

4. Musterlösung zur Vorlesung  
Einführung in die Informatik:  
Programmierung und Software-Entwicklung

**Aufgabe 4-1. (4 Punkte)** (Abgabeformat: .txt oder .pdf)

Gegeben ist das rechts abgedruckte fürchterliche Programm. Berechnen Sie die Ausgabe des fürchterlichen Programms mit Papier & Bleistift, also *ohne* Computer. Geben Sie zusätzlich Schritt für Schritt an, welche Variable in welcher Programmzeile welchen absoluten Wert zugewiesen bekommt (d.h. Angaben wie "Zeile 5: x = 28" und *nicht* "Zeile 5: x = y+8").

*Beispiel:*

Für das kleine Programm

```
10 int x = 0;
20 int y = 3;
30 for (int i = 1; i <3; i++){
40  x += y;
50 }
60 System.out.println(x);
70 y = x + 1;
```

würden wir von Ihnen diese Angabe erwarten:

Zeile	Zuweisung
10	$x = 0$
20	$y = 3$
30	$i = 1$
40	$x = 3$
30	$i = 2$
40	$x = 6$
30	$i = 3$
70	$y = 7$

```
010 int x = 42;
020 {
030   int y = 36;
040   {
050     while (x > y)
060     {
070       int z = x / y;
080       z*=2;
090       z++;
100       y+= z;
110       if (z<=3)
120         z*=2;
130       else
140         z--;
150       x++;
160     };
170     if (x != y)
180       if ( x < 44)
190         y/=2;
200       else
210         x/=3;
220     else
230       x+=11;
240     for (int z = 0; z<4; z*=x/y) {
250       x+=4;
260       y+=z;
270       z++;
280     }
290   }
300   System.out.println(x+y);
310 }
```

**Lösung:** Das Programm gibt am Ende den String "123" aus.

Zeile	Zuweisung	Kommentar
010	$x = 42$	
030	$y = 36$	1.Schleife
070	$z = 1$	
080	$z = 2$	
090	$z = 3$	
100	$y = 39$	
120	$z = 6$	If-Zweig
150	$x = 43$	2.Schleife
070	$z = 1$	
080	$z = 2$	
090	$z = 3$	
100	$y = 42$	
120	$z = 6$	If-Zweig
150	$x = 44$	3.Schleife
070	$z = 1$	
080	$z = 2$	
090	$z = 3$	
100	$y = 45$	
120	$z = 6$	If-Zweig
150	$x = 45$	Schleifenabbruch
230	$x = 56$	Else-Zweig
240	$z = 0$	1.Schleife
250	$x = 60$	
260	$y = 45$	$y$ unverändert
270	$z = 1$	
240	$z = 1$	2.Schleife
250	$x = 64$	
260	$y = 46$	$y$ unverändert
270	$z = 2$	
240	$z = 2$	3.Schleife
250	$x = 68$	
260	$y = 48$	$y$ unverändert
270	$z = 3$	
240	$z = 3$	4.Schleife
250	$x = 72$	
260	$y = 51$	$y$ unverändert
270	$z = 4$	Schleifenabbruch
300		Ausgabe: "123" (=72+51)

**Aufgabe 4-2. (4 Punkte)** (Geben Sie alle kompilierbaren .java-Dateien Ihrer Lösung ab)

In dieser Aufgabe wollen wir eine gegebene Zahl in ein sortiertes Array an die richtige Stelle einfügen. Da ein Array in Java nicht erweitert werden kann, müssen wir das Array dazu in ein größeres umkopieren. Vervollständigen Sie das folgende Programm, welches eine Zahl zwischen 0 und 128 von der Tastatur einliest, und diese dann in eine Kopie eines bereits sortierten Arrays `liste1` an der richtigen Stelle einfügt.

Beachte, dass Array `liste1` nur ein Beispiel ist. Ihr Programm muss auch mit einem beliebigen anderen sortierten `int`-Array funktionieren. Testen Sie dies durch verschiedene Änderungen des Arrays in der erste Zeile.

```
int[] liste1 = new int[] { 1, 3, 5, 6, 12, 32, 55, 69, 99, 125 };
/* Ihre Aufgabe #1
 * Prüfe ob Variable zahl größer gleich 0 und kleiner gleich 128 ist,
 * und fordere ggf. so lange zur Eingabe auf, bis das der Fall ist.
 */
int zahl = -1;
while ( zahl < 0 | zahl > 128 ) {
    System.out.print("Geben Sie eine Zahl zwischen 0 und 128 ein: ");
    zahl = Input.readInt();
}
int[] liste2 = new int[liste1.length + 1];
/* Ihre Aufgabe #2
 * Kopiere liste1 in das Array liste2, und füge dabei die Variable zahl
 * an die richtige Stelle ein, so dass liste2 hinterher sortiert ist.
 */
int j = 0;
while (j < liste1.length && liste1[j] < zahl)
{
    liste2[j] = liste1[j];
    j++;
}
liste2[j] = zahl;
while (j < liste1.length) {
    liste2[j + 1] = liste1[j];
    j++;
}
System.out.println("Eine sortierte Liste:");
for (int i=0; i<liste2.length; i++)
{
    System.out.print(" "+liste2[i]);
}
System.out.println();
```

**Aufgabe 4-3. (4 Punkte)** (Geben Sie alle kompilierbaren .java-Dateien Ihrer Lösung ab)

Schreiben Sie ein Programm, das den Benutzer zuerst dazu auffordert, einen String einzugeben und welches dann anschließend eine Meldung ausgibt, ob der eingegebene String ein Palindrom ist.

Wie in der Vorlesung besprochen, ist ein Palindrom ein String, welcher rückwärts gelesen identisch zu sich selbst ist, z.B. "Ton tut not", "Amok oma" oder auch "Relieffeiler". Wie man an den Beispielen sieht, spielen Leerzeichen, Groß- und Kleinschreibung für Palindrome eigentlich keine Rolle. Zur Vereinfachung der Aufgabe reicht es aber auch aus, wenn Ihr Programm wenigstens Palindrome akzeptiert, welche tatsächlich identisch sind. Also keines der obigen drei Beispiele müsste akzeptiert werden, wohl aber eine Eingabe wie "ton tut not". Benutzen Sie zur Eingabe wie bisher die Klasse Input.java von Übungsblatt 1.

Ein Problem bei der Prüfung ob der eingegebene String ein Palindrom ist, liegt darin, dass Sie nicht vorhersagen können, welche Länge der eingegebene String hat. Dies können Sie aber mit Hilfe von Arrays und einer While-Schleife lösen! Strings und Char-Arrays können übrigens so ineinander konvertiert werden:

```
String toll = "Wie toll ist das denn?";
char[] schoen = toll.toCharArray();
String super = new String(schoen);
```

Hinweis: Vergleiche zwischen Strings werden zur Lösung dieser Aufgabe *nicht* benötigt. Es sei dennoch daran erinnert, dass Vergleiche zwischen Strings nicht mit == sinnvoll durchgeführt werden können, im Gegensatz zu Vergleichen zwischen Chars.

**Lösung:**

```
System.out.print("Gib ein Palindrom ein: ");
String palindrom = Input.readString();
char[] paliarray = palindrom.toCharArray();
// Wir merken uns das Ergebnis der Vergleiche in einer
// Variablen des Typs Boolean
Boolean ispali = true;
// Für die Schleife reicht ein halber Durchlauf;
// bei einer ungeraden Länge schauen wir uns den
// mittleren char-Wert erst gar nicht an.
for (int j = 0; ispali && j < (paliarray.length / 2); j++) {
    // Werte des Typs char können wir direkt mit == vergleichen
    ispali = paliarray[j] == paliarray[paliarray.length - (j + 1)];
    // Da wir die Schleife sofort abbrechen, wenn ispali falsch
    // wird, brauchen wir nicht "ispali = ispali & ..." schreiben
}
if (ispali) {
    System.out.println("Gratulations! It's a palindrome!");
} else {
    System.out.println("Not a palindrome!");
}
```

**Abgabe:** Sie können ihre Lösungen bis Montag, den 15.11., 14 Uhr über UniWorx abgeben. Java Dateien, welche nicht kompilieren, werden nicht beachtet!