

Softwareentwicklung I (IB)

Blatt 1

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 28.01.2020 17:18

Abgabe aller Aufgaben auf diesem Blatt: bis 01.11.17 10:00 Uhr durch Pushen in das zur Aufgabe gehörende GitHub-Repository.

Bewertung: Eine Aufgabe ist korrekt gelöst, wenn

1. Sie **fristgerecht** den Code in das zur Aufgabe gehörende GitHub-Repository gepusht haben und
2. der zugehörige Jenkins-Job **bei Fristende** grün ist und
3. Sie bis spätestens eine Woche nach Fristende **kein** Feedback erhalten haben, dass etwas **falsch** ist.

Diese Bewertungskriterien gelten für alle Blätter, außer auf einem Blatt oder bei einer Aufgabe ist explizit etwas anderes angegeben.

Aufgabe 1

Das Repository für diese Aufgabe bekommen Sie unter <https://classroom.github.com/a/OCKnUuyA>.

In diesem Repository finden Sie eine Vielzahl von Dateien.

So ähnlich finden Sie das in Zukunft in jedem Repository für die Aufgaben. Wenn etwas neues hinzu kommt, werde ich das dann erklären.

Nun zu den einzelnen Dateien:

- `BigHello.java`: In dieser Datei sollen Sie Ihren Code bearbeiten. Alle anderen Dateien lassen Sie unverändert.

- **Jenkinsfile:** Ihre Abgaben werden automatisch auf meinem [Jenkins-Server](#) überprüft. Dabei wird genau das gemacht was in dieser Datei steht.
- **checkstyle-8.3-all.jar** und **checkstyle.xml:** [Checkstyle](#) ist ein Programm mit dem Java-Code nach einem vorgegebenen Stil überprüft wird. Die **.jar**-Datei enthält das Programm, die **.xml**-Datei enthält den vorgegebenen Stil. Sie entspricht in dieser Version dem [Google Java Style Guide](#) nachdem Sie sich richten sollen, außer ich gebe explizit etwas anderes vor. Sie können die Checkstyle-Überprüfung mit Hilfe dieser Dateien auch selbst auf der Kommandozeile mit der folgenden Eingabe durchführen:

```
java -jar checkstyle-8.3-all.jar -c checkstyle.xml BigHello.java
```

Exakt diesen Aufruf finden Sie auch in der Datei **Jenkinsfile**.

- **expected.txt:** Diese Datei enthält die erwartete Ausgabe aus der Aufgabenstellung. Der Jenkins vergleicht Ihre Ausgabe mit dieser Datei.

Jetzt zur eigentlichen Aufgabe:

In der Vorlesung haben Sie das HelloWorld-Programm kennen gelernt.

```
class Hello {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Schreiben Sie ausgehend von diesem Programm ein neues Programm, das beim Aufruf “Hello, World!” folgendermaßen ausgibt:

```
| | | | _ _ _ | | | _ _ _ | \ \ / / _ _ _ _ _ | | _ _ | |
| | _ | / _ \ | | / _ \ \ \ / / / _ \ | ' _ _ | | / _ \ | |
| _ | | _ / | | ( ) | \ v v / ( ) | | | | ( _ | _ |
| _ | | _ \ _ _ | | \ _ _ ( ) \ _ \ / \ _ _ / | | | _ \ _ _ ( )
| /
```

Anmerkung: In Ihrem Repository finden Sie die Datei **expected.txt**, die den Text enthält.

Allerdings gibt es noch ein kleines Problem: Im oben dargestellten “Hello, World!” tauchen einige Backslashes auf. Diese haben innerhalb einer Zeichenkette jedoch eine besondere Bedeutung. Sie bilden mit dem unmittelbar darauf folgenden Zeichen ein sogenanntes **Steuerzeichen**. Beispielsweise wird bei

```
System.out.println("\n");
```

kein Backslash und ein `n` ausgegeben, sondern die beiden Zeichen stehen zusammen für einen Zeilenumbruch. D.h. die Anweisung

```
System.out.println("123\n456");
```

gibt

```
123
```

```
456
```

aus. Es wird also durch Einfügen des Steuerzeichens bei der Ausgabe in eine neue Zeile gewechselt.

Um nun das Zeichen Backslash aber tatsächlich ausgeben zu können, muss es mit einem vorangestellten Backslash entwertet werden. Das heißt also beispielsweise die dritte Zeile können Sie nicht mit

```
System.out.println("| |_| |/_ \ | |/_ \ \ \ / \ / / _ \ | ' _ | |/_ ` | |");
```

sondern nur mit

```
System.out.println("| |_| |/_ \\ | |/_ \\ \\ \\ / \\ / / _ \\ | ' _ | |/_ ` | |");
```

korrekt ausgeben. Statt einem Backslash stehen also immer gleich zwei davon hintereinander.

Verwenden Sie die Datei `BigHello.java` und implementieren Sie darin die Klasse `BigHello`.

Aufgabe 2

Das Repository für diese Aufgabe bekommen Sie unter <https://classroom.github.com/a/Lkepttwu>. Im Repository finden Sie das fehlerhafte Java-Programm in der Datei `UserInfo`.

Finden und verbessern Sie die drei Fehler.

Beseitigen Sie außerdem alle Fehler die Checkstyle Ihnen anzeigt:

```
java -jar checkstyle-8.3-all.jar -c checkstyle.xml UserInfo.java
```

Checkstyle kann Ihren Code übrigens erst überprüfen, wenn er kompiliert werden kann.

Anmerkung: Die in der Vorlesung noch nicht behandelten Funktionsaufrufe

```
System.getProperty("user.name")
System.getProperty("user.home")
```

sind korrekt und zeigen Ihren Benutzernamen und Ihr sog. Home-Verzeichnis auf dem von Ihnen gerade genutzten Rechner an.

Die Ausgabe auf meinem Rechner ist z.B.

```
Benutzername: obraun
Home-Verzeichnis: /Users/obraun
```

Die Ausgabe auf dem Jenkins lautet

```
Benutzername: jenkins
Home-Verzeichnis: /usr/local/jenkins
```

Aufgabe 3

Das Repository für diese Aufgabe bekommen Sie unter <https://classroom.github.com/a/XHnP6EiT>.

Geben Sie mit Hilfe eines Java-Programms Bill folgende Zeilen auf dem Bildschirm aus:

```
Position      Preis
=====
Regenschirm   5
Muetze        25
Schal         15
              =====
              45
```

Beachten Sie bei Ihrer Lösung folgende Randbedingungen:

- Sie dürfen in Ihrem gesamten Programm nur einen einzigen Aufruf der Methode `System.out.print` verwenden. Die Verwendung von `System.out.println` und `System.out.printf` ist verboten!
- In keiner Zeichenkette darf eine Zahl vorkommen.
- Außerhalb von Zeichenketten dürfen nur die drei Zahlen 5, 15 und 25 vorkommen. Diese aber beliebig oft.
- Es wird **nicht** erwartet, dass Sie in Ihrem Code Variablen nutzen, da wir diese noch nicht in der Vorlesung behandelt haben.

Auch diese Aufgabe ist natürlich erst wieder bestanden, wenn Jenkins inklusive Checkstyle zufrieden ist. Ob Sie die Randbedingungen eingehalten haben, überprüft der Jenkins nicht automatisch. Dies wird von uns direkt überprüft und muss zum Bestehen natürlich auch in Ordnung sein.