

Übungen I: Grundelemente der Programmierung

Aufgaben

1. Welchen Wert enthält die Variable y nach der Abarbeitung dieser Anweisung:

```
double y = 5 / 3;
```

2. Schreiben Sie ein Programm, das die Werte zweier Variablen x und y untereinander vertauscht.
3. Schreiben Sie ein Programm, das eine quadratische Gleichung löst. Die allgemeine Formel zur Lösung quadratischer Gleichungen lautet:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(Hinweis: Zur Wurzelberechnung kann man in Java die Funktion `Math.sqrt()` verwenden.)

4. Schreiben Sie eine Anwendung, welche eine Liste aller Primzahlen von 1 bis 1000 ermittelt und diese auf der Konsole ausgibt. (Hinweis: Der Modulo-Operator („%“) ist hierbei hilfreich. Dadurch ermittelt man den Rest einer Division, also z.B. $7 \% 3$ ergibt 1)
5. Schreiben Sie ein Programm, das zwei Zahlen von der Konsole (in beliebiger Reihenfolge) einliest, dann die Summe aller Zahlen von der kleineren bis zur größeren Zahl ausrechnet und das Ergebnis wiederum auf der Konsole ausgibt. Werden also z.B. die Zahlen 5 und 3 eingelesen, soll als Ergebnis 12 ausgegeben werden, da $3 + 4 + 5 = 12$. Sie können zum Einlesen der beiden Zahlen folgendes Programmfragment verwenden:

```
int von, bis;  
Scanner sc = new Scanner (System.in);
```

```
System.out.print("Zahl1: ");  
von = sc.nextInt();
```

```
System.out.print("Zahl2: ");  
bis = sc.nextInt();
```

6. Implementieren Sie eine Funktion zur Berechnung der Quadratwurzel. Verwenden Sie dazu das Heron-Verfahren, bei dem der gegebene Radikand a als Flächeninhalt eines Quadrates verstanden wird. Um die Seitenlänge des Quadrates zu ermitteln, definiert man zunächst eine beliebige Seitenlänge x (idealerweise möglichst nahe an dem gesuchten Ergebniswert). Nun berechnet man die fehlende Seite so, dass der Flächeninhalt des dadurch entstandenen Rechtecks wieder a ergibt. Der Folgewert von x wird nun durch das arithmetische Mittel beider Seitenlängen bestimmt. Dies wird so oft wiederholt, bis beide Seiten (fast) gleich lang sind. x ist dann die Wurzel von a . Es ergibt sich folgende Formel:

$$x_{n+1} = \frac{1}{2} * \left(x_n + \frac{a}{x_n} \right)$$

7. Schreiben Sie eine Funktion `ggT()`, die als Parameter zwei `int`-Variablen mit den Namen `a` und `b` erhält und aus beiden den größten gemeinsamen Teiler ermittelt (Hinweis: Der GGT ist die größte Zahl, durch die man `a` und `b` ganzzahlig teilen kann).
8. Schreiben Sie eine Funktion `kgv()`, die als Parameter zwei `int`-Variablen mit den Namen `a` und `b` erhält und aus beiden das kleinste gemeinsame Vielfache ermittelt (Hinweis: Das KGV ist die kleinste Zahl, durch die `a` und `b` ganzzahlig teilbar ist).
9. Schreiben Sie eine Funktion zur Berechnung der Quersumme einer Zahl (Hinweis: Die Quersumme ergibt sich aus der Summe der einzelnen Ziffern einer Zahl).
10. Implementieren Sie sowohl eine iterative als auch eine rekursive Funktion zur Berechnung der Fibonacci-Zahl an einer gegebenen Stelle x . Für Fibonacci-Zahlen gilt folgendes Bildungsgesetz:

$$fib(x) = \begin{cases} 0 & \forall x = 0 \\ 1 & \forall x = 1 \\ fib(x-1) + fib(x-2) & \forall x > 1 \end{cases}$$

Daraus ergibt sich diese Folge: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ...

11. Im Folgenden ist das sog. *Pascal'sche Dreieck* dargestellt:

```

          1
        1 1
       1 2 1
      1 3 3 1
     1 4 6 4 1
    1 5 10 10 5 1
  
```

Ein Wert in dem Pascal'schen Dreieck ergibt sich grundsätzlich aus der Summe der beiden darüberliegenden Werte. Ausnahmen bilden die Spitze des Dreiecks, die immer den Wert 1 annimmt, und die Werte am linken und am rechten Rand des Dreiecks, die nur einen darüberliegenden Wert besitzen und daher lediglich aus diesem abgeleitet sind. Schreiben Sie zunächst eine Funktion `pascal()`, der beim Aufruf als Parameter Zeilen- und Spaltennummer übergeben werden und die den entsprechenden Wert in dem Pascal'schen Dreieck zurückgibt. Versuchen Sie, die Funktion rekursiv zu definieren. Meines Erachtens kommen Sie am leichtesten auf eine Lösung, wenn Sie sich diese anhand der folgenden Darstellung überlegen, und nicht anhand der obigen (zentrierten):

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
  
```

Haben Sie eine funktionierende `pascal()`-Funktion implementiert, erzeugen Sie in der `main()`-Funktion ein vollständiges Dreieck unter Verwendung der `pascal()`-Funktion.

(Hinweis: Um Zahlen rechtsbündig auszugeben, empfiehlt sich die Funktion `System.out.printf()`. Mit folgender Anweisung geben Sie beispielsweise den Wert der `int`-Variablen `x` rechtsbündig in einem Bereich von 8 Stellen aus:

```
System.out.printf("%8d", x);
```

Weitere Informationen zur `printf()`-Funktion finden Sie in der Java-Dokumentation).

12. Schreiben Sie eine Funktion, die eine gegebene Dezimalzahl als Binärzahl auf der Konsole ausgibt. Die Umrechnung von Dezimal- in Binärzahlen veranschaulicht folgendes Beispiel:

44 : 2 = 22 Rest **0**
22 : 2 = 11 Rest **0**
11 : 2 = 5 Rest **1**
5 : 2 = 2 Rest **1**
2 : 2 = 1 Rest **0**
1 : 2 = 0 Rest **1**



Die jeweiligen Reste werden
umgekehrt zusammengefügt.
44 dezimal ergibt demnach 101100
in Binärdarstellung.

(Hinweis: Eine rekursive Lösung bietet sich hier an).

13. Erweitern Sie Aufgabe 4 dahingehend, dass die Primzahlen von 1 bis 1000 in einem Array gespeichert werden und dieser Array dann auf der Konsole ausgegeben wird. Beachten Sie, dass von vorneher ein nicht bekannt ist, wie viele Primzahlen es von 1 bis 1000 gibt. Der Array muss groß genug sein, um alle Primzahlen zu fassen. Die Primzahlen zweimal zu ermitteln, sie also erst zu zählen und dann den Array zu füllen, ist ineffizient. Der Array sollte auch nicht zu groß (z.B. Länge = 1000) definiert werden. Dabei würde unnötig Speicher verschwendet werden. Gehen Sie daher einen Kompromiss zwischen maximaler Performance und minimaler Speicherbelastung ein.
14. Zeigen Sie, dass man Verzweigungen auch mithilfe von Schleifen darstellen kann.