Aufgaben zur Veranstaltung
Programmierung mit Java, WS 2024/2025

H. Pflug, J. Dietel, J. Fuchs FH Aachen, Campus Jülich; IT Center, RWTH Aachen

Klausurvorbereitung Blatt 3 22./23.01.2025

1.) Die Ackermann-Funktion ist folgendermaßen definiert:

$$\begin{aligned} \texttt{ackermann}(0,m) &:= m+1 \\ \texttt{ackermann}(n,0) &:= \texttt{ackermann}(n-1,1) \quad \text{für } n>0 \\ \texttt{ackermann}(n,m) &:= \texttt{ackermann}(n-1,\texttt{ackermann}(n,m-1)) \quad \text{für } m,n>0 \end{aligned}$$

Schreiben Sie eine Java-Funktion

```
public static int ackermann(int n, int m)
```

die die Ackermann-Funktion rekursiv nach der oben gezeigten Formel berechnet. Gehen Sie davon aus, dass m und n größer als 0 sind.

2.) Eine DNA-Gensequenz kann man als String darstellen, der aus einer Kette der Buchstaben A, G, T und C besteht. Schreiben Sie eine Funktion

```
public static boolean enthaeltDoppelteTeilsequenz(String s)
```

die zurückgibt, ob in `s` eine Teilsequenz der Länge 3 doppelt vorkommt. Die beiden Teilsequenzen dürfen sich überschneiden. Gehen Sie davon aus, das auch tatsächlich eine korrekte Gensequenz übergeben wird. Beispiele:

```
agtcgagta   --> true (agt ist doppelt)
aaaacgt     --> true (aaa (ab Zeichen 0) und aaa (ab Zeichen 1))
```

3.) Schreiben Sie eine Funktion

```
public static ArrayList<Integer> getValues(String s)
```

Die Funktion untersucht den String auf eingebettete Zahlen und gibt diese Zahlen in einer ArrayList zurück.

- Eine Zahl ist dabei eine Folge der Ziffern 0 bis 9. Es werden also nur positive ganze Zahlen gesucht.
- Alle anderen Zeichen (auch Kommas, Minuszeichen usw.) trennen Zahlen voneinander ab.
- Gehen Sie davon aus, dass die Zahlen den Integer-Bereich nicht überschreiten.

Beispiele:

```
getValues("skdjn1sdjn586jy2")  --> {1,586,2}
getValues("hf0nw-1kdn3.6")     --> {0,1,3,6}
getValues("00082ms f44,12nf")  --> {82,44,12}
getValues("sdjkhbhs")          --> {}
getValues("364621")            --> {364621}
getValues("")                  --> {}
```

4.) Schreiben Sie eine Klasse `Counter`, die einen Zähler mit folgenden Eigenschaften darstellt:

- Der Zähler hat als Attribut eine Zählvariable mit dem Startwert 0.
- Es gibt zwei Methoden `plus` und `minus`, die die Variable um 1 erhöhen bzw. erniedrigen.
- Die Variable muss stets zwischen 0 und einer festen Obergrenze (die im Konstruktor festgelegt wird) liegen. Wird versucht, die Grenzen zu verlassen, wird stattdessen eine

  `CounterOutOfRangeException`

  geworfen. Diese Exception ist selbst zu schreiben. Sie soll *checked* sein.

Erzeugen Sie in der main-Methode einen Zähler, der zwischen 0 und 10 liegt. Versuchen Sie, den Zähler bis 100 hochzuzählen. Wenn die Exception auftritt, beenden Sie das Programm mit einer Fehlermeldung.

5.) a) Schreiben Sie eine Methode, der ein String als Parameter übergeben wird. Dieser String ist eine Folge von kommagetrennten Anweisungen mit Schrittzahlen und Richtungsänderungen: **N, O, S, W** (für Norden, Osten, Süden, und Westen).

Beispiel-Eingabe: **2,3,W,1,S,2**

Ist z.B. die Blickrichtung Osten, wird die Koordinate in x-Richtung um die Schrittzahl erhöht, die y-Koordinate bleibt gleich. Ist z.B. die Blickrichtung Norden, wird die Koordinate in y-Richtung um die Schrittzahl erniedrigt, die x-Koordinate bleibt gleich.

Die Methode soll die Endkoordinate berechnen und sie als `Koordinate`-Objekt zurückliefern. Diese ist für das angegebene Beispiel die Koordinate **(4;2)**. Implementieren Sie dafür eine Klasse **Koordinate**.

Die Anfangsposition ist die Koordinate **(0;0)** mit Blickrichtung nach Osten. Es gibt keine Beschränkung der Koordinaten in alle Richtungen.

Sie können davon ausgehen, dass die Eingabewerte korrekt sind und nur positive Zahlen und Großbuchstaben enthalten.

b) Schreiben Sie eine **main**-Methode, die die Methode mit einem vordefinierten String testet und die Endkoordinate auf dem Bildschirm ausgibt.

6.) In dieser Aufgabe soll eine Liste unterschiedlicher geometrischer Objekte erzeugt werden.

Es gibt erstens Kreise, die einen bestimmten Radius besitzen, und zweitens Quadrate mit einer bestimmten Seitenlänge (jeweils als `double`-Wert). Schreiben Sie dazu je eine Klasse `Kreis` und `Quadrat`. Es muss möglich sein, Kreise und Quadrate zusammen in eine `ArrayList` zu stecken und von allen Elementen der `ArrayList` den Flächeninhalt auszugeben.

Hinweise:

- Kreise und Quadrate sollen unveränderlich sein. Sie benötigen mindestens einen Konstruktor und eine getter-Methode für das Attribut. Werfen Sie eine `ArithmeticException`, wenn versucht wird, einen negativen Wert zu übergeben.
- Schreiben Sie eine objektorientierte Lösung. Sie müssen entweder eine zusätzliche Basisklasse oder ein Interface verwenden.
- Schreiben Sie ausserdem eine Testklasse mit einer Funktion `printFlaechen`, die eine ArrayList von Kreisen und Quadraten erhält (wie wählt man am besten den Übergabeparameter?) und alle Flächeninhalte auf dem Bildschirm ausgibt. Sie können die Funktion einfach ohne umgebende Klasse aufschreiben.
- Fügen Sie der Testklasse eine main-Funktion hinzu, die einen Kreis mit Radius 2 und ein Quadrat mit Seitenlänge 3 erzeugt und die Flächeninhalte beider Objekte durch (insgesamt) einen Aufruf von `printFlaechen` auf dem Bildschirm ausgibt.