

Programmierung (PRG)

Übungen Block 02

Version: 1.1

Dozent: Jordan Šućur

Nacharbeitung Input

Arbeiten Sie noch einmal den Theorie Input zu Block 02 (Foliensatz) nach.

Studium des Buches

Die behandelten Themen finden Sie im Buch in Kapitel 5, Kapitel 6 (6.1, 6.2 und 6.4.1) und Kapitel 7 (ausser 7.2 und 7.6.6). Sollten Sie mit dem im Unterricht behandelten Stoff noch Mühe haben, wird empfohlen, die genannten (Unter)Kapitel im Buch durchzuarbeiten. Wenn etwas unklar ist, halten Sie es fest, um bei nächster Gelegenheit mit Ihren Mitstudierenden oder Dozierenden darüber diskutieren zu können.

Aufgaben im Buch

Lösen Sie folgende Aufgaben:

- Kapitel 5: Aufgabe 5.2.2, 5.2.3, 5.2.4, 5.2.5 und 5.2.6
- Kapitel 7: Aufgabe 7.1.1, 7.1.2, 7.3.1, 7.3.2 (ohne g, j und m), 7.5.1 und 7.5.2

Lösen Sie danach die Aufgaben 1, 2, 3, und 4 aus diesem Dokument.

Hinweis:

Es wird empfohlen, zuerst die Aufgaben im Buch zu lösen und erst danach die restlichen Aufgaben (Aufgabe 1 – Aufgabe 4) in Angriff zu nehmen.

Wichtig:

Den gesamten Output müssen Sie an der Zwischen- und Schlussprüfung auch ohne Computer notieren können. Bevor Sie den Code in ihre Entwicklungsumgebung hineinkopieren und laufen lassen, überlegen Sie sich, wie die jeweiligen Ausgaben aussehen werden und halten Sie diese schriftlich fest (um es anschliessend kontrollieren zu können).

Eine Tabelle mit Prioritäts- und –Assoziativitätsregeln von Operatoren in Java finden Sie im Bedarfsfall im Buch (Seite 275).

Aufgabe 1

Für eine neue Küche soll ein Kredit im Betrag von CHF 23 859.00 aufgenommen werden. Der Kredit soll bei einer lokalen Bank aufgenommen werden, die folgendes Angebot offeriert:

- Die Laufzeit beträgt 36 Monate.
- Der jährliche Zinssatz beträgt 4.85 %.
- Die Abschlusskosten von 259.00 CHF kommen noch dazu.

Berechnen Sie den Betrag, der jeden Monat von dem Kreditnehmer an die Bank während 3 Jahre überwiesen werden muss¹. Versuchen Sie dabei, auf die Verwendung von Formeln für die Zinsberechnung zu verzichten: Für drei Jahre kann man es auch ohne Formeln problemlos berechnen.

Aufgabe 2 (eine "etwas komplexere" Berechnung)

Für eine Produktionsmaschine sollen die folgenden Parameter eingegeben werden können:

- Maschinenleistung (Stück pro Stunde)
- Maschinenkosten (CHF pro Stunde)
- Lohnkosten (CHF pro Stunde)
- Auftragskosten (Fixkosten pro Auftrag)
- Stückzahl (Menge der zu produzierenden Einheiten)

Teilaufgabe 1

Schreiben Sie ein Programm, mit dem der zeitliche Aufwand (in Stunden) für einen beliebigen Auftrag berechnet werden können. Der zeitliche Aufwand hängt von der gewünschten Stückzahl und Maschinenleistung (Stück pro Stunde).

Teilaufgabe 2

Erweitern Sie das Programm anschliessend so, dass die Produktionskosten (für einen beliebigen Auftrag) berechnet werden. Überlegen Sie sich dabei zuerst, wie die Produktionskosten berechnet werden und halten Sie diesen Algorithmus schriftlich fest (z.B. in Textform). Erst wenn dieser Algorithmus steht, soll er im Programm abgebildet (geschrieben) werden.

Hinweis:

Sollten Sie trotz allen Bemühungen nicht wissen, wie die Berechnung aussehen könnte, sehen Sie sich den (etwas vereinfachten) Algorithmus auf der letzten Seite des Dokuments an.

¹Monatsrate

Teilaufgabe 3

Ergänzen Sie das Programm so, dass bei einer Stückzahl von weniger als 100 ein Preiszuschlag von 20 % berechnet wird. Als Basis für den Preiszuschlag dient der Produktionskostenbetrag.

Teilaufgabe 4 (fakultativ)

Ergänzen Sie das Programm so, dass bei einer Stückzahl von 1000 und mehr ein Rabatt von 5 % gewährt wird. Als Basis für den Preiszuschlag dient der Produktionskostenbetrag.

Hinweis:

Bei den Teilaufgaben 3 und 4 soll der ternäre Operator "?" verwendet werden.

Aufgabe 3

Gegeben sei das folgende Codefragment, in dem versucht wurde, die Berechnung des Endkapitals (nach beliebig vielen Jahren) zu realisieren.

```
public class EndkapitalRechner {

    public static void main(String[] args) {
        /* Scanner erstellen */
        Scanner sc = new Scanner(System.in);

        /* Startkapital einlesen */
        System.out.print(Bitte Startkapital eingeben: );
        int startKapital = scanner.nextDouble();

        /* Zinssatz einlesen */
        System.out.print(Bitte Zinssatz in Prozent eingeben: );
        int zinssatz = scanner.nextDouble();

        /* Anzahl Jahre einlesen */
        System.out.print(Bitte Anzahl Jahre eingeben: );
        int anzahlJahre = scanner.nextInt();

        /* Endkapital berechnen und Ausgeben */
        double endKapital = startKapital * Math.pow((1 + zinssatz / 100),
            anzahlJahre);
        System.out.println("\nEndkapital: endKapital);
```

```
}  
}
```

Der obige Code enthält diverse Syntaxfehler und kann nicht übersetzt werden.

- Versuchen Sie ohne Hilfe des Compilers möglichst viele Fehler zu identifizieren und durch passende Korrekturen zu beseitigen.
- Kompilieren Sie anschliessend das Programm und kontrollieren Sie, ob das Kompilieren erfolgreich war oder nicht. Falls nicht, sehen Sie sich die Fehlermeldungen des Compilers genau an und versuchen Sie mit deren Hilfe die restlichen Fehler zu identifizieren und zu beseitigen.
- Führen Sie anschliessend das Programm aus und kontrollieren Sie, ob das Programm auch semantisch korrekt ist.

Formel für Endkapital-Berechnung:

$$Endkapital = Startkapital * \left(1 + \frac{Prozentsatz}{100}\right)^{AnzahlJahre}$$

Hinweis:

Für die Potenzrechnung wird hier die Methode `Math.pow` (basis, exponent) verwendet (mehr dazu in der API-Dokumentation der Klasse `Math`).

Aufgabe 4

Gegeben sei das folgende Programm, in dem diverse arithmetische Operatoren eingesetzt werden.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int x = 10, y = 20, z = 30;  
        System.out.println("y: " + y++);  
        System.out.println("z: " + --z);  
  
        int p = x-- + y-- + --z;  
        System.out.println("p: " + p);  
  
        x = 10;  
        y = 133;  
        System.out.println(x + "%" + y + " = " + x % y);  
        System.out.println(y + "%" + x + " = " + y % x);  
    }  
}
```

```

    int a = 3, b = 8, c = 12, res = 4;
    res += a * b;
    System.out.println("res: " + res);
}
}

```

Geben Sie die Programmausgabe an, die das obige Programm bei einer Ausführung produzieren würde.

Hinweise:

- Die Aufgabe ist zuerst ohne Zuhilfenahme des Compilers zu lösen. Nehmen Sie evtl. die Tabelle mit Assoziations- und Prioritätsregeln für Operatoren zur Hilfe.
- Machen Sie eine Hilfstabelle, mit der Sie die einzelnen Variablen und Teilausdrücke bzw. deren Änderungen übersichtlich verwalten können.

Aufgabe 5

Sehen Sie sich die nachfolgend gegebenen Ausdrücke genau an.

```

boolean a = false, b = true, c = true, d = false, v = false;
int p = 9, q = 10;

```

// Ausdruck A

```

d = !b;                                // d =

```

// Ausdruck B

```

v = !a || b && d || !c && !b;          // v =

```

// Ausdruck C

```

v = a && b && (p++ < q);                // v =    p =    q =

```

// Ausdruck D

```

v = a || b && (p++ < q);                // v =    p =    q =

```

- Geben Sie jeweils den Rückgabewert an, der von dem jeweiligen Ausdruck zurück geliefert wird (rechts von der jeweiligen Codezeile)
- Erstellen Sie ein kleines Programm, in dem die obigen Ausdrücke passend integriert werden und führen Sie es aus. Entspricht die Programmausgabe dem, was Sie erwartet haben? Falls nicht, probieren Sie den Fehler zu finden, den Sie bei der Auswertung gemacht haben.

Hinweis:

Nehmen Sie evtl. die Tabelle mit Assoziativitäts- und Prioritätsregeln für Operatoren zur Hilfe.

Anhang

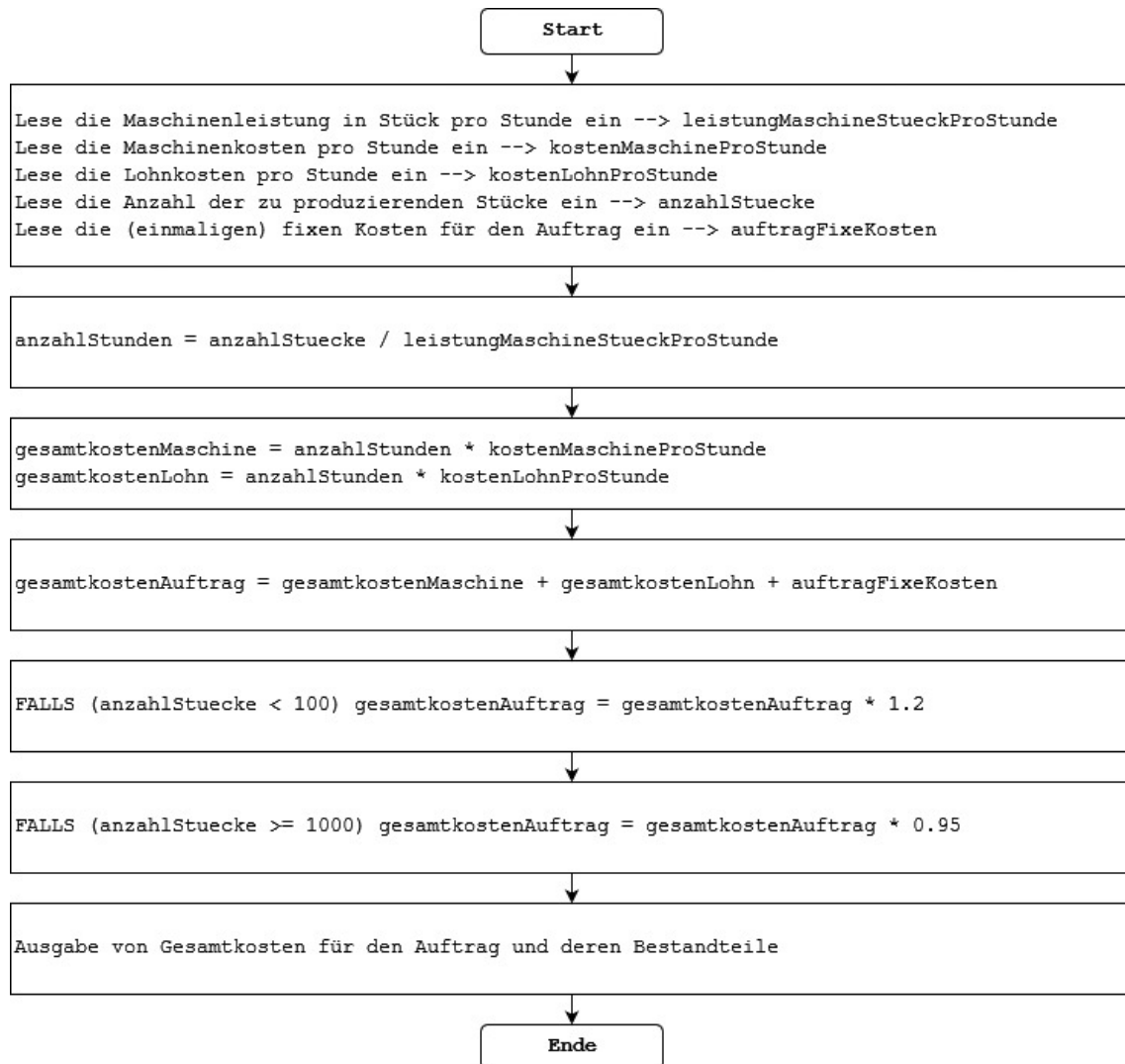


Abbildung 1: Produktionskosten-Berechnung