



Programmierung für Mathematik HS11

Übung 2

1 Aufgabe: Calculator Fortsetzung

1.1 Lernziele

1. Probleme bei der Benutzung von Operatoren erkennen und verstehen.
2. Probleme mit Integer-Rechnung kennenlernen und Lösungen dafür finden.

1.2 Aufgabenstellung

a) Operatoren und Klammern

Betrachten Sie nochmals die Methoden `add`, `subtract` und `multiply` aus der `Calculator`-Klasse vom Übungsblatt 1 und vergleichen Sie sie mit den Untenstehenden. Wo liegen die Unterschiede zwischen den Methoden? Versuchen die Auswirkungen der Veränderungen vorauszusagen, überprüfen Sie anschliessend ihre Ergebnisse, indem Sie ihre eigene `Calculator`-Klasse anpassen und versuchen Sie schliesslich, das unterschiedliche Verhalten zu erklären.

```
1 public void add(int number1, int number2) {
2     System.out.println("The sum of " + number1 + " and " +
3         number2 + " is " + number1 + number2);
4 }
5
6 public void subtract(int number1, int number2) {
7     System.out.println("The difference of " + number1 + " and " +
8         number2 + " is " + number1 - number2);
9 }
10
11 public void multiply(int number1, int number2) {
12     System.out.println("The product of " + number1 + " and " +
13         number2 + " is " + number1 * number2);
14 }
```

Listing 1: Veränderte Methoden `add`, `subtract` und `multiply`

b) Grosse Zahlen addieren

Testen Sie die Methode `add` von der Klasse `Calculator` vom Übungsblatt 1 mit den Zahlen 2000000000 und 1000000000. Was fällt Ihnen auf und wie erklären Sie sich das? Was wäre eine einfache Lösungsmöglichkeit für dieses Problem? Schreiben Sie die Methode `add` so um, dass Sie auch für die Zahlen 2000000000 und 1000000000 das korrekte Ergebnis liefert.

c) Division von Integervariablen

Erweitern Sie Ihre `Calculator`-Klasse schliesslich noch um die Methode `divide`, welche den Quotient zweier Zahlen vom Typ `int` berechnen soll. (Das Resultat soll als Dezimalzahl angezeigt werden. Man darf voraussetzen, dass der Benutzer als zweite Zahl nicht 0 eingibt.)

d) TestDriver

Testen Sie zum Schluss sowohl die veränderte `add` Methode mit den Zahlen 2000000000 und 1000000000 als auch Ihre `divide` Methode mit den Zahlen 5 und 2.

2 Aufgabe: Rechnen mit komplexen Zahlen

2.1 Lernziele

1. Komplexe Zahlen in Java-Code realisieren können.
2. Vorgegebene Klassen in eigenem Code einsetzen können.

2.2 Aufgabenstellung

Gegeben ist die Klasse `ComplexNumber`, welche eine komplexe Zahl $a + ib$ darstellen soll.

```
1 public class ComplexNumber{
2
3     private double a;
4     private double b;
5
6     public double getA() {
7         return a;
8     }
9
10    public void setA(double newA) {
11        a = newA;
12    }
13
14    public double getB() {
15        return b;
16    }
17
18    public void setB(double newB) {
19        b = newB;
20    }
21
22    public String toString() {
23        String result = a + " + i*" + b;
24        return result;
25    }
26
27 }
```

Listing 2: `ComplexNumber` Klasse

Speichern Sie diese Klasse unter `ComplexNumber.java` ab. Schreiben Sie eine neue Klasse `ComplexCalculator`, welche die Methoden `add`, `subtract`, `multiply`, `divide` und `modulus` implementiert. Die Methoden sollen dabei jeweils eine Addition, eine Subtraktion, eine Multiplikation oder eine Division von zwei komplexen Zahlen, bzw. den Betrag einer komplexen Zahl berechnen und das Ergebnis auf der Konsole ausgeben. Testen Sie anschliessend ihre `ComplexCalculator` Klasse mittels eines TestDrivers.

Für das Rechnen mit komplexen Zahlen gelten dabei folgende Formeln:

$$(a + i \cdot b) + (c + i \cdot d) = (a + c) + i \cdot (b + d)$$

$$(a + i \cdot b) - (c + i \cdot d) = (a - c) + i \cdot (b - d)$$

$$(a + i \cdot b)(c + i \cdot d) = ac + i \cdot bc + i \cdot ad + i^2 \cdot bd = (ac - bd) + i \cdot (bc + ad)$$

$$\begin{aligned} \frac{a + i \cdot b}{c + i \cdot d} &= \frac{(a + i \cdot b)(c - i \cdot d)}{(c + i \cdot d)(c - i \cdot d)} = \frac{ac - i \cdot ad + i \cdot bc - i^2 \cdot bd}{c^2 - (i \cdot d)^2} \\ &= \frac{ac - i \cdot ad + i \cdot bc - (-1) \cdot bd}{c^2 - i^2 \cdot d^2} = \frac{(ac + bd) + i \cdot (bc - ad)}{c^2 - (-1) \cdot d^2} \\ &= \frac{ac + bd}{c^2 + d^2} + i \cdot \frac{bc - ad}{c^2 + d^2} \end{aligned}$$

$$|a + i \cdot b| = \sqrt{a^2 + b^2} \in \mathbb{R}$$

Die Wurzel einer positiven Zahl können Sie dabei mit der Methode `sqrt()` aus der Klasse `Math` berechnen. Bspw. liefert `Math.sqrt(2)` die Wurzel aus zwei. Der Rückgabewert der Methode ist dabei vom Typ `double`.

3 Aufgabe: Code Snippets

3.1 Lernziele

1. Java API kennenlernen.
2. Code verstehen können.

3.2 Aufgabenstellung

Überlegen Sie sich, was der Output bei folgenden Code-Ausschnitten wäre, wenn sie so in einer main-Methode stehen würden. Versuchen Sie das Ergebnis herauszufinden, ohne den Code selbst auszuführen.

Tipp: benutzen Sie die [Java API](#) um die vorimplementierten Java-Methoden zu verstehen, in dem folgenden Code werden Methoden von den Klassen `Math`, `String` und `Character` verwendet.

a)

```
1 int a = 2000;
2 int b = a++ + a++;
3 a += b++;
4 System.out.println(a);
5 System.out.println(b);
```

Listing 3: Ausschnitt 1

b)

```
1 String s1 = "animal";
2 String s2 = "dog";
3 System.out.println((Character.toString(s1.charAt(0))).toUpperCase()
4     + " " + s2 + " " + (Character.toString(s1.charAt(2))).concat("s")
5     + " " + s1.substring(0,2) + " " + s1);
```

Listing 4: Ausschnitt 2

c)

```
1 System.out.println(Math.round(20/7));
2 System.out.println(Math.round(20.0/7));
3 System.out.println(Math.ceil(20.0/7));
4 System.out.println(Math.floor(20.0/7));
5 System.out.println(Math.signum(20.0/7));
```

Listing 5: Ausschnitt 3

d)

```
1 System.out.println(Math.log(1));  
2 System.out.println(Math.exp(0));  
3 System.out.println(Math.cos(Math.PI));  
4 System.out.println(Math.cbrt(8));
```

Listing 6: Ausschnitt 4