

Prüfung Softwareentwicklung I (IB)

Datum : 01.02.2018, 14:30 Uhr
Bearbeitungszeit : 90 Minuten
Prüfer : Prof. Dr. Oliver Braun
Hilfsmittel : Keine
Erreichbare Punkte : 90

Name: _____

Vorname: _____

Matrikelnummer: _____ Studiengruppe: _____

Hörsaal: _____ Platz Nr.: _____

Unterschrift: _____

Bitte kontrollieren Sie, ob Sie eine vollständige Angabe mit 5 Aufgaben auf 8 Seiten erhalten haben.

Aufgabe	1	2	3	4	5	Summe
max. Punkte	15	20	20	15	20	90

Anmerkungen:

- Sie müssen als Antworten **keine** kompletten Programme schreiben, sondern nur den explizit verlangten Teil eines Programms.
- Schreiben Sie die Lösungen in die dafür vorgesehenen Kästchen. Sollte Ihnen der Platz dabei nicht reichen, benutzen Sie die Rückseite **und vermerken Sie das im dazugehörigen Kästchen!**

Aufgabe 1 (15 Punkte)

Mit dem Programm MyTrip sollen Sie eine Reise planen können. Dazu werden alle Teilabschnitte mit der Länge in Kilometer und der erlaubten Geschwindigkeit in km/h eingegeben. Das Programm berechnet die Gesamtlänge in Kilometer und die Gesamtdauer bei jeweiliger Höchstgeschwindigkeit in **Minuten**. Wird bei der Eingabe die Länge 0 eingegeben, so wird die Eingabe beendet. Negative Eingaben der Länge sowie der Geschwindigkeit werden ignoriert. Wird eine 0 bei der Geschwindigkeit eingegeben wird dies auch ignoriert, aber das Programm nicht beendet.

Eine Beispielsitzung könnte wie folgt aussehen:

```
$ java MyTrip
Kilometer: 10
Erlaubte Geschwindigkeit (km/h): 60
Kilometer: 10
Erlaubte Geschwindigkeit (km/h): 100
Kilometer: 0
20 km in 16 Minuten
```

Vervollständigen Sie das folgende Programm MyTrip:

```
public class MyTrip {
    public static void main(String[] args) {
        java.util.Scanner in = new java.util.Scanner(System.in);
```

```
        System.out.printf("%d km in %d Minuten", km, (int) time);
    }
}
```

Aufgabe 2 (20 Punkte)

Implementieren Sie ein Programm das zwei ganze Zahlen auf der Kommandozeile bekommt. Die erste Zahl ist die Anzahl von Sternen pro Zeile und Spalte. Die zweite Zahl gibt an wieviele Leerzeichen zwischen jeweils zwei Sternen sein sollen.

Zur besseren Verständlichkeit werden bei den folgenden Beispielen statt der Leerzeichen Punkte ausgegeben:

```
$ java ZoomIntoStars 2 0
```

```
**
```

```
**
```

```
$ java ZoomIntoStars 2 1
```

```
*.*
```

```
.
```

```
*.*
```

```
$ java ZoomIntoStars 2 2
```

```
*..*
```

```
.
```

```
.
```

```
*..*
```

```
$ java ZoomIntoStars 3 2
```

```
*..*..*
```

```
.
```

```
.
```

```
*..*..*
```

```
.
```

```
.
```

```
*..*..*
```

```
$ java ZoomIntoStars 4 2
```

```
*..*..*..*
```

```
.
```

```
.
```

```
*..*..*..*
```

```
.
```

```
.
```

```
*..*..*..*
```

```
.
```

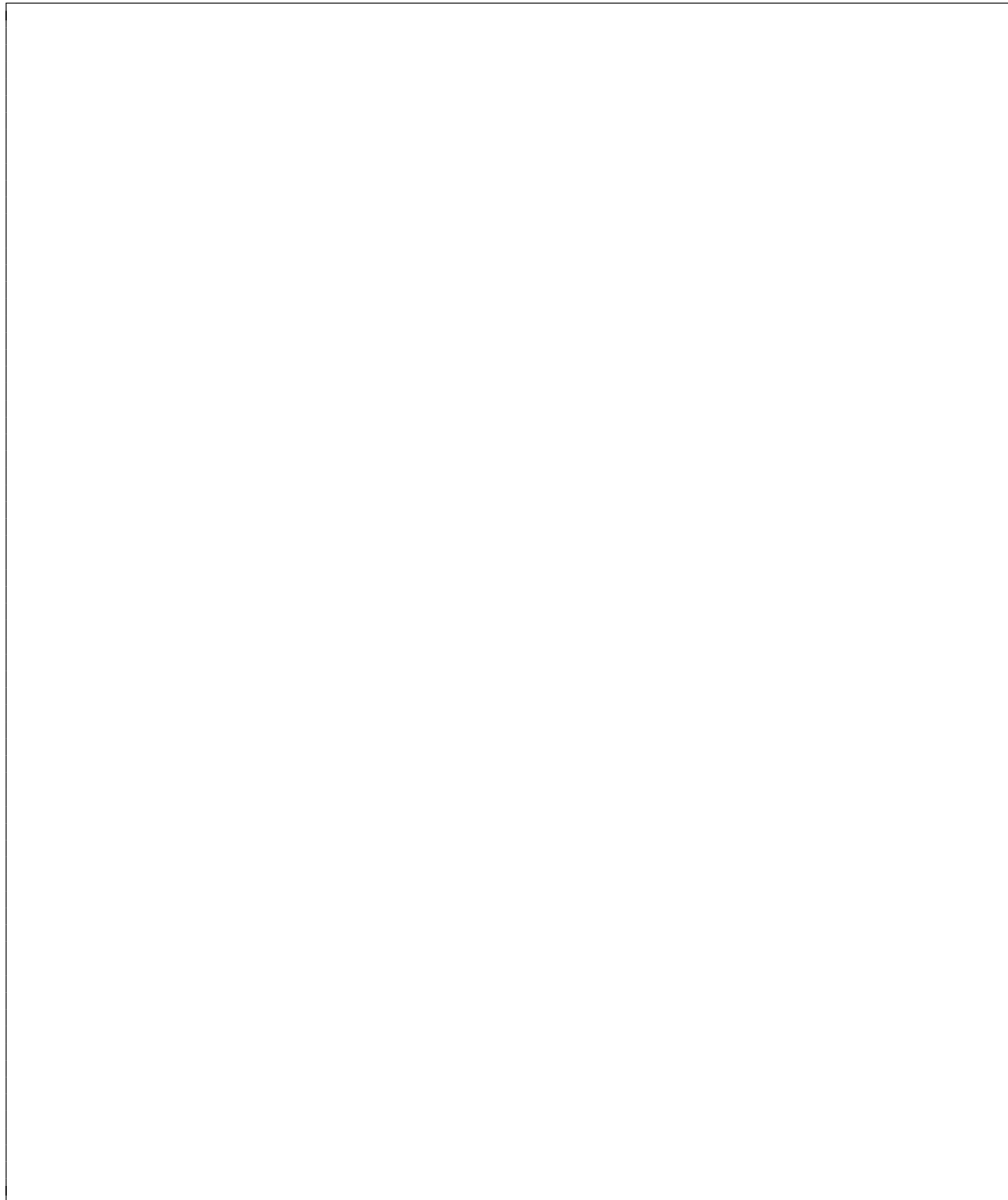
```
.
```

```
*..*..*..*
```

Für die Ausgabe dürfen Sie nur `System.out.print` mit einem einzelnen `char` als Argument und `System.out.println` ohne Argumente nutzen.

Vervollständigen Sie das folgende Programm `ZoomIntoStars`:

```
class ZoomIntoStars {  
    public static void main(String[] args) {  
        final int linesWithStars = Integer.parseInt(args[0]);  
        final int spacesBetweenStars = Integer.parseInt(args[1]);
```



```
    }  
}
```

Aufgabe 3 (20 Punkte)

Gegeben seien die Klassen A, B und App.

```
class A {
    int v = 23;
    void print() {
        System.out.println(this.v++);
    }
}
class B {
    int v;
    void print() {
        System.out.println(++v);
    }
}
class App {
    public static void main(String[] args) {
        int v = 5;
        A a = new A();
        B b = new B();
        b.v = 1;
        do {
            a.print();
            a.v = 13;
            a.print();
            if (a.v % b.v == 0) {
                b.v += 13;
            }
        } while (b.v % a.v >= 5);
        b.v += a.v++;
        a.print();
        b.print();
    }
}
```

Was wird beim Ausführen von App am Bildschirm ausgegeben?

Aufgabe 4 (15 Punkte)

Der folgende Code ist fehlerhaft und wird nicht kompiliert:

```
1 class Main {
2     public static void main(String[] args) {
3         int value = Integer.parse(args[0]);
4         int final result = 42;
5         while (int x = 0; x < 10; x += 2) {
6             switch (value)
7                 case 1:
8                     System.out.println("Oans");
9                     result += result;
10                case 2:
11                    System.out.printf("Ned");
12                    result = 1;
13                    break;
14                default;
15                    System.out.println("Wups");
16            }
17        }
18    }
19 }
```

Durch Änderung von 5 Zeilen in obigem Code, wird er kompilierbar. Geben Sie jeweils die Zeilennummer an und schreiben Sie dahinter wie die Zeile **korrekt** aussehen muss.

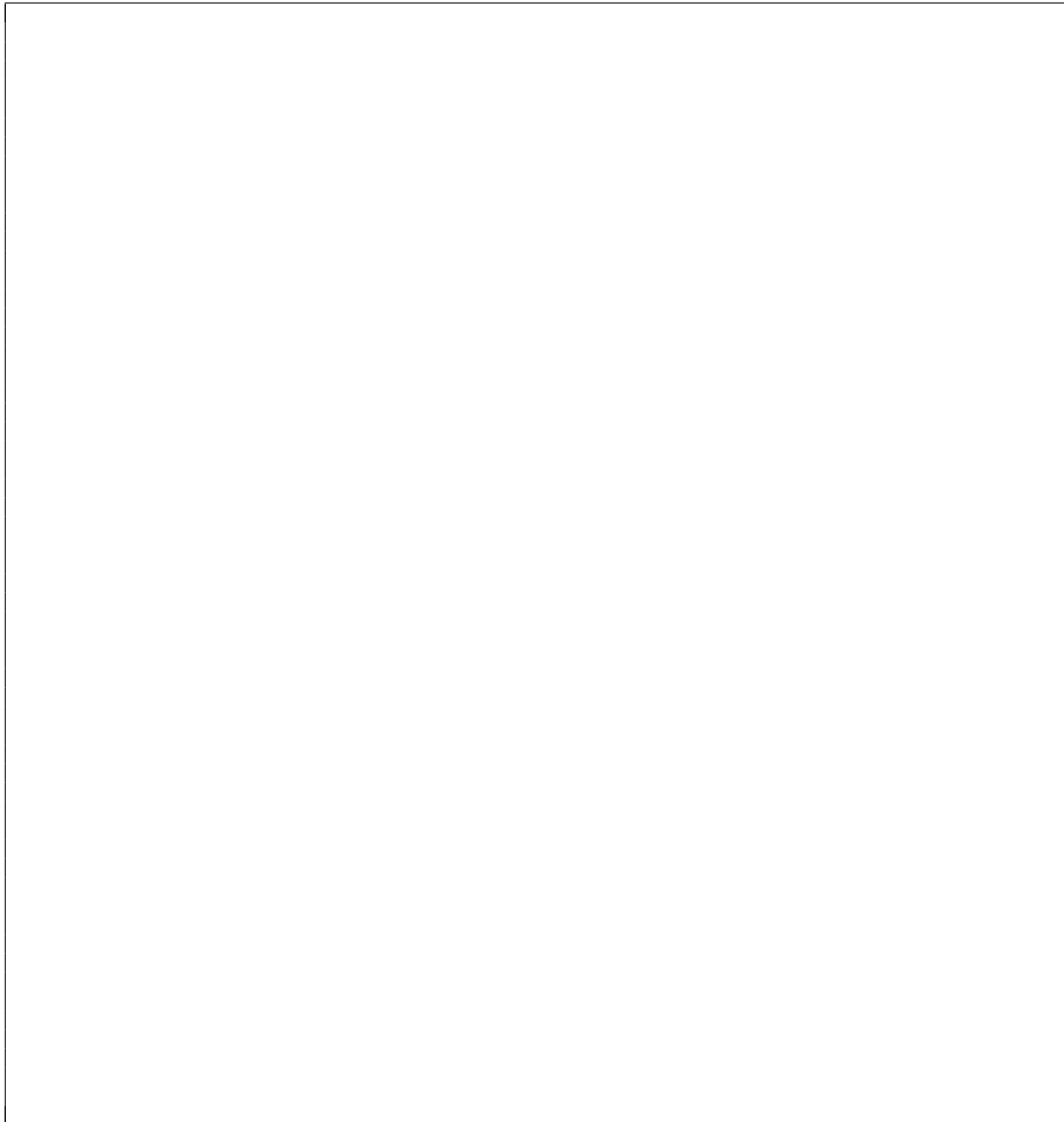
Aufgabe 5 (20 Punkte)

Gegeben sei folgende Java-Klasse `SkiSet` für Ski-Sets bestehend aus Ski, Bindung und Schuhen. Die Preise sind (sinnvollerweise) in Cent gespeichert.

```
class SkiSet {  
    String skiModel;  
    int skiPrice;  
    String bindingModel;  
    int bindingPrice;  
    String bootModel;  
    int bootPrice;  
}
```

- (a) Erzeugen Sie (wie innerhalb der `main`-Methode) ein Ski-Set bestehend aus folgenden Teilen: (5)

Ski: Head Jerry für € 600,-, Bindung: Marker KingPin für € 399,- und Schuhe: Movement Free Touring für € 299,-.



- (b) Erweitern Sie die Klasse um die Methode `printSetWithPrice`, die das Ski-Set inkl. aller Bestandteile, dem Set-Preis und der Ersparnis gegenüber der Summe der Einzelpreise, ausgibt. (15)

Berechnen Sie zunächst den Set-Preis wie folgt: Die Ski kosten den regulären Preis, die Bindung nur 90% und die Schuhe nur die Hälfte im Set. Berechnen Sie außerdem die Differenz zur Summe der Einzelpreise.

Gestalten Sie die Ausgabe so, dass beispielsweise das in Teilaufgabe a) erzeugte Set folgendermaßen ausgegeben wird:

Ski: Head Jerry (€ 600,0)

Bindung: Marker KingPin (€ 399,0)

Schuh: Movement Free Touring (€ 299,0)

Summe: € 1108,60 (€ 189,40 gespart)

