03 – Verständnisfragen zur selbstständigen Beantwortung

(1) Nenne die drei Programmprinzipien.

(2) Lassen sich alle Schleifentypen ineinander überführen? Wenn ja, warum brauchen wir die unterschiedlichen Varianten überhaupt?

(3) Ist das Einrücken von Programmcode eine sinnvolle Sache?

(4) Warum ist die Auslagerung von Funktionen sinnvoll?

03 – Aufgabensammlung

(1) Versuchen Sie ihr Lieblingsrezept in Pseudocode zu notieren und anschließend in ein Aktivitätsdiagramm zu übertragen.

(2) Überlegen Sie sich Fälle, bei denen ein Programm nicht terminiert. Verwenden Sie für die Erläuterung ebenfalls Aktivitätsdiagramme.

(3) Gehen Sie die einzelnen Schritte aus Abschnitt 2.3.2 durch und bringen Sie das Programm `ProgrammEins` zum Laufen:

```
public class ProgramEins {
   public static void main(String[] args) {
      System.out.println("Endlich ist es soweit! Mein erstes Programm läuft...");
   }
}
```

(4) Geben Sie ein Programm in Java an, das folgende Formeln in jeweils separaten Funktionen berechnet:

a) $f_1(x) = x$ b) $f_2(x) = \frac{x^2}{2} + 17 \cdot 2$ c) $f_2(x) = ((x-1)^3 - 14)/2$

03 – Aufgabensammlung

(5) Schreiben Sie ein Programm, das für `i=1,2,...,20` die Fakultätsfunktion berechnet und die Funktionswerte zeilenweise ausgibt. Die Fakultätsfunktion ist wie folgt definiert:

$$fakultaet(n) = 1 \cdot 2 \cdot \cdots \cdot n = \prod_{i=1}^{n} i = n!$$

Beispiel: `fakultaet(4) = 1 · 2 · 3 · 4 = 24`

(6) Geben Sie für die folgenden Summen entsprechende `for`-Schleifen an:
a) $\sum_{i=0}^{28}(i-1)^2$ b) $\sum_{i=0}^{100}\frac{i\cdot(i+1)}{2}$ c) $\sum_{i=1}^{25}\frac{i+1}{i}$

(7) Überführen Sie die folgenden Schleifen in `for`- bzw. `while`-Schleifen:

```
for (int x=7; x<12; x++)
   <Anweisung>
```

```
for (int y=0; x=10; x>y; y++, x--)
   <Anweisung>
```

```
int a = 1024;
while (a>2) {
   <Anweisung>
   a=a/2;
}
```


03 – Aufgabensammlung

(8) In Abschnitt 1.7.2 wurden Typumwandlungen mittels Casten vorgestellt. Überprüfen Sie, ob der größte darstellbare Wert für einen `long` in einen `float` passt (kleiner Hinweis: der größte darstellbare `long` ist `Long.MAX_VALUE`), indem Sie zunächst den Inhalt des `long` in den `float` speichern, zurückcasten und beide, den Startwert und den neuen Wert vergleichen.

(9) Erklären Sie warum `Integer.MIN_VALUE-1` zu einer positiven Zahl führt.

(10) Welches Ergebnis liefern die beiden Programmzeilen `c = c++;` und `c = ++c;` und warum?