



Informatik I – EProg HS13

Zwischentest I

Allgemeine Hinweise:

- Die Aufgabe muss vollständig bearbeitet werden, damit Sie die volle Punktzahl erhalten. Insgesamt können **60 Punkte** erreicht werden.
- Die Bearbeitungszeit beträgt **60 Minuten**.
- Unterlagen sind grundsätzlich **keine** erlaubt. Als Ausnahme gilt, falls Sie nicht deutscher Muttersprache sind, ein entsprechendes Wörterbuch.
- Schreiben Sie Ihren **Namen** und Ihre **Matrikelnummer** bitte in die dafür vorgesehenen Kästchen am Ende **jeder** Seite.
- Die Verwendung unerlaubter Hilfsmittel oder das Abschreiben von eine(r/m) Kommiliton(in / en) hat die sofortige Abgabe und das Nichtbestehen der Veranstaltung zur Folge. Zudem ist mit einem Disziplinarverfahren zu rechnen.
- Eine englische Version dieses Zwischentestes ist auf Nachfrage verfügbar. Bei Differenzen zwischen der deutschen und der englischen Version dieses Zwischentestes ist die deutsche Version massgebend.

Ich bestätige mit meiner Unterschrift, dass ich:

- die obenstehenden Hinweise gelesen und verstanden habe.
- die Prüfung unter zumutbaren Bedingungen geschrieben habe.

Unterschrift: _____ Datum: _____

Punkte
/60

1 Aufgabe: Code-Verständnis (4 Punkte)

```
1 public boolean mysteryMethod(long n) {  
2  
3     if (n == 1) {  
4         return false;  
5     }  
6  
7     if (n == 2) {  
8         return true;  
9     }  
10  
11    if (n % 2 == 0) {  
12        return false;  
13    }  
14  
15    for(int i = 3; i*i <= n; i++) {  
16        if(n % i == 0) {  
17            return false;  
18        }  
19    }  
20  
21    return true;  
22  
23 }
```

1. Welchen Wert liefert die Methode `mysteryMethod(long n)` zurück, wenn sie mit den folgenden Parametern aufgerufen wird? (3 Punkte)

3: _____

4: _____

17: _____

18: _____

54: _____

59: _____

2. Wie könnte man die Methode besser benennen? (1 Punkte)

Antwort: _____

Lösung:

Return-Wert der Methode:

3: true

4: false

17: true

18: false

54: false

59: true

(0.5 Punkte pro richtige Antwort)

Name der Methode: isPrime(long n) (1 Punkt)

2 Aufgabe: Variablen (8 Punkte)

Analysieren Sie das folgende Code Snippet. Kreuzen Sie alle richtigen Aussagen an. Beachten Sie, dass pro Teilaufgabe auch mehrere Antworten richtig sein können. Jede korrekt gelöste Teilaufgabe (alle Kreuze für diese Teilaufgabe richtig gesetzt) gibt 1 Punkt. Fehlerhafte Teilaufgaben geben 0 Punkte.

```
1 public class OOPTerms{
2     public int a = 10;
3     private String m = "abcdef";
4
5     public void c(int b) {
6         b = b + 5;
7         System.out.println(b);
8     }
9
10    private void d() {
11        a = a + 5;
12        System.out.println(a);
13    }
14
15    public void e() {
16        Student s = new Student();
17        int t = 8;
18    }
19
20    public int f;
21    public void f() {
22        m += m;
23    }
24 }
```

- | | | |
|---------------------|--|---|
| Zeile 2: a ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |
| Zeile 3: m ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |
| Zeile 7: b ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |
| Zeile 16: s ist ... | <input type="checkbox"/> eine Klasse | <input type="checkbox"/> ein Objekt |
| Zeile 16: s ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |
| Zeile 17: t ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |
| Zeile 20: f ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |
| Zeile 22: m ist ... | <input type="checkbox"/> eine Instanzvariable | <input type="checkbox"/> keine Instanzvariable |
| | <input type="checkbox"/> eine Variable eines primitiven Typs | <input type="checkbox"/> eine Variable eines Referenztyps |

Lösung:

- Zeile 2: a ist ... ☒ eine Instanzvariable ☐ keine Instanzvariable
☒ eine Variable eines primitiven Typs ☐ eine Variable eines Referenztyps
- Zeile 3: m ist ... ☒ eine Instanzvariable ☐ keine Instanzvariable
☐ eine Variable eines primitiven Typs ☒ eine Variable eines Referenztyps
- Zeile 7: b ist ... ☐ eine Instanzvariable ☒ keine Instanzvariable
☒ eine Variable eines primitiven Typs ☐ eine Variable eines Referenztyps
- Zeile 16: s ist ... ☐ eine Klasse ☒ ein Objekt
- Zeile 16: s ist ... ☐ eine Instanzvariable ☒ keine Instanzvariable
☐ eine Variable eines primitiven Typs ☒ eine Variable eines Referenztyps
- Zeile 17: t ist ... ☐ eine Instanzvariable ☒ keine Instanzvariable
☒ eine Variable eines primitiven Typs ☐ eine Variable eines Referenztyps
- Zeile 20: f ist ... ☒ eine Instanzvariable ☐ keine Instanzvariable
☒ eine Variable eines primitiven Typs ☐ eine Variable eines Referenztyps
- Zeile 22: m ist ... ☒ eine Instanzvariable ☐ keine Instanzvariable
☐ eine Variable eines primitiven Typs ☒ eine Variable eines Referenztyps

3 Aufgabe: String API (2 Punkte)

Welche Ausgabe erfolgt auf der Konsole wenn die Methode `analyseString` ausgeführt wird?

```
1 public void analyseString(){
2     String value = "We all love Java!".substring(3,6).concat("\n");
3     value.toUpperCase();
4     System.out.println(value);
5 }
```

- ☐ allN
- ☐ llN
- ☐ alln
- ☐ ALLN
- ☐ Keine der aufgeführten Aussagen trifft zu

Hinweis: Folgender Auszug aus der Java API zeigt, wie die Methode `substring(int beginIndex, int endIndex)` der Klasse `String` verwendet werden kann.

```
public String substring(int beginIndex,
                        int endIndex)
```

Returns a new string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex - 1`. Thus the length of the substring is `endIndex - beginIndex`.

Lösung:

Welche Ausgabe erfolgt auf der Konsole wenn die Methode `analyseString` ausgeführt wird?

- ☐ allN
- ☐ llN
- ☒ alln
- ☐ ALLN
- ☐ Keine der aufgeführten Aussagen trifft zu

4 Aufgabe: Typumwandlung (3 Punkte)

Was wird auf die Konsole ausgegeben, wenn das folgende Code Snippet in einer `main`-Methode ausgeführt wird?

```
1 int value = (int) 12.9 + (int) 3.1;  
2 System.out.println(value);
```

Lösung:

15

5 Aufgabe: Implementation (43 Punkte)

Aufgabenstellung

Der Schweizerische Sportverband möchte ab dem Jahr 2014 die Snowboard-Profis besser unterstützen. Dazu sollen Sie eine Simulation entwickeln, die die aktuelle Situation der Snowboard-Profis abbildet:

Snowboarder (`Snowboarder`) haben einen Namen (`name`) und sind bereits für eine bestimmte Anzahl Jahre (`years`) als Profi tätig. Jeder Snowboarder kann bis zu 10 Snowboards (`Snowboard`) lizenzieren lassen, um diese dann anschliessend an offiziellen Wettkämpfen verwenden zu können. Jedes Snowboard hat einen Spitznamen (`nickName`) und eine bestimmte Länge, die in Zentimetern angegeben wird (`boardLength`).

Wenn sich ein Snowboarder für einen Wettkampf vorbereitet, wählt er eine bestimmte Länge die sein Snowboard haben soll aus. Die Simulation soll die Möglichkeit bieten, den Namen und die Länge des Snowboards auf die Kommandozeile auszugeben, zwischen dessen Länge und der gewünschten Länge die kleinste Differenz besteht.

Bilden Sie den oben stehenden Sachverhalt auf Klassen, Methoden, Attribute und Objekte ab und befolgen Sie zusätzlich folgende Implementierungshinweise:

- Wenn ein Snowboarder ein neues Snowboard lizenzieren will, dann soll zuerst überprüft werden ob der Snowboarder bereits 10 Snowboards lizenziert hat. In diesem Fall soll eine entsprechende Meldung auf die Konsole ausgegeben werden. Andernfalls soll das neue Snowboard lizenziert werden.
- Die Werte von Instanzvariablen sollen nur über Methoden verändert und gelesen werden können.
- Wenn die Simulation das Snowboard ausgeben soll, das die kleinste Differenz zur gewünschten Länge aufweist, dann wird zuerst überprüft ob mindestens 1 Snowboard lizenziert ist. Andernfalls wird eine entsprechende Meldung auf die Konsole ausgegeben.
- Sie können davon ausgehen, dass jeweils nur 1 Snowboard die kleinste Differenz zur gewünschten Länge aufweist.

Testen Sie in einem `TestDriver` die folgenden Vorgänge:

1. Erstellen Sie einen Snowboarder mit dem Namen Iouri, der bereits seit 5 Jahren Profisportler ist.
2. Lassen sie Iouri 3 Snowboards mit den Längen 150cm, 170cm, 190cm und den Spitznamen "Shadow", "Lightning", und "Power" lizenzieren.
3. Lassen sie Iouri das Snowboard auswählen, das zu der Länge von 165cm die kleinste Differenz aufweist.

Hinweis: Folgender Auszug aus der Java API zeigt wie die statische Methode `abs(double a)` der Klasse `Math` benutzt werden kann:


```
public static double abs(double a)
```

Returns the absolute value of a `double` value. If the argument is not negative, the argument is returned. If the argument is negative, the negation of the argument is returned. Special cases:

- If the argument is positive zero or negative zero, the result is positive zero.
- If the argument is infinite, the result is positive infinity.
- If the argument is NaN, the result is NaN.

In other words, the result is the same as the value of the expression:

```
Double.longBitsToDouble((Double.doubleToLongBits(a)<<1)>>>1)
```

Parameters:

a - the argument whose absolute value is to be determined

Returns:

the absolute value of the argument.

Es sind verschiedene Lösungswege denkbar!

Ihre Lösung:

Lösung:

```
1 public class Snowboard {
2
3     private String nickName;
4
5     private double boardLength;
6
7     public void setNickName(String nickName) {
8         this.nickName = nickName;
9     }
10
11     public void setBoardLength(double boardLength) {
12         this.boardLength = boardLength;
13     }
14
15     public double getBoardLength() {
16         return boardLength;
17     }
18
19     public String getNickName() {
20         return nickName;
21     }
22
23 }
```

Listing 1: Snowboard.java

```

1 public class Snowboarder {
2
3     private String name;
4
5     private int years;
6
7     private Snowboard[] snowboards = new Snowboard[10];
8
9     private int numberOfBoards = 0;
10
11     public void setName(String name) {
12         this.name = name;
13     }
14
15     public void setYears(int years) {
16         this.years = years;
17     }
18
19     public void licenceSnowboard(Snowboard snowboard) {
20         if(numberOfBoards < 10) {
21             snowboards[numberOfBoards++] = snowboard;
22         } else {
23             System.out.println("There are already 10 snowboards licensed");
24         }
25     }
26
27     public void getSnowboard(double length) {
28         if (numberOfBoards > 0) {
29             Snowboard bestBoard = snowboards[0];
30             for (int i = 1; i < numberOfBoards; i++) {
31                 if (Math.abs(length - bestBoard.getBoardLength()) > Math.abs
32                     (length - snowboards[i].getBoardLength())) {
33                     bestBoard = snowboards[i];
34                 }
35             }
36             System.out.println("Name: " + bestBoard.getNickName() + " /
37                 Length: " + bestBoard.getBoardLength());
38         } else {
39             System.out.println("There are no snowboards licensed");
40         }
41     }
42 }

```

Listing 2: Snowboarder.java

```

1 public class TestDriver {
2
3     public static void main(String[] args) {
4
5         Snowboarder iouri = new Snowboarder();
6         iouri.setName("Iouri");
7         iouri.setYears(5);
8
9         Snowboard shadow = new Snowboard();
10        shadow.setNickName("Shadow");
11        shadow.setBoardLength(150);
12
13        Snowboard lightning = new Snowboard();
14        lightning.setNickName("Lightning");
15        lightning.setBoardLength(170);
16
17        Snowboard power = new Snowboard();
18        power.setNickName("Power");
19        power.setBoardLength(190);
20
21        iouri.licenceSnowboard(shadow);
22        iouri.licenceSnowboard(lightning);
23        iouri.licenceSnowboard(power);
24
25        iouri.getSnowboard(165);
26
27    }
28
29 }

```

Listing 3: TestDriver.java

Hier steht zusätzlicher Raum für Ihre Lösung zur Verfügung.

Hier steht zusätzlicher Raum für Ihre Lösung zur Verfügung.