

Abgabe: 26.10.2015 (vor 5:00 Uhr)

Hinweis: In diesem Übungsblatt wird vorausgesetzt, dass Sie *NetBeans* auf Ihrem Computer installiert haben.

Aufgabe 1.1 (P) Umfrage

Um die Vorlesung und das Praktikum noch gezielter auf die Vorkenntnisse der Studierenden abzustimmen, möchten wir Sie bitten an folgender Umfrage teilzunehmen. Dabei erfragen wir einige statistische Daten und erfassen eventuell schon vorhandene Kenntnisse in der objektorientierten Programmierung. Die Teilnahme ist freiwillig und dauert ca. 30 Minuten.

<http://www.ddi.edu.tum.de/umfrage-info1/>

Anmerkung: Für ein erfolgreiches Informatik-Studium sind keine Vorkenntnisse in objektorientierter Programmierung notwendig! Es ist also kein Problem, wenn Sie bei diesen Fragen die richtigen Lösungen nicht wissen. Die Erhebung dient auch genau dazu, diesen Wissensstand festzustellen, um darauf gezielt eingehen zu können.

Aufgabe 1.2 (P) Java auf der Kommandozeile

Eine Entwicklungsumgebung zum Programmieren (wie z.B. *NetBeans*) ist im Grunde nur ein besserer Text-Editor, welcher hilfreiche Funktionalität bietet (wie etwa das Arbeiten im *Debug-Modus*, um Fehler zu finden, oder das Hervorheben von Java-Schlüsselwörtern, um die Struktur des Java-Programms zu verdeutlichen). Unter dieser Oberfläche verbirgt sich im Wesentlichen ein *Compiler*, welcher Ihr Java-Programm in Computer-Befehle übersetzt, damit der Computer es ausführen kann. Per Kommandozeile ist es möglich, direkt den Compiler aufzurufen und seine Dienste in Anspruch zu nehmen. Mit dem Aufruf

```
javac Programm.java
```

lässt sich der Quellcode, der in der Datei `Programm.java` steht, in eine ausführbare Form bringen, den Java-Bytecode, welcher dann in die Datei `Programm.class` geschrieben wird. Beim Übersetzungsvorgang wird zugleich geprüft, ob der Quellcode syntaktisch korrekt ist, d.h. ob er der Grammatik der Sprache Java entspricht. Ist dies nicht der Fall, so findet keine Übersetzung statt, sondern es werden Fehlermeldungen ausgegeben.

Um den Java-Bytecode aus der Datei `Programm.class` auszuführen, ruft man nun einfach

```
java Programm
```

auf der Kommandozeile auf.

Schreiben Sie den folgenden Text mit Hilfe eines Text-Editors (z.B. *Editor*, *Notepad(++)*, *Kate*, ...) und speichern Sie ihn als eine Text-Datei namens `MeinProgramm.java` ab:

```
public class MeinProgramm {  
    public static void main (String args[]) {  
        System.out.print("Hell");  
        System.out.print("o Worl");  
        System.out.println("d");  
    }  
}
```

Wechseln Sie nun auf der Kommandozeile mit dem Befehl `cd` in das Verzeichnis, in welchem sich Ihre Datei befindet, und geben Sie die Befehle zum Übersetzen und Ausführen Ihres Programms ein.

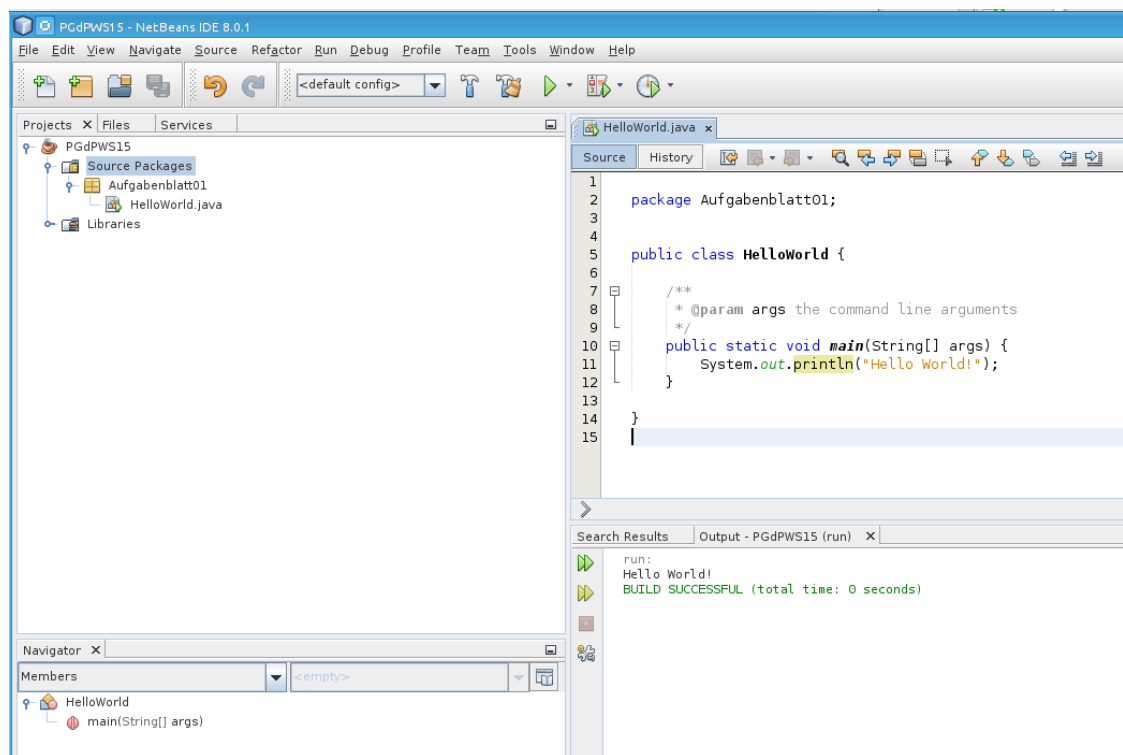
Hinweis: Evtl. müssen Sie die PATH-Systemvariable ändern. Eine Anleitung finden Sie dazu unter <http://www.java.com/de/download/help/path.xml>

Aufgabe 1.3 (P) Programmieren mit *NetBeans*

- a) Starten Sie die Entwicklungsumgebung *NetBeans* und erstellen Sie ein neues Java-Projekt namens PGdPWS15. Erstellen Sie nun die Klasse `HelloWorld` im Package `Aufgabenblatt01`. Diskutieren Sie mit Ihrem Tutor die automatisch erstellten Elemente des Programms. Ergänzen Sie Ihr Programm folgendermaßen:

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

Am Ende sollte Ihr *NetBeans* in etwa so aussehen:



Machen Sie sich mit den grundlegenden Funktionen wie dem Erstellen und Ausführen des Programms vertraut.

- b) Machen Sie den Ordner in Ihrem Benutzerverzeichnis ausfindig, in dem *NetBeans* Ihre Java-Dateien ablegt. Normalerweise finden Sie diese Dateien unter Unix im Ordner

`/home/username/NetBeansProjects/PGdPWS15/src`

und unter Windows im Ordner

`%HOMEPATH%\Eigene Dokumente\NetBeansProjects\PGdPWS15\src`

Üben Sie nun die Abgabe Ihrer Hausaufgaben auf Moodle, indem Sie Ihr HelloWorld-Programm auf der Moodle-Seite¹ als Lösung des ersten Aufgabenblattes abgeben (und anschließend wieder löschen).

Aufgabe 1.4 (P) Groß- und Kleinschreibung

In einer gegebenen Zeichenkette sollen alle Kleinbuchstaben (a bis z) in die entsprechenden Großbuchstaben umgewandelt werden. Legen Sie hierfür zunächst im Programm einen String `s` an, den Sie umwandeln wollen, und verwenden Sie zum Lösen der Aufgabe eine Java-Bibliotheksfunktion, mit der man Zeichen in einem String ersetzen kann.

Erweitern Sie Ihr Programm nun so, dass der ursprüngliche String je einmal mit ausschließlich Kleinbuchstaben und einmal mit ausschließlich Großbuchstaben ausgegeben wird. Beispiel: Die Zeichenkette "Hallo Welt!" liefert die beiden Ausgaben "HALLO WELT!" und "hallo welt!".

Hinweis: Sehen Sie z.B. unter <http://docs.oracle.com/javase/8/docs/api/java/lang/String.html> nach.

Aufgabe 1.5 [3 Punkte] (H) Syntaktische Fehler

Finden und beheben Sie die fünf syntaktischen Fehler im folgenden Java-Quellcode, und beschreiben Sie anschließend, auf welche Weise das Programm geändert werden muss, damit bei der Ausführung die vom Programmierer beabsichtigten acht Zeilen ausgegeben werden.

```
public class DieErlkoenigin {
    public static void main(String[] arg) {
//Zeile
/* 1. */ System.out.println("Wer reitet so spät");
/* 2. */ System.out.println("durch Nacht und Wind?");
/* 3. */ System.out.println("Es ist " "die Mutti");
/* 4. */ System.out.println("mit ihrem Kind.");
/* 5.   System.out.println("Sie hat den Knaben");
   6. */ System.out.println("wohl in dem Arm,"),
/* 7. */ System.out.println ( "Sie fasst ihn sicher,");
// 8.   System.out.println("sie hält ihn warm.");
}
```

Aufgabe 1.6 [3 Punkte] (H) Stringtausch

In dieser Aufgabe sollen die Werte von zwei bzw. drei String-Variablen getauscht werden. Initialisieren Sie dazu die Variablen `varA`, `varB`, `varC` mit den Werten "eins", "zwei", "drei". Tauschen Sie nun die Werte der Variablen `varA` und `varB`. Verwenden Sie dazu lediglich *eine* weitere Hilfsvariable des Typs `String`. Tauschen Sie anschließend wieder nur unter Verwendung der Hilfsvariablen des Typs `String` die Variablenwerte wie folgt: die Variable `varA` bekommt den Wert der Variablen `varC`, die Variable `varB` bekommt den Wert der Variablen `varA` und die Variable `varC` bekommt den Wert der Variablen `varB`.

Geben Sie die Variablen und deren Werte nach der Initialisierung, nach dem ersten Tausch und nach dem zweiten Tausch jeweils aus. Die Ausgabe sollte dann so aussehen:

```
varA = eins
varB = zwei
varC = drei
```

¹ <https://www.moodle.tum.de/course/view.php?id=22916>
bzw. <https://www.moodle.tum.de/course/view.php?id=25200> (Gruppen 51-58), wie im Merkblatt beschrieben

```

varA = zwei
varB = eins
varC = drei

varA = drei
varB = zwei
varC = eins

```

Nach der Initialisierung dürfen Sie lediglich Zuweisungen von Variablen an Variablen verwenden, also zum Beispiel `varA = tmp` oder `varA = varB`, wobei `tmp` die Hilfsvariable ist, jedoch keine Zuweisungen der Form `varA = "zwei"`.

Aufgabe 1.7 [3 Punkte] (H) Vokale ersetzen

Schreiben Sie ein Programm `Vokalersetzung`, das im unten stehenden Text alle Vokale durch einen vom Benutzer eingegebenen Vokal ersetzt. Speichern Sie dazu den angegebenen Text inklusive Zeilenumbrüchen in einer `String`-Variablen ab. Achten Sie darauf, dass die Groß- und Kleinschreibung nach der Ersetzung mit der vor der Ersetzung übereinstimmt. Umlaute (Ä/ä, Ö/ö, usw.) sollen dabei nicht als Vokale betrachtet werden. Beispiel: Das Wort "`Exenmeister`" wird bei Eingabe von O oder o zu "`Oxonmoostor`".

```

Hat der alte Hexenmeister
sich doch einmal wegbegeben!
Und nun sollen seine Geister
auch nach meinem Willen leben.
Seine Wort und Werke
merkt ich und den Brauch,
und mit Geistesstärke
tu ich Wunder auch.
Walle! walle
Manche Strecke,
daß, zum Zwecke,
Wasser fließe
und mit reichem, vollem Schwalle
zu dem Bade sich ergieße.

```