

Prüfung Softwareentwicklung I (IB)

Datum	:	25.01.2016, 10:30 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 7 Seiten erhalten haben.

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

Eine natürliche Zahl x heißt Primzahl genau dann wenn sie nur durch 1 und sich selbst teilbar ist.

Implementieren Sie ein Programm das alle Primzahlen zwischen 1 und der auf der Kommandozeile übergebenen Zahlen berechnet und diese zeilenweise ausgibt.

Vervollständigen Sie das folgende Programm `Primzahl`:

```
class Primzahl {  
    public static void main(String[] args) {  
        final int number = Integer.parseInt(args[0]);
```



```
    }  
}
```

Aufgabe 2 (15 Punkte)

Mit einem Scanner-Objekt können Sie Eingaben von der Kommandozeile einlesen. Eine ganze Zahl beispielsweise können Sie mit dem **Scanner** so einlesen und einer Variablen zuweisen:

```
int input = in.nextInt();
```

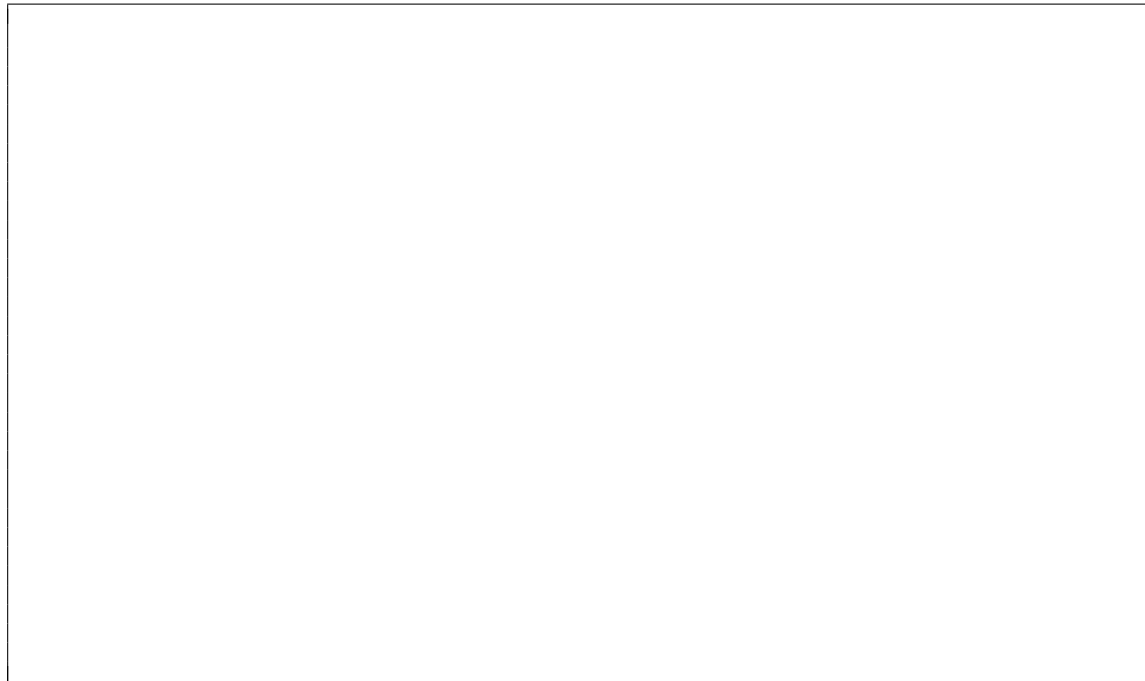
Mit einem Programm **GPA** (für grade point average) sollen Noten (nur ganze Zahlen zwischen 1 und 5) eingelesen werden. Gibt die BenutzerIn eine 0 ein, soll der Durchschnitt der bisher eingegebenen Noten berechnet und ausgegeben werden. Eine ungültige Eingabe (negative Note oder Note > 5) soll ignoriert werden.

Eine Beispielsitzung könnte wie folgt aussehen:

```
$ java GPA
Note: 2
Note: 2
Note: 3
Note: 0
Durchschnitt: 2.33
```

Vervollständigen Sie das folgende Programm **GPA**:

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



```
        System.out.printf("Durchschnitt: %.2f%n", gpa);
    }
}
```

Aufgabe 3 (15 Punkte)

Der folgende Code ist fehlerhaft und wird nicht kompiliert:

```
1  class Main {
2      public static void Main(String[] args) {
3          final int i = 0;
4          int j = 2;
5          if (i == 5) {
6              int j = i + 13;
7              System.out.print("%d == %d%n", i, j);
8          }
9          while (i > 13 // true) {
10             if (j == j) {
11                 System.out.printf("Hallo %s!%n", "Welt");
12                 breaking;
13             }
14             j--;
15         }
16     }
17 }
```

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.

Was wird vom korrigierten Programm alles ausgegeben?

Aufgabe 4 (20 Punkte)

Gegeben seien die Klassen A, B und App.

```
class A {
    int a = 7;
    void print() {
        a+=9;
        System.out.print(a);
    }
}
class B {
    final int b = 21;
    void print() {
        for (int i = 0; i < b ; i+=5) {
            if (b%7>4)
                System.out.print(i);
            System.out.println(i);
        }
    }
}
class App {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        a.print();
        b.print();
        a.print();
        a.print();
    }
}
```

Was wird beim Ausführen von App am Bildschirm ausgegeben?

\$ java App

Aufgabe 5 (20 Punkte)

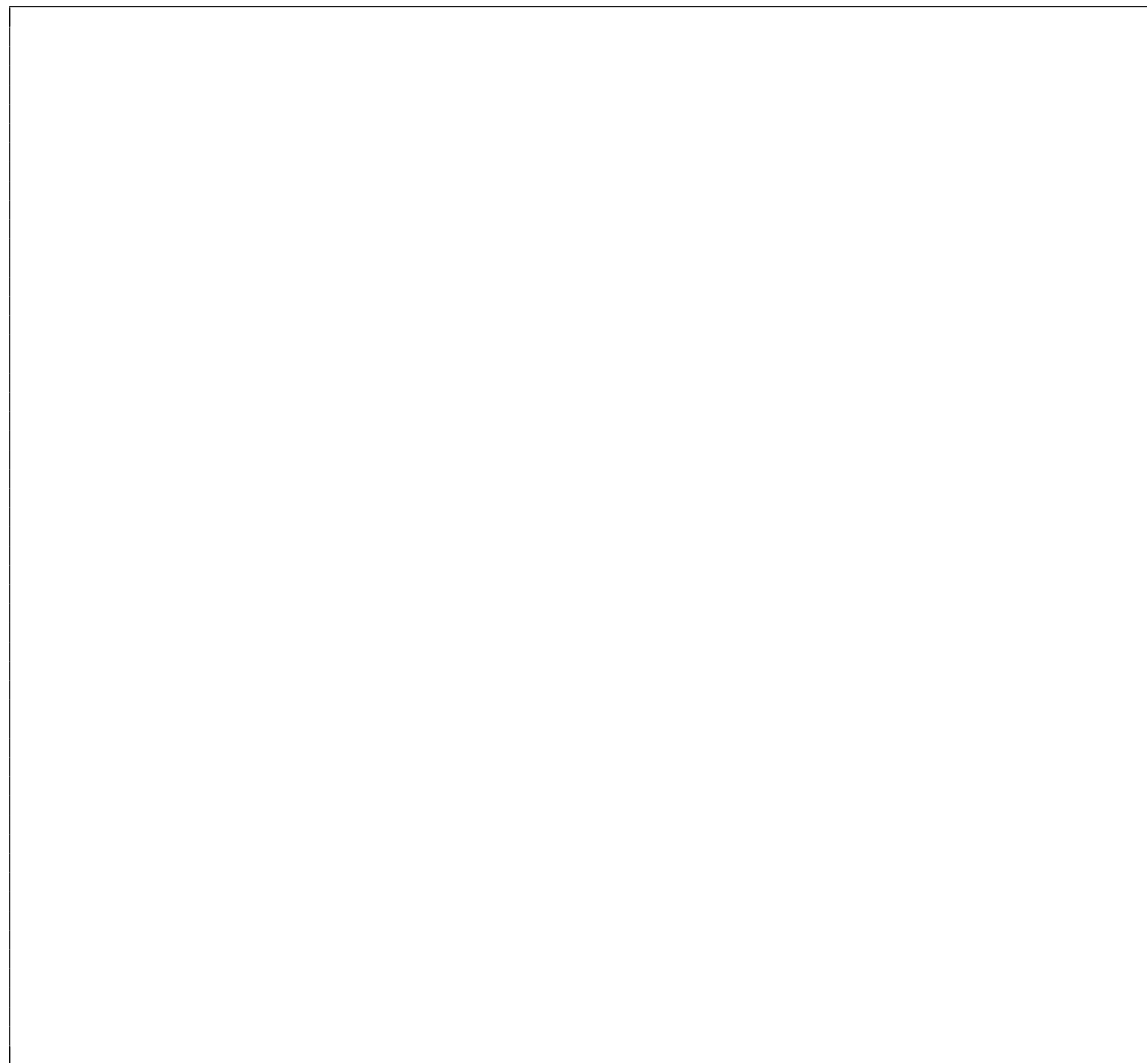
Implementieren Sie ein Programm das für die auf der Kommandozeile übergebene Anzahl von Zeilen ein gleichseitiges Dreieck ausgibt. Geben Sie statt der Leerzeichen links vom Dreieck einzelne Punkte aus, geben Sie für die Seiten Plussymbole und für die Fläche im Dreieck Minussymbole aus. Sie brauchen keine Leerzeichen ausgeben.

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

Beispielausgaben finden Sie auf der nächsten Seite.

Vervollständigen Sie das folgende Programm `Dreieck`:

```
class Dreieck {  
    public static void main(String [] args) {  
        final int lines = Integer.parseInt(args[0]);
```



```
    }  
}
```

Beispielausgaben für Dreiecke der Höhe 1 bis 7.

```
$ java Dreieck 1
+
$ java Dreieck 2
.+
+++
$ java Dreieck 3
..+
.+--+
+++++
$ java Dreieck 4
...+
..+--+
.+----+
+++++++
$ java Dreieck 5
....+
...+--+
..+----+
.+-----+
+++++++
$ java Dreieck 6
.....+
....+--+
...+----+
..+-----+
.+-----+
+++++++
$ java Dreieck 7
.....+
....+--+
...+----+
..+-----+
.+-----+
+++++++
```