

## Aufgabenblatt 02: IO, Collections, Operatoren

### 1 Listen und IO

Erzeugen Sie bitte 1000 komplexe Zahlen nach folgender Regel

$$\begin{aligned} z_0 &= 0 \\ z_{n+1} &= z_n^2 + c \end{aligned}$$

für einen Wert von  $c = -0.7 + 0.2i$ . Sie können auch mit anderen Werten experimentieren.

Schreiben Sie die Darstellung dieser Zahlen, die *toString* liefert, in eine Textdatei, eine Stringdarstellung pro Zeile!

Lesen Sie bitte diese komplexen Zahlen, d. h. Darstellungen als String einer Klasse *Complex*, aus dieser Textdatei ein, erzeugen Sie daraus Objekte und speichern Sie diese bitte in einer Liste und in einem Array.

Sortieren Sie die Zahlen nach absolutem Betrag mit Methoden aus den Klassen *Collections* bzw. *Arrays* (*sort*, *parallelSort*) und schreiben Sie die Ergebnisse in weiteren Textdateien. Überprüfen Sie in *JUnit*-Testfällen, dass Sie korrekt sortiert haben.

Sie können gerne „Umwege“ über weitere Klassen nehmen, wenn es Ihnen die Programmierung vereinfacht (Liste von String's o.ä.).

### 2 Operatoren

Ich empfehle, dass Sie diese Aufgaben zunächst versuchen „theoretisch“ zu lösen! Dazu sollten Sie auch in die Java Language Specification (JLS) gucken. Überprüfen Sie möglichst dann erst Ihre Theorie durch Ausführen des Codes! Ich habe aus der Lösung solcher Aufgaben (mehr als 100) viel gelernt.

1. Ein Java-Anfänger hat folgende Methode geschrieben, um zu prüfen, ob eine ganze Zahl ungerade ist:

```
public static boolean isOdd(long i) {  
    return i % 2 == 1;  
}
```

Ist die Methode korrekt? Wenn ja, warum, wenn nicht, warum? Was liefert sie und wie macht man das richtig?

2. In der folgenden Methode werden drei Casts verwendet:

```
System.out.println((int) (char) (byte) -1);
```

Welches Ergebnis erwarten Sie? Welches kommt tatsächlich heraus? Was passiert genau?

**Hinweis:** Lesen mal in der JLS unter „narrowing and widening conversions“ nach!

3. Der folgende Code soll zwei ganze Zahlen vertauschen.

```
10      int x = 1984;
20      int y = 2001;
30      x ^= y ^= x ^= y;
40      System.out.println("x = " + x + "; y = " + y);
```

Was wird ausgegeben? Erläutern Sie bitte genau, wie das Ergebnis zu Stande kommt!

**Hinweis:** Liste der Operatoren, insbesondere bitweise Operatoren am Anfang von Abschnitt 5.6 im Skript.

4. Was schreibt das folgende Programm auf die Konsole?

```
/**
 * Generated by the IBM IDL-to-Java compiler, version 1.0
 * from F:\TestRoot\apps\al\units\include\PolicyHome.idl
 * Wednesday, June 17, 1998 6:44:40 o'clock AM GMT+00:00
 *
 * What's wrong here?
 */
public class Test {
    public static void main(String[] args) {
        System.out.print("Hell");
        System.out.println("o world");
    }
}
```

Warum der Code zu dem Effekt führt, habe ich in der Vorlesung nicht behandelt!

5. Was gibt der folgende Code auf der Console aus?

```
public class Increment {  
    public static void main(String[] args) {  
        int j = 0;  
        for (int i = 0; i < 100; i++)  
            j = j++;  
        System.out.println(j);  
    }  
}
```

Erklären Sie bitte warum!

Abgabetermin:

Donnerstag, 19.04.2018