Universität der Bundeswehr München
Elektrotechnik und Technische Informatik
Oswald / Baumann

Wintertrimester 2021
Praktikum 03
Abgabe: 01.02.-02.02.2021

Grundlagen der Programmierung

Programmieren mit Zuweisung

Aufgabe 1: Berechnung der Fakultät

Schreiben Sie eine Klassenmethode `public static int fac(int n)`, die die Fakultät einer natürlichen Zahl berechnet. Verwenden Sie zur Implementierung der Klassenmethode keine Rekursion, sondern eine `for`-Schleife. Die Fakultät ist wie folgt definiert:

$$\begin{aligned} 0! &= 1 \\ n! &= 1 \cdot 2 \cdot 3 \cdot .... \cdot n = \prod_{i=1}^{n} i \end{aligned}$$

Beachten Sie bei der Umsetzung zusätzlich die folgenden Anforderungen: Falls es sich bei dem Eingabeparameter um eine negative Zahl handelt, dann soll die Klassenmethode -1 zurückgeben. Falls der Rückgabewert größer als die größte darstellbar Zahl ist, die im Datenyp `int` gespeichert werden kann, dann soll die Klassenmethode auch -1 zurückliefern.

Den Wert der kleinsten bzw. größten darstellbaren ganzen Zahl stellt die Klasse `Integer` in den Konstanten `MIN_VALUE` bzw. `MAX_VALUE` zur Verfügung. Da sich die Klasse `Integer` im Package `java.lang` befindet, benötigt man keine zusätzliche `import`-Anweisung. Der Zugriff auf den Wert erfolgt z.B. durch: `Integer.MAX_VALUE`.

Testen Sie Ihre Methode und finden Sie heraus, bis zu welcher Zahl die Fakultät berechnet werden kann.

Aufgabe 2: Größter gemeinsamer Teiler

Schreiben Sie eine Klassenmethode `public static int ggT(int a, int b)`, die den größten gemeinsamen Teiler (ggT) zweier ganzer Zahlen berechnet. Verwenden Sie zur Implementierung der Klassenmethode keine Rekursion, sondern eine oder mehrere `while`-Schleifen. Verwenden Sie für die Umsetzung die Eigenschaft, die wir schon aus der Rekursion kennen: $ggT(a, b) = ggT(a, b - a)$ falls $b > a$ und $ggT(a, b) = ggT(a - b, b)$ falls $a > b$. Sie dürfen die Operatoren `%` (Modulo), `/` (Division) und `*` (Multiplikation) nicht verwenden.

Beachten Sie bei Ihrer Umsetzung die folgenden Anforderungen: Falls einer der beiden Eingabeparameter gleich oder kleiner 0 ist, dann soll die Klassenmethode -1 zurückgeben.

Aufgabe 3: Dreiecke malen

In dieser Aufgabe sollen Sie gleichseitige Dreiecke malen, und zwar mit Hilfe einer rekursiven Klassenmethode. Zum Malen der Dreiecke können Sie wieder die Klasse `SimpleGraphicPanel` verwenden.

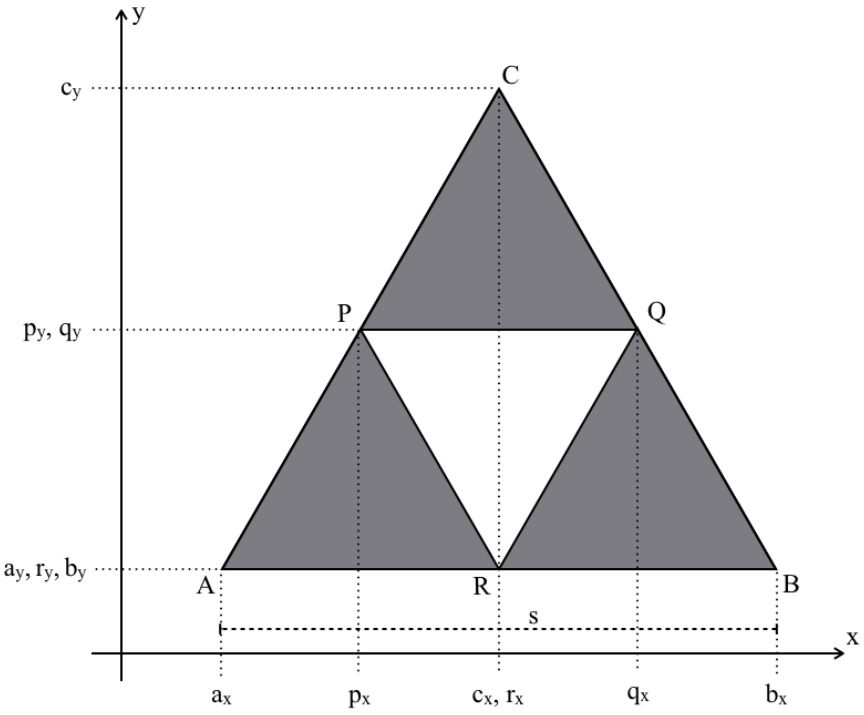

Abbildung 1: Gleichseitiges Dreieck

Für das Malen der Dreiecke benötigen wir die in der Abbildung 1 gegebenen Punkte. Wir können die Werte b_x, b_y, c_x, c_y, r_x, r_y, q_x, q_y, p_x und p_y sukzessive in Abhängigkeit von a_x, a_y und s berechnen:

```
bx = ax + s
by = ay
cx = (ax + bx) / 2.0
cy = ay + Math.sqrt(3.0) * (bx - ax) / 2.0
px = (ax + cx) / 2.0
py = (ay + cy) / 2.0
qx = (bx + cx) / 2.0
qy = (by + cy) / 2.0
rx = (ax + bx) / 2.0
ry = (ay + by) / 2.0
```

a) In der ersten Teilaufgabe sollen Sie die Dreiecke rekursiv malen. Schreiben Sie dazu eine rekursive Klassenmethode mit der folgenden Signatur:
`public static int maleDreieckRekursiv(double ax, double ay, double s, double t)`
In Abbildung 2 sehen Sie Dreiecke in verschiedenen Rekursionstiefen. Ist der Wert von $s \leq t$, dann soll Ihre Klassenmethode ein einfaches Dreieck mit den Ecken A, B und C malen. Ist dies nicht der Fall, also $s > t$, dann soll Ihre Klassenmethode sich selbst dreimal aufrufen. Für die Parameter `ax` und `ay` werden bei diesen drei Aufrufen die Punkte A, R und P verwendet. Der Parameter `s` wird halbiert und `t` bleibt gleich.

In Abbildung 2 sehen Sie das Ergebnis der Aufrufe:

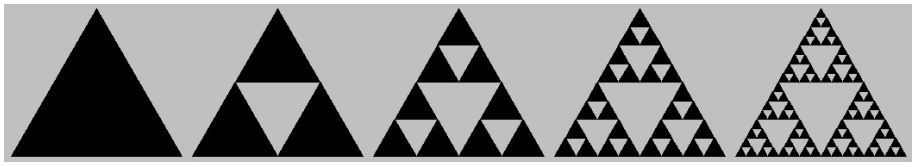

Abbildung 2: Dreiecke mit unterschiedlichen Aufrufparametern

```
maleDreieckRekursiv(10.0, 10.0, 150.0, 150.0)
maleDreieckRekursiv(170.0, 10.0, 150.0, 75.0)
maleDreieckRekursiv(330.0, 10.0, 150.0, 37.5)
maleDreieckRekursiv(490.0, 10.0, 150.0, 18.75)
maleDreieckRekursiv(650.0, 10.0, 150.0, 9.375)
```