein default-Zweig implementiert werden?
6. Sind if- und switch-Anweisungen gegeneinander austauschbar?

Aufgabe 2

Erweitern Sie die Methode `main`:

a) Schreiben Sie eine for-Schleife, die alle Zahlen im Intervall [0, 100] aufsummiert und das Ergebnis auf der Konsole ausgibt. Erwartetes Ergebnis: 5050

b) Schreiben Sie eine for-Schleife, die alle durch 3 und 7 teilbaren Zahlen im Intervall [53, 146] auf der Konsole untereinander ausgibt. Erwartetes Ergebnis: 63 84 105 126

c) Schreiben Sie eine for-Schleife, die alle geraden und durch 13 teilbaren Zahlen im Intervall]130, 13[auf der Konsole durch Leerzeichen getrennt nebeneinander ausgibt. Erwartetes Ergebnis: 104 78 52 26

d) Schreiben Sie eine for-Schleife, die alle ungeraden Zahlen im Intervall]-15, 15[multipliziert und das Ergebnis auf der Konsole ausgibt. Erwartetes Ergebnis: -18261468225

e) Schreiben Sie eine for-Schleife, die alle Vorkommen des Buchstaben `'a'` innerhalb des gegebenen Strings `"Das ist eine Metallwaage"` zählt und die Summe auf der Konsole ausgibt. Erwartetes Ergebnis: 4

Zusatzfrage(n):

1. Können Endlosschleifen bei jeder Form von Schleife entstehen?
2. Kann bei einer for-Schleife das Schleifeninkrement auch innerhalb des Schleifenrumpfes stehen?
3. Macht es bei einer for-Schleife einen Unterschied, ob der Schleifenindex mit `i++`, oder `++i` inkrementiert wird?

Aufgabe 3

Erweitern Sie die Methode `main`:

- Deklarieren Sie eine String-Variable `text` und initialisieren Sie diese mit `"Um ins Ziel zu treffen, sollte richtig gezielt werden!"`. Testen Sie zusätzlich auch mit dem String `"Ich will ins Ziel treffen."`, um Ihre Implementierung zu prüfen.

a) Schreiben Sie eine while-Schleife, die den String `text` von vorne beginnend durchläuft und alle Zeichen untereinander ausgibt, bis der Buchstabe `'z'` gefunden wird. Der Buchstabe `'z'` wird nicht mehr ausgegeben. Wenn kein Buchstabe `'z'` vorhanden ist, wird der komplette String ausgegeben. Erwartetes Ergebnis:
`"Um ins Ziel zu treffen, sollte richtig gezielt werden!"` liefert `"Um ins Ziel"`
`"Ich will ins Ziel treffen."` liefert `"Ich will ins Ziel treffen."`

b) Schreiben Sie eine while-Schleife, die den String `text` von vorne beginnend durchläuft und das erste Vorkommen des Buchstabens `'u'` sucht. Wird der Buchstabe `'u'` gefunden, dann wird dessen Index auf der Konsole ausgegeben, ansonsten -1. Erwartetes Ergebnis:
`"Um ins Ziel zu treffen, sollte richtig gezielt werden!"` liefert 13
`"Ich will ins Ziel treffen."` liefert -1

c) Schreiben Sie eine while-Schleife, die alle ungeraden und durch 7 teilbaren Zahlen im Intervall [20, 130] aufsummiert und das Ergebnis auf der Konsole ausgibt. Erwartetes Ergebnis: 560

d) Schreiben Sie eine while-Schleife, die vom String `text` von hinten beginnend jedes **zweite** Zeichen auf der Konsole durch Leerzeichen getrennt nebeneinander ausgibt. Das erste ausgegebene Zeichen ist in diesem Fall das Zeichen `'n'`, dann `'d'`, usw. Erwartetes Ergebnis:
`"Um ... werden!"` liefert `"n d e  l i e  i h i  t l s , e f r  z l i  n  U"`
`"Ich will ins Ziel treffen."` liefert `"n f e t l i  n  l w h I"`

e) Schreiben Sie eine while-Schleife, die im String `text` alle Vokale (Selbstlaute) zählt (Groß- und Kleinschreibung) und das Ergebnis auf der Konsole ausgibt. Erwartetes Ergebnis:
`"Um ins Ziel zu treffen, sollte richtig gezielt werden!"` liefert 16
`"Ich will ins Ziel treffen."` liefert 7

Zusatzfrage(n):

1. Sind verschiedene Arten von Schleifen gegeneinander austauschbar?

2. Darf eine while-Schleife innerhalb einer for-Schleife verwendet werden und umgekehrt?

Aufgabe 4

Erweitern Sie die Methode `main` um folgende Funktionalität:

- 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.
- Das gesamte Bild hat eine Größe von 400×400 Pixel und der Punkt $P(x = 0, y = 0)$ befindet sich in der unteren linken Ecke.
- Mit `StdDraw.setPenRadius(0.005)` wird die Linienbreite für das gesamte Bild gesetzt.
- `StdDraw.setPenColor(StdDraw.BLUE)` setzt die aktuelle Zeichenfarbe auf blau. Andere Farben können entsprechend gesetzt werden.
- Verwenden Sie für die orangen und grünen Linien eine geeignete Schleife. Zusätzlich soll jede dritte Linie (beginnend links oben nach rechts laufend) grün eingefärbt werden.
- Die Dokumentation unter dem Link `http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html` kann Ihnen dabei helfen, diese Aufgabe zu lösen.

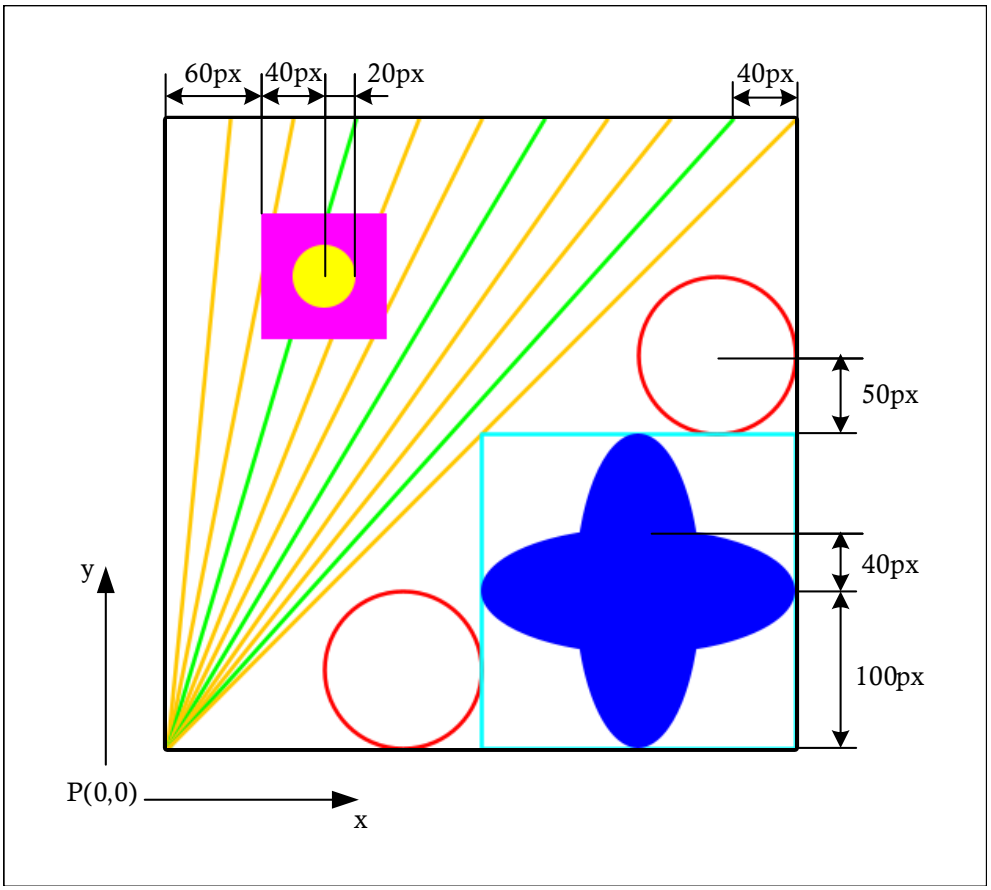


Abbildung 1: Ergebnisbild mit den entsprechenden Maßangaben

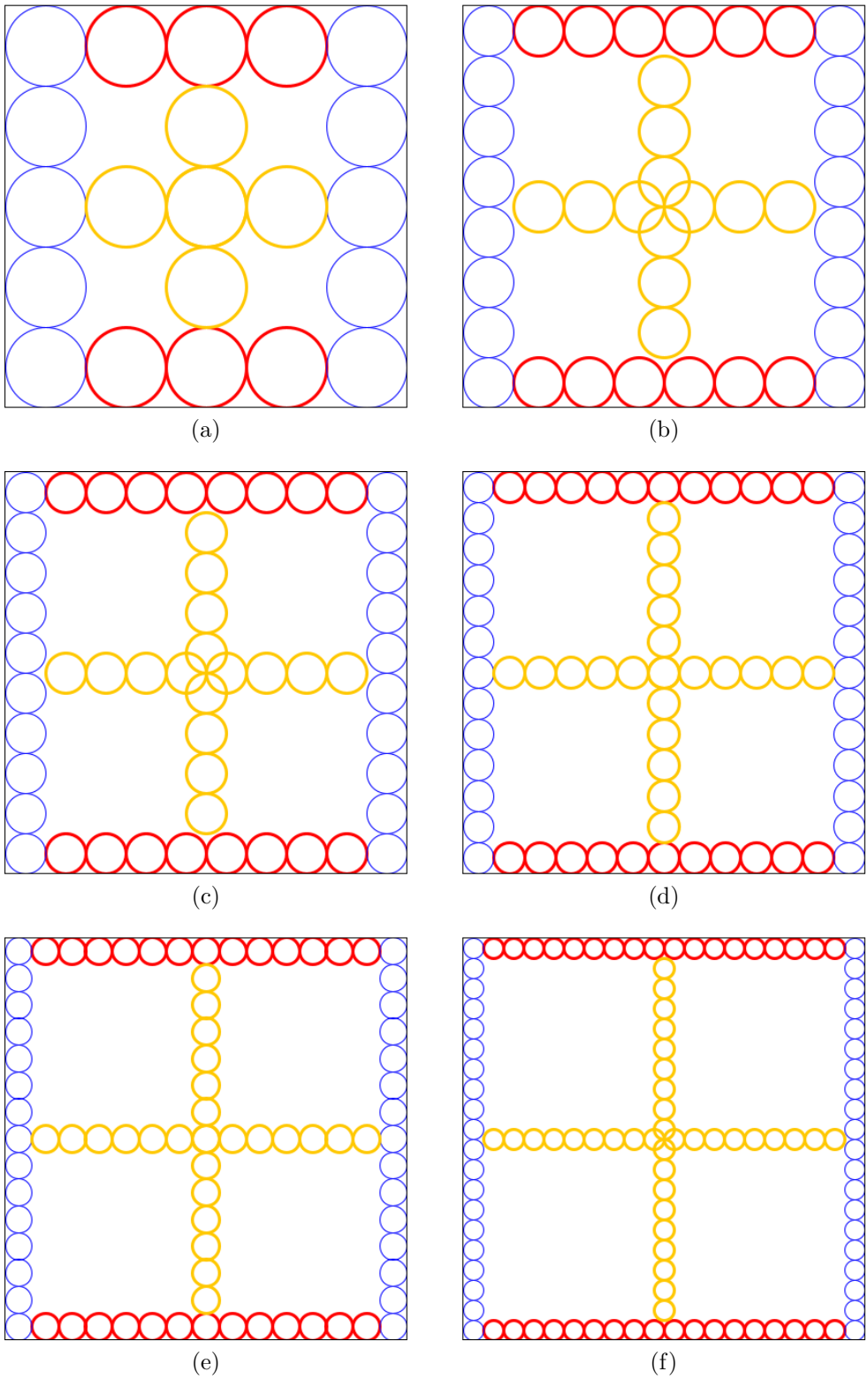
Aufgabe 5

Erweitern Sie die Methode `main`:

- Setzen Sie die Größe des Ausgabefensters auf 300×300 Pixel.
- Schreiben Sie ein Programm, das unter Vorgabe einer Variablen `n` die Abbildungen 2a-2f erzeugen kann. Die Variable `n` gibt an, wie viele Kreise horizontal und vertikal zu zeichnen sind. Die Kreise am linken und rechten Rand sind blau einzufärben und sollen eine Liniendicke von 0.006 aufweisen. Die Kreise am oberen und unteren Rand sind rot einzufärben, sollen dabei aber die bereits vorhandenen blauen Kreise nicht überzeichnen. Die Kreise dürfen sich aber berühren. Die Liniendicke der roten Kreise beträgt 0.006. Zusätzlich sollen noch orange Kreise als Kreuz angeordnet in der Mitte der Abbildung eingezeichnet werden, ohne dabei die am Rand befindlichen Kreise zu überzeichnen (Kreise dürfen sich berühren). Die Liniendicke der orangen Kreise beträgt 0.002. Achten Sie hier darauf, dass sich je nach geradem oder ungeradem `n` unterschiedliche Darstellungen für die orangen Kreise ergeben. In den Abbildungen 2a, 2d und 2e ist `n` auf 5, 13 und 15 gesetzt und daher gibt es einen Kreis zentral in der Mitte. Bei den Abbildungen 2b, 2c und 2f ist `n` auf 8, 10 und 20 gesetzt und daher ergeben sich Überschneidungen in der Mitte bei den orangen Kreisen. Hinweis: Verwenden Sie Gleitkommaarithmetik, um für jedes `n` das StdDraw-Fenster füllend auszunutzen.

Zusatzfrage(n):

1. Wie kann der Koordinatenursprung des StdDraw-Fensters verschoben werden?



(a) (b) (c) (d) (e) (f)

Abbildung 2: Verschiedene Kreismuster mit unterschiedlicher Anzahl an Kreisen `n`. a) `n = 5`, b) `n = 8`, c) `n = 10`, d) `n = 13`, e) `n = 15` und f) `n = 20`.