



0.7. Aufgabe (1 Punkt)

- a) Geben Sie die Lösungsworte der Quizze aus der Lernnotiz an.
- b) Füllen Sie die per Link über E-Mail verschickten Evaluationsbögen aus.

25. Aufgabe (7 Punkte, einfache Schleifen ohne echte Objektorientierung, VL 12)

Zur Veranschaulichung der Ergebnisse Teilaufgaben können folgende Webseiten genutzt werden: e) <https://kleuker.iui.hs-osnabrueck.de/querschnittlich/prog1/ingabeanalyse.html>,

f) <https://kleuker.iui.hs-osnabrueck.de/querschnittlich/prog1/abzahlungZeigen.html>,

g) <https://kleuker.iui.hs-osnabrueck.de/querschnittlich/prog1/zahlenraten.html> und

h) <https://kleuker.iui.hs-osnabrueck.de/querschnittlich/prog1/zahlenratenCheat.html>.

Die Programme werden mit einem Klick auf den Reload-Button des Browsers neu gestartet.

Schreiben Sie eine neue Klasse Zahlenbetrachtung, mit der int-Werte genauer betrachtet werden können. Bei den zu entwickelnden Methoden kann es immer sinnvoll sein, sich den Ablauf zunächst mit Aktivitätsdiagrammen zu veranschaulichen und dabei noch keinen konkreten Programmcode in die Aktionen zu schreiben. Bei zwei Teilaufgaben ist ein solches Diagramm in der Aufgabenstellung gefordert, beim Rest optional. Machen Sie solchen Skizzen zumindest zunächst per Hand, durchaus mit schwerer zu lesenden Korrekturen.

- a) Schreiben Sie eine Methode istTeilerVon(int,int), die zurückgibt, ob die erste übergebene Zahl Teiler der zweiten übergebenen Zahl ist. Nutzen Sie die Modulo-Rechnung mit %. Falls die erste Zahl 0 ist, soll das Ergebnis false sein.
- b) Schreiben Sie eine Methode alleTeilerVon(int), die alle Teiler der übergebenen Zahl zurückgibt. Für Zahlen kleiner 1 und wird eine leere Liste zurückgegeben. Überprüfen Sie dazu innerhalb einer Schleife alle möglichen Teiler ab der Zahl 1. Zeichnen Sie zunächst ein Aktivitätsdiagramm für den Ablauf Ihrer Methode.
- c) Schreiben Sie eine Methode groessterGemeinsamerTeiler(int,int), der die größte Zahl findet, die Teiler beider übergebenen Zahlen ist. Nutzen Sie dabei ein einfaches Verfahren, bei dem Sie z. B. alle Zahlen von 1 bis zur ersten übergebenen Zahl ausprobieren. Sollte ein Parameter kleiner 1 sein, ist das Ergebnis 0. Zeichnen Sie zunächst ein Aktivitätsdiagramm für den Ablauf Ihrer Methode.
- d) Aufgabe c) kann auch durch den sogenannten Euklidischen Algorithmus gelöst werden. Informieren Sie sich über das Verfahren und implementieren Sie es als Methode groessterGemeinsamerTeiler2(int,int).

- e) Zu entwickeln ist eine Methode eingabeanalyse(), die bei der Nutzung immer wieder zur Eingabe einer Fließkommazahl (double) auffordert, die immer zwischen -100.0 und 100.0 liegt (muss geprüft werden, falsche Werte ignorieren) und die nach jeder Eingabe den kleinsten und den größten bisher eingegebenen Wert sowie

```
Blue: Terminal Window - AufgabeZahlenbetrachtung...
Options
Geben Sie einen Wert ein: 3.5
Minimum: 3.5 Maximum: 3.5 Schnitt: 3.5
Geben Sie einen Wert ein: 7.5
Minimum: 3.5 Maximum: 7.5 Schnitt: 5.5
Geben Sie einen Wert ein: 2
Minimum: 2.0 Maximum: 7.5 Schnitt: 4.333333333333333
Geben Sie einen Wert ein: 4.33
Minimum: 2.0 Maximum: 7.5 Schnitt: 4.3325
Geben Sie einen Wert ein: 101
Geben Sie einen Wert ein: -1
```

den Mittelwert (Durchschnitt aller eingegebenen Werte) ausgibt. Überlegen Sie, welche Informationen man in lokalen Variablen speichern muss (man muss sich nicht alle Zahlen merken). Das Programm wird mit der Eingabe der Zahl -1 terminiert. Ein Beispieldialog mit Eingaben in blau steht auf der rechten Seite.

- f) Entwickeln Sie ein Methode abzahlungZeigen(), die einen Kredit k in Euro, einen Zinssatz z in % und eine jährliche Tilgungssumme t in Euro jeweils als double-Werte einliest. Die Kreditsumme des Folgejahres mit jährlicher Tilgung berechnet sich als $k \cdot (1 + z/100) - t$. Ihr Programm soll das Jahr, die verbleibende Kreditsumme und die



bisher insgesamt gezahlten Zinsen (im Jahr $k^*(z/100)$) ausgehen. Ihr Programm endet, wenn der Kredit zurückgezahlt ist. Falls dieses nicht möglich ist, soll Ihr Programm eine Warnung ausgeben. Zwei Beispieldialoge mit Eingaben in blau stehen wie folgt aus.

```
BlueJ: Terminal Window - AufgabeZahlenbetrachtungLoesung
Options
Kreditsumme: 100
Zinssatz: 10
Tilgung: 20
1.Jahr Restkredit: 90.00000000000001, gezahlte Zinsen:10.0
2.Jahr Restkredit: 79.00000000000003, gezahlte Zinsen:19.0
3.Jahr Restkredit: 66.90000000000003, gezahlte Zinsen:26.90000000000002
4.Jahr Restkredit: 53.59000000000046, gezahlte Zinsen:33.59
5.Jahr Restkredit: 38.94900000000055, gezahlte Zinsen:38.94900000000005
6.Jahr Restkredit: 22.84390000000062, gezahlte Zinsen:42.8439000000001
7.Jahr Restkredit: 5.12829000000071, gezahlte Zinsen:45.1282900000002
8.Jahr Restkredit: -14.35888099999922, gezahlte Zinsen:45.64111900000003
Kreditsumme: 250000
Zinssatz: 4.5
Tilgung: 1000.42
Kredit so nicht abzahlbar!
```

- g) Entwickeln Sie eine Methode `zahlenRaten()` mit der Sie mit dem Computer „Zahlenraten“ spielen können. Dazu soll der Computer eine Zahl zwischen 1 und 100 (jeweils einschließlich) wählen. Die Klasse `EinUndAusgabe.java` bietet dazu die Methode `zufall(int,int)`, die einen Zufallswert zwischen den übergebenen minimalen und maximalen Werten liefert. Danach kann der Spieler seinen Tipp eingeben und der Computer antwortet mit „zu hoch“, „zu tief“ oder „korrekt“. Weiterhin zählt der Computer die Anzahl der Versuche und gibt dies vor jedem Rateversuch aus. Ein Beispieldialog mit Eingaben in blau steht auf der rechten Seite.

```
BlueJ: Terminal Window - AufgabeZahleint
Options
1. Versuch: Zahl raten: 42
zu tief
2. Versuch: Zahl raten: 69
zu tief
3. Versuch: Zahl raten: 85
zu hoch
4. Versuch: Zahl raten: 77
Bravo, getroffen
```

- h) (freiwillige Bonusaufgabe) Ändern Sie Ihr Programm aus g) so, dass der Computer zwar immer korrekt antwortet, allerdings den Spieler immer dazu bringt, die maximale Anzahl an Versuchen zu machen. Der Computer rät also keine Zahl sondern versucht sich möglichst viele Möglichkeiten offen zu halten. Dabei sind natürlich alle Hinweise mit „zu tief“ und „zu hoch“ korrekt.
Beispiel: Der Spieler gibt am Anfang 60 ein, dann stellt der Computer fest, zwischen 1 und 60 liegen mehr Zahlen als zwischen 60 und 99, deshalb gibt der Computer „zu hoch“ aus. Beim nächsten Tipp 25, gibt der Computer dann zu tief aus, da zwischen 25 und 60 mehr Zahlen liegen als zwischen 1 und 25. Ein Beispieldialog mit Eingaben in blau steht auf der rechten Seite.

```
BlueJ: Terminal Window - AufgabeZahle
Options
1. Versuch: Zahl raten: 42
zu tief
2. Versuch: Zahl raten: 80
zu hoch
3. Versuch: Zahl raten: 60
zu tief
4. Versuch: Zahl raten: 70
zu hoch
5. Versuch: Zahl raten: 65
zu hoch
6. Versuch: Zahl raten: 62
zu tief
7. Versuch: Zahl raten: 64
zu hoch
8. Versuch: Zahl raten: 63
Bravo, getroffen
```

Kleine Denkaufgabe: Überlegen Sie sich eine Zahl zwischen 1 und 100 und zeigen Sie mit Ihrem Programm zu h), dass sich der Computer genau die gleiche Zahl gedacht hat (also nach mehreren Versuchen bei Ihrer überlegten Zahl die Erfolgsmeldung kommt)



26. Aufgabe (10 Punkte, Bearbeitung von Sammlungen, VL 13)

Zur Veranschaulichung des Ergebnisses der abschließenden Teilaufgabe kann <https://kleuker.iui.hs-osnabrueck.de/querschnittlich/prog1/saeulendiagramm.html>

genutzt werden. Das Programm veranschaulicht die zu erreichende Funktionalität. Da das Programm nicht in Java geschrieben ist, sieht das Interaktionsbrett etwas anders aus. Das Programm wird mit einem Klick auf den Reload-Button des Browsers neu gestartet.

Erstellen Sie ein Programm mit dem ein Säulendiagramm erstellt werden kann. Eine einzelne Säule hat dabei einen Titel (Beschriftung) und einen anzuzeigenden Wert. Ein Säulendiagramm besteht aus einer Sammlung von Säulen. Ein Steuerungsprogramm ermöglicht für ein Säulendiagramm das Hinzufügen und Löschen von Säulen über einen Dialog. Die Darstellung der Säulen erfolgt nach folgenden Regeln:

- am linken Rand wird der kleinste und der größte Wert aller Säulen jeweils sinnvoll platziert,
- oberhalb jeder Säule steht der konkrete Wert der Säule,
- unterhalb jeder Säule steht der Titel der Säule,
- gibt es nur eine Säule, wird diese als vollständige Säule dargestellt, kleinster und größter Wert sind gleich und stehen oben und unten,
- gibt es mehrere Säulen, die alle den gleichen Wert darstellen, werden alle als vollständige Säulen dargestellt, kleinster und größter Wert sind gleich,
- werden unterschiedliche Säulen dargestellt, wird jeweils die Differenz zur Säule mit dem kleinsten Wert als positiver Wert nach oben dargestellt, die kleinste Säule ist damit ein Strich (hat die Höhe 0),
- zum Löschen wird der Titel der Säule eingegeben und danach die erste Säule mit diesem Titel gelöscht; ist der Titel nicht vorhanden, wird die Eingabe ignoriert,
- wird die einzig existierende Säule gelöscht, wird ein „leeres Blatt“ angezeigt.

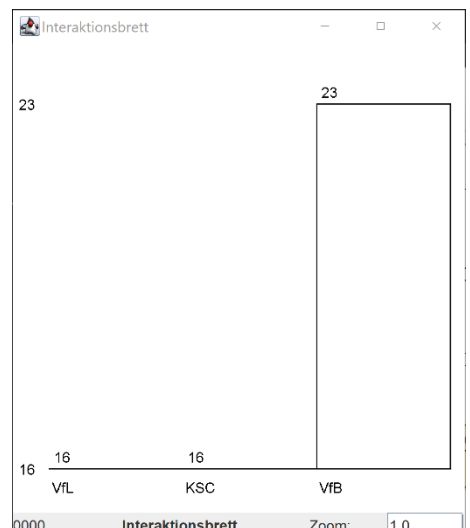
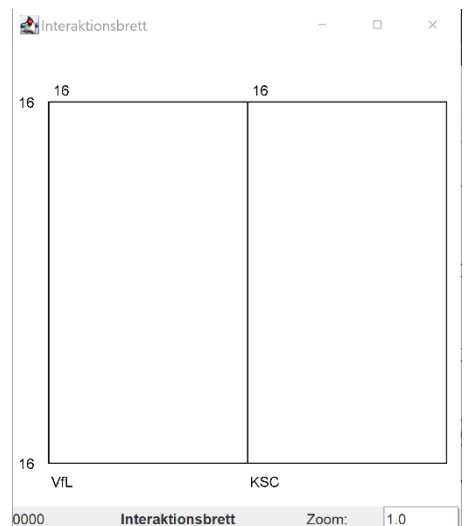
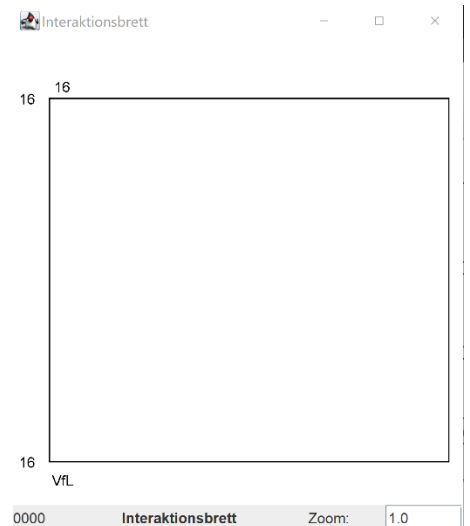
Überlegen Sie sich feste Bezugspunkte, wie die Position der unteren Linie und die y-Position für den maximalen Wert, die Sie in Ihren Berechnungen nutzen. Zwar sollen alle Werte der Säulen int-Werte sein, trotzdem ist es notwendig mit double-Werten zu rechnen, wenn z. B. der minimale Wert 42 und der maximale Wert 1967 ist, da die Säulen auf den Bildschirm passen müssen. Zur Umwandlung in double-Werte kann man int-Werte z. B. mit 1.0 multiplizieren, z. B. für `int max`

```
double einheit = 300.0 / (max * 1.0 - min);
```

Zum Umwandeln von double-Werten in int-Werte kann z. B. der Umwandlungsoperator (`int`) genutzt werden, z. B. für einen double-Wert `einheit`

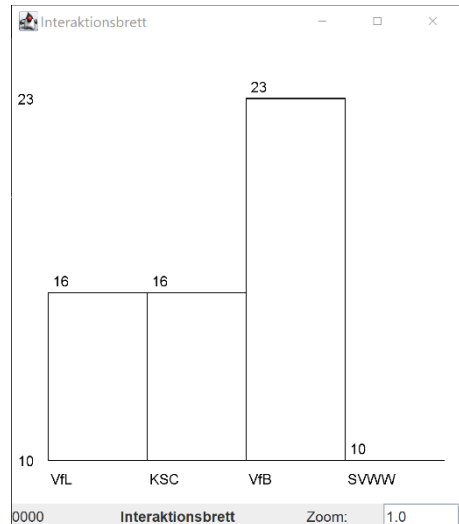
```
int abstand = (int) ((max - tmp) * einheit)
```

- a) Schreiben Sie eine Klasse `Saeule` mit den Exemplarvariablen `titel` (`String`) und `wert` (`int`), passendem Konstruktor und `get`- und `set`-Methoden.





- b) Schreiben Sie eine Klasse Säulendiagramm, die in einer Exemplarvariablen saeulen eine Sammlung von Saeulen-Objekten enthält. Weiterhin gibt es eine Methode zeichnen, die ein Interaktionsbrett übergeben bekommt und das Diagramm zeichnet. Weitere Methoden zur Strukturierung wären sinnvoll.
- c) Schreiben Sie eine Klasse Saeulendiagrammsteuerung, die u. a. ein Interaktionsbrett sowie ein Saeulendiagramm erzeugt und einen Dialog anbietet, mit der ein Saeulendiagramm bearbeitet werden kann. Im folgenden Nutzungsdialog sind Eingaben umrandet. Die zugehörigen Ausgaben sind am Rand dargestellt.



- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

1

neuer Titel: VfL

Wert: 16

- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

1

neuer Titel: KSC

Wert: 16

- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

1

neuer Titel: VfB

Wert: 23

- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

1

neuer Titel: SVWW

Wert: 10

- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

1

neuer Titel: ArB

Wert: 28

- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

2

zu loeschender Titel: SVWW

- 0 - Programm beenden
- 1 - Saeule hinzufuegen
- 2 - Saeule loeschen

0

Interaktionsbrett schliessen

