Aufgabenblatt 1

Kompetenzstufe 1

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Donnerstag, 17.11.2022 20:00 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**.
- 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 `CodeDraw`, `Math` 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 `CodeDraw`

Aufgabe 1 (1 Punkt)

Erweitern Sie die Methode `main`:

a) Schreiben Sie eine for-Schleife, die alle durch 17 teilbaren Zahlen im Intervall [1] [17, 170] aufsummiert und das Ergebnis auf der Konsole ausgibt.
Erwartetes Ergebnis: 935

b) Schreiben Sie eine for-Schleife, die jede 6. Zahl im Intervall [60, 82] (beginnend mit 60) nebeneinander durch Leerzeichen getrennt ausgibt.
Erwartetes Ergebnis: 60 66 72 78

c) Schreiben Sie eine for-Schleife, die alle durch 9 und 13 teilbaren Zahlen im Intervall von]117, 585[hintereinander und getrennt durch Pluszeichen (`'+'`) ausgibt. Zusätzlich wird noch vor der ersten Zahl und nach der letzten Zahl ein Pluszeichen ausgegeben.
Erwartetes Ergebnis: +234+351+468+

d) Schreiben Sie eine for-Schleife, die alle Zeichen der ASCII [2]-Werte im Intervall]45, 65] in absteigender Reihenfolge durch Leerzeichen getrennt ausgibt.
Erwartetes Ergebnis: A @ ? > = < ; : 9 8 7 6 5 4 3 2 1 0 / .

e) Schreiben Sie eine for-Schleife, die alle Vorkommen des Buchstabens `'e'` und `'E'` im Satz `Es ist kein echtes Edelmetall!` zählt.
Erwartetes Ergebnis: 7

[1] Intervall-Schreibweise: `https://de.wikipedia.org/wiki/Intervall_(Mathematik)`
[2] ASCII-Code: `https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange`

Aufgabe 2 (2 Punkte)

Erweitern Sie die Methode `main`:

- Deklarieren Sie eine String-Variable `text` und initialisieren Sie diese mit `"Wir beginnen am Anfang mit den Grundlagen."`. Testen Sie zusätzlich mit dem String `"Kein gesuchtes Zeichen im String!"`, um Ihre Implementierung zu prüfen. Sie dürfen auch mehrere String-Variablen deklarieren und durch entsprechende Zuweisungen unterschiedliche Tests realisieren.

a) Schreiben Sie eine while-Schleife, die vom String `text` von vorne beginnend jedes zweite Zeichen überprüft und nebeneinander ausgibt, falls es sich nicht um das Zeichen `'a'` handelt. Das erste ausgegebene Zeichen für den String `"Wir fangen am Anfang mit den Grundlagen an."` ist in diesem Fall das Zeichen `'i'`, dann `' '`, usw.
Erwartetes Ergebnis:
`"Wir beginnen am Anfang mit den Grundlagen."` **liefert** `"i einn ngmtdnGude."`
`"Kein gesuchtes Zeichen im String!"` **liefert** `"engsctsZihni tig"`

b) Schreiben Sie eine while-Schleife, die im String `text` von vorne beginnend das erste Vorkommen des Buchstabens `'s'` sucht. Wird der Buchstabe `'s'` gefunden, dann wird dessen Index auf der Konsole ausgegeben. Andernfalls wird -1 ausgegeben. Für diese Implementierung dürfen die Methoden `indexOf(...)` und `lastIndexOf(...)` **nicht** verwendet werden.
Erwartetes Ergebnis:
`"Wir beginnen am Anfang mit den Grundlagen."` **liefert** -1
`"Kein gesuchtes Zeichen im String!"` **liefert** 7

c) Schreiben Sie eine while-Schleife, die alle geraden durch 17 teilbaren Zahlen im Intervall]17, 238[nebeneinander ausgibt.
Erwartetes Ergebnis: 34 68 102 136 170 204

d) Schreiben Sie eine while-Schleife, die den String `text` von hinten beginnend durchläuft und Zeichen für Zeichen nebeneinander ausgibt, bis der Buchstabe `'e'` 3-Mal gefunden wurde. Gab es weniger als 3 Vorkommen von `'e'`, dann wird der komplette String ausgegeben.
Erwartetes Ergebnis:
`"Wir beginnen am ... Grundlagen."` **liefert** `".negaldnurG ned tim gnafnA ma ne"`
`"Kein gesuchtes Zeichen im String!"` **liefert** `"!gnirtS mi nehcieZ se"`

e) Schreiben Sie eine while-Schleife, die im String `text` die Vorkommen aller Leerzeichen sowie Satzzeichen `'.'` und `'!'` zählt und das Ergebnis auf der Konsole ausgibt.
Erwartetes Ergebnis:
`"Wir beginnen am Anfang mit den Grundlagen."` **liefert** 7
`"Kein gesuchtes Zeichen im String!"` **liefert** 5

Aufgabe 3 (2 Punkte)

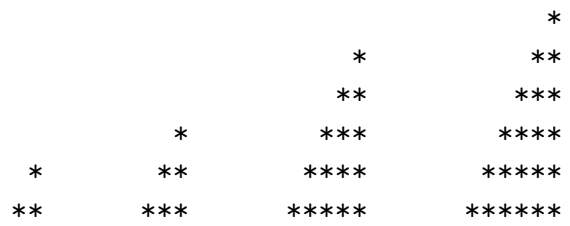
Erweitern Sie die Methode `main`:

- Implementieren Sie ein Programm, welches verschiedene Formen in unterschiedlicher Größe auf der Konsole ausgeben kann. Es gibt für die Steuerung des Programms zwei Variablen. Die Höhe einer Form in Zeilen wird mit der Variable `int height` gesteuert und sollte nur positive Werte größer 1 annehmen. Die Art der Form wird mit der Variable `int pattern` ausgewählt. Hier steht eine 0 für eine Linie, 1 für ein rechtwinkeliges Dreieck und 2 für ein Parallelogramm. Implementieren Sie eine Verzweigungsstruktur, die je nach Inhalt von `pattern` und `height` die entsprechende Form mit dem Zeichen `'*'` auf der Konsole ausgibt. Es reicht bei dieser Aufgabe aus, wenn Sie den Variablen konkrete Werte direkt zuweisen. Sie dürfen aber zusätzlich Überprüfungen einbauen und bei falschen Werten Fehlermeldungen auf der Konsole ausgeben.

  Beispiele für unterschiedliche Linien (`pattern = 0`):

Ausgaben für Linien (von links nach rechts) mit `height` 2, 3, 5 und 6.

  Beispiele für unterschiedliche Dreiecke (`pattern = 1`):



Ausgaben für Dreiecke (von links nach rechts) mit `height` 2, 3, 5 und 6.

  Beispiele für unterschiedliche Parallelogramme (`pattern = 2`):

Ausgaben für Parallelogramme (von links nach rechts) mit `height` 2, 3, 5 und 6.

Aufgabe 4 (1 Punkt)

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.
- Das gesamte Bild hat eine Größe von 400×400 Pixel und der Punkt $P(x = 0, y = 0)$ befindet sich in der oberen linken Ecke. Dazu erstellen Sie ein Objekt der Klasse `CodeDraw` mit der Anweisung `CodeDraw myDrawObj = new CodeDraw(400, 400)`, um ein Zeichenfenster mit der Größe 400×400 Pixel zu erstellen.
- Mit `myDrawObj.setLineWidth(2)` wird die Linienbreite auf 2 Pixel gesetzt.
- Mit `myDrawObj.setColor(Palette.RED)` können Sie die aktuelle Zeichenfarbe auf rot setzen. Andere Farben können entsprechend gesetzt werden. Verwendete Farben in der Abbildung: `MAGENTA`, `CYAN`, `BLUE`, `YELLOW`, `RED` und `GREEN`.
- Verwenden Sie für die grünen Linien **eine** einzige Schleife.
- Um Zeichnungen im Ausgabefenster sichtbar zu machen, müssen Sie die Methode `myDrawObj.show()` aufrufen.
- Die Dokumentation unter dem Link `https://krassnig.github.io/CodeDrawJavaDoc/v3.0.x/codedraw/package-summary.html` kann Ihnen dabei helfen, diese Aufgabe zu lösen. Beachten Sie bitte dabei, dass sich die Methoden zum Zeichnen von Figuren in der Klasse Image befinden.

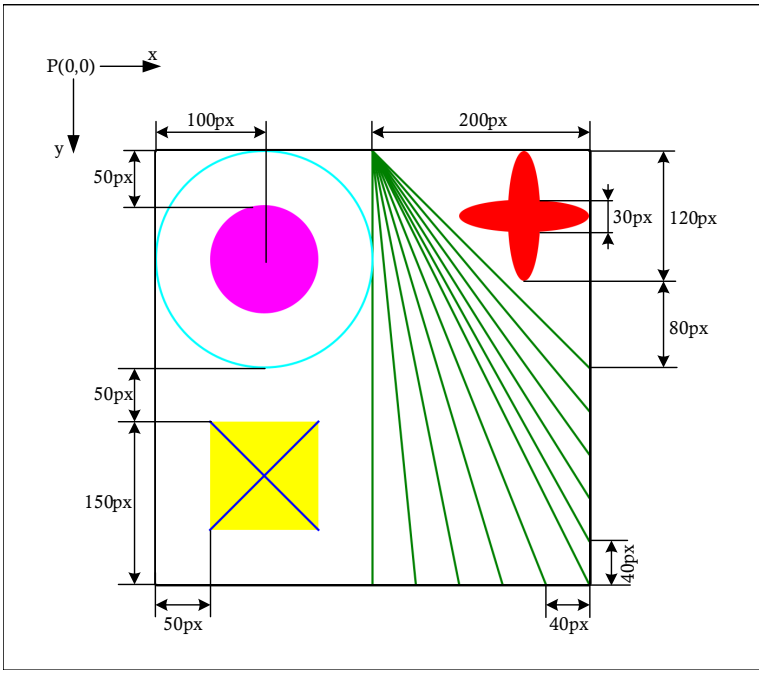


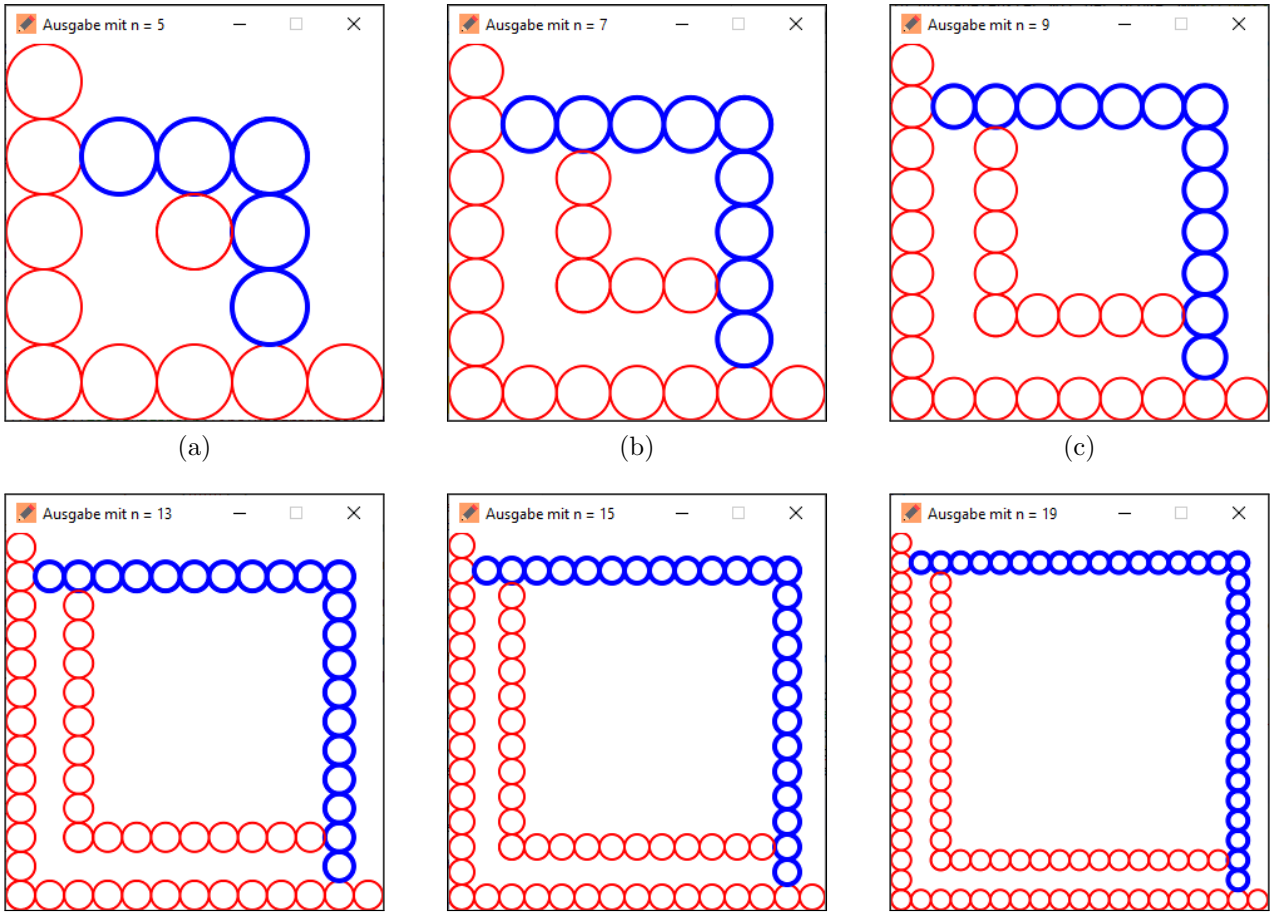
Abbildung 1: Ergebnisbild mit den entsprechenden Maßangaben

Aufgabe 5 (2 Punkte)

Erweitern Sie die Methode `main`:

- Erstellen Sie ein Ausgabefenster mit der Größe 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 maximal horizontal und vertikal zu zeichnen sind und darf dabei nur ungerade Werte größer gleich 5 und kleiner gleich 19 annehmen. Überprüfen Sie in Ihrer Implementierung, ob die Vorgaben für `n` eingehalten werden, ansonsten geben Sie eine Fehlermeldung aus. Die roten Kreise haben eine Liniendicke von 2 Pixel und die blauen Kreise eine Liniendicke von 4 Pixel.

(!) Hinweis: Verwenden Sie Gleitkommaarithmetik, um für jedes `n` das CodeDraw-Fenster füllend auszunutzen.



(a) (b) (c)

(d) (e) (f)

Abbildung 2: Verschiedene Kreismuster mit unterschiedlicher Anzahl an Kreisen `n`. a) `n = 5`, b) `n = 7`, c) `n = 9`, d) `n = 13`, e) `n = 15` und f) `n = 19`.