Hochschule der Medien Stuttgart

Studiengang Audiovisuelle Medien

Übungen Informatik - 1

Aufgabensammlung

Prof. Uwe Schulz / Prof. Simon Wiest

Sommersemester 2020

Organisation der Informatik Übungen

Die Note für das Modul Informatik 1 wird wie folgt ermittelt:

- Im Laufe des Semesters geben Sie jede Woche eine Aufgabe ab, die Abgabefrist endet jeweils am Sonntag um 24 Uhr. Für jede **fristgerecht** eingereichte Aufgabe erhalte Sie einen oder zwei Punkte, maximal 20 Punkte
- Eine Abgabe besteht aus einem selbst geschriebenen Java Programm und eventuell einem kurzen Test mit Fragen zum Stoff. Die Punkte erhalten Sie unabhängig vom Ergebnis des Tests. Bei den Java Programmieraufgaben mache ich Stichproben, ob es sich um eine Lösung der gestellten Aufgabe handelt, auch hier muss die Lösung nicht perfekt sein.
- Zweimal im Semester werden Sie zu einem Interview aufgefordert, bei dem ich Ihnen zu der zuletzt abgegebenen Aufgabe und zum bisherigen Stoff Fragen stelle. Für jedes Interview gibt es bis zu 15 Punkte, also maximal 30 Punkte.
- Nach Ende der Vorlesungen ist eine schriftliche Klausur geplant, die mit maximal 50 Punkten bewertet wird, also die Hälfte der Note ausmacht. Falls im August aufgrund der Corona-Krise keine Klausur möglich ist, werde ich mit jedem nochmals ein ausführliches Interview durchführen.
- Die Informatik Note wird ergibt sich aus den erreichten Punkten von maximal 100 Punkten.
- **Wenn Sie an der Klausur nicht teilnehmen, gilt das Fach als nicht bestanden, falls Sie kein ärztliches Attest eingereicht haben.** Bei Vorlage eines ärztlichen Attests ist ein Nachtermin nach Vereinbarung möglich. Die Interviews sind Prüfungsleistungen, es sind keine Hilfsmittel erlaubt.

Bitte vergessen Sie nicht, Informatik 1 als Prüfungsleistung anzumelden!

Installation der Software für die Übungen auf dem eigenen Rechner

Sie benötigen folgende Programme, um die Übungen am eigenen Rechner durchzuführen (u.A. aus lizenzrechtlichen Gründen arbeiten wir mit **Java Version 8**):

Java Virtual Machine (JVM): Unter www.java.com können Sie kostenlos das Java Runtime Environment (JRE) herunterladen, die darin enthaltene JVM ermöglicht die Ausführung von Java Programmen. Auf vielen Rechnern ist Java bereits installiert.

Eclipse Entwicklungsumgebung: www.eclipse.org, unter „*Download*" klicken Sie auf „*Download 64 bit*" unterhalb von „*Get Eclipse IDE …*".
Nach dem Start des Installers wählen Sie „*Eclipse IDE for Java Developers*",
Sie brauchen NICHT die viel umfangreichere Version für Java EE Entwickler.

Falls der Installer meldet „*The required 63-bit Java 1.8 vrtual machine could not be found*" kommt, muss erst eine JVM installiert werden (siehe oben).

Web-Resourcen

API Dokumentation:

Die wichtigste Dokumentation ist die Beschreibung der Standard-Klassen mit ihren Methoden (API Documentation**).** Wenn Sie online arbeiten, finden Sie sie hier: http://docs.oracle.com/javase/8/docs/api/

Oder Sie können sich die Verzeichnisse mit den HTML-Dateien lokal installieren: docs.oracle.com/javase/8/docs/, dort unter ‚Downloads' die Datei *jdk-8-apidocs.zip* herunterladen und entpacken, danach die Datei *index.html* in einem Browser öffnen und ein Bookmark darauf setzen.

Java ist auch eine Insel: openbook.galileocomputing.de/javainsel/

Hier finden Sie ein weiteres Java Buch, das Sie direkt online lesen können. Alles was wir in den ersten 2 Semestern behandeln (und noch viel mehr) finden Sie hier ausführlich beschrieben.

Weitere Quellen zur Java Programmierung:

Das Handbuch zu Java 8, Guido Krügers und Heiko Hansen: dieses Buch ist ein guter Einstieg in Java.

für alle die lieber mit Videos lernen: auf ***youtube*** finden Sie viele Kurse zu Java, auf ***www.udemy.com*** gibt es ebenfalls mehrere gute Einführungskurse.

Bitte lösen Sie Aufgaben 1-3 bis zum Vorlesungsbeginn, spätestens zum 22.04.2020

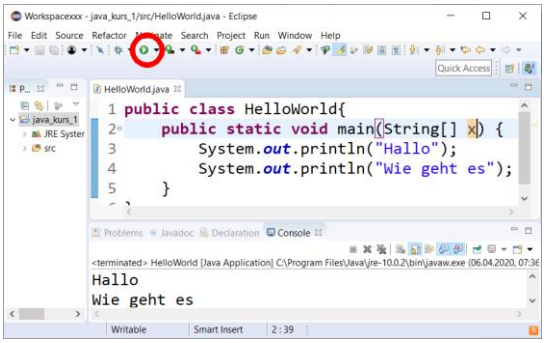
Aufgabe 1

Lernen Sie nebenstehendes Java Programm auswendig!
Häufige Fehler sind:

- fehlendes Semikolon,
- falsche Klammer, z. B. **(** statt **{** .

```
public class Aufgabe1 {
   public static void  main (String[] x) {
         System.out.println(”Hallo”);
         System.out.println (”wie geht es”);
   }
}
```

1. Starten Sie Eclipse und klicken Sie die „*Welcome to Eclipse“*-Seite weg.
2. Legen Sie ein neues Java-Projekt an: auf ***File->New->Project*** klicken und dort ***Java Project*** auswählen. Wichtig: **keinen anderen Projekt-Typ nehmen**!
   Nennen Sie das Projekt z.B. **java_kurs_1**.
3. Erzeugen Sie die Klasse *Aufgabe1*, durch Klick auf ***File->New->Class*** und geben Sie als Name ***Aufgabe1*** ein (Klassen beginnen mit Großbuchstaben) und tippen obiges Programm ein.
4. Speichern Sie das Programm (***File->Save*** oder die Tastenkombination ***Strg-s***). Dabei wird es übersetzt und eventuelle Fehler werden im unteren Fenster von Eclipse unter *Problems* angezeigt. Fehler werden auch als Tooltip angezeigt, wenn man mit der Maus darüber fährt.
5. Um das Programm auszuführen, klicken Sie entweder auf die Schaltfläche mit dem weißen Dreieck im grünen Kreis (im Bild rot markiert) oder Sie rufen mit Rechtsklick das Kontextmenü auf und wählen „*Run as → Java Application*“.
   Im Fenster mit dem Titel *Console* erscheint die Ausgabe Ihres Programmes:
   Hallo
   wie geht es
   Gratulation: Sie haben soeben Ihr erstes Java-Programm geschrieben und ausgeführt!



6. Ein weiterer Klick auf die Schaltfläche mit dem weißen Dreieck im grünen Kreis staret das zuletzt ausgeführte Programm noch einmal.
7. Öffnen Sie die Java Dokumentation in einem Web Browser (docs.oracle.com/javase/8/docs/api). Diese Dokumentation können Sie sich auch wie oben beschrieben auf Ihren Rechner herunterladen und offline lesen
8. *System.out* ist ein Objekt, *System.out.println(…)* ruft eine Methode dieses Objekts. Versuchen Sie herauszufinden, zu welcher Klasse die Methoden *print() und println()* gehören.

Aufgabe 2

Finden Sie die Fehler in untenstehendem Java Programm (Hinweis: auf fehlende Semikolon und Anführungszeichen reagiert der Compiler besonders heftig…).
Finden Sie heraus (durch Probieren), was die Bedeutung der Zeichenfolge \n ist.
Schauen Sie in der Java API Dokumentation(http://docs.oracle.com/javase/8/docs/api/) nach, was der Unterschied zwischen den Methoden ***println()*** und ***print()*** ist.

```
public class Aufgabe2 {                    // Tipp:  geben Sie main ein, dann
   public static void  main (String x ) {  // STRG sowie Leertaste, Eclipse
         System.out.println(Hallo);        // fügt die main-Methode ein.
         System.out.print (”wie \ngeht \nes”)
         String tier = "Die Kuh";
         System.out.println(tier + " ist uebern Fence gejumpt\n");
         System.out.print("und hat dabei den Benz gerammt\n");
   }
}
```

Um das Programm zu starten, führen Sie die Schritte 5 aus Aufgabe 1 erneut durch. Falls das Programm nicht startet, schauen Sie noch mal ganz genau die Zeile *public static void main …* an
Lesen Sie die Einleitung zur Klasse *String* in der Java API Dokumentation aufmerksam durch.
Was bewirkt der Operator **+** bei Strings?

Aufgabe 3: grafische Oberfläche, JFrame

Programme, die nur Text ausgeben sind unzeitgemäß und Langweilig. Unser nächstes Programm hat deshalb schon eine grafische Oberfläche. Das ist anspruchsvoller, macht aber mehr Spaß.

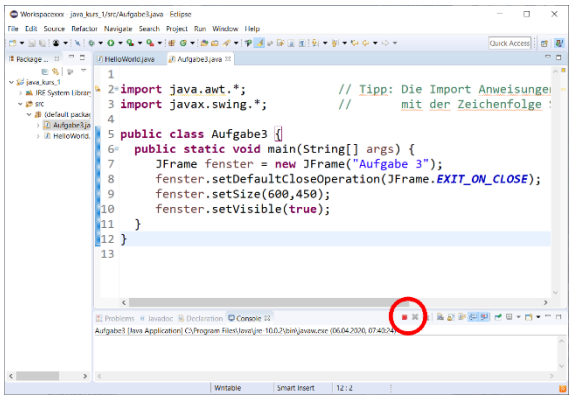
Geben Sie folgendes Programm ein, übersetzen Sie es und führen sie es aus:

```
import java.awt.*;              // Tipp: Import Anweisungen können Sie auch automatisch mit der
import javax.swing.*;           // Zeichenfolge STR+SHFT+O  einfügen lassen, dabei werden alle
                                // im Programm vorkommenden Klassen berücksichtigt
public class Aufgabe3 {
  public static void main(String[] args) {
     JFrame fenster = new JFrame("Aufgabe 3");
     fenster.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     fenster.setSize(600,450);
     fenster.setVisible(true);
  }
}
```

Dieses Programm erzeugt ein Objekt vom Typ *JFrame*. Dann werden einige Eigenschaften dieses Objekts festgelegt. Was das genau bedeutet, wird erst nach und nach in der Vorlesung erklärt!

Wichtig: Schlagen Sie in der API Dokumentation die Klasse *JFrame* und dort die Methoden *setTitle*, *setSize* und *setVisible* nach und versuchen Sie, deren Funktion so weit wie möglich zu verstehen.

Die Bedeutung der Eigenschaft *setDefaultCloseOperation* ist nicht sofort ersichtlich:
Wenn Sie die Zeile *setDefaultCloseOperation* auskommentieren (// einfügen) und das Fenster schließen, verschwindet das Fenster zwar, aber die Anwendung läuft weiter, was man am roten Quadrat oberhalb des Konsolen-Fensters sieht (siehe Bild).



Durch Klicken auf das rote Quadrat wird die Anwendung beendet und das Quadrat wird grau.

Dieses Verhalten ändert sich, wenn die Eigenschaft *DefaultCloseOperation* wieder gesetzt wird. Schlagen Sie die Methode *setDefaultCloseOperation* sowie die Eigenschaft *EXIT_ON_CLOSE* in der API Beschreibung der Klasse *JFrame* nach.

Fügen Sie folgende Zeilen vor *fenster.setVisible(true)* ein und führen Sie Ihr Programm aus:

```
JPanel p = new JPanel();
p.setBackground(Color.GREEN);
JButton knopf = new JButton("Do not push!");
p.add(knopf);
fenster.add(p);
```

Im Inneren des Fensters sehen Sie jetzt eine grüne Fläche mit einem klickbaren Knopf.