

UNIVERSITÄT TÜBINGEN  
PROF. DR.-ING. HENDRIK P.A. LENSCH  
LEHRSTUHL FÜR COMPUTER GRAFIK  
DENNIS BUKENBERGER  
(DENNIS.BUKENBERGER@UNI-TUEBINGEN.DE)  
RAPHAEL BRAUN  
(RAPHAEL.BRAUN@UNI-TUEBINGEN.DE)

EBERHARD KARLS  
UNIVERSITÄT  
TÜBINGEN



16. APRIL 2019

## INFORMATIK II ÜBUNGSBLATT 0

**Abzugeben bis:** Dienstag 23.04.2019, 13:29 Uhr

### InfoMark

Wir verwenden in diesem Semester das eigens dafür entwickelte Abgabe System InfoMark

<https://infomark.informatik.uni-tuebingen.de>.

- Auf YouTube finden Sie eine kurze Einführung in die Benutzung des Systems:

<https://www.youtube.com/watch?v=ifyUssK6PJ4>

- Registrieren Sie sich bei InfoMark. Sie müssen ihren Account via E-Mail bestätigen. Beachten Sie, dass hierfür ihre **studentische** E-Mail-Adresse (@student.uni-tuebingen.de) benötigt wird. Wir können aus technischen Gründen nur an Adressen der **uni-tuebingen.de** Domain E-Mails versenden.
- Schreiben Sie sich in den Kurs **Informatik II (INF1120)** ein (Enroll).
- Unter Gruppen Präferenzen geben Sie ihre Präferenzen für die Übungsgruppenauswahl an. Die Zuweisung wird anhand ihrer Angaben (via Integer-Linear-Programming) optimiert. Wir können nicht garantieren, dass Sie einer bestimmten Gruppe zugeteilt werden. Es ist ratsam für alle Übungsgruppen ihre Präferenzen anzugeben. Falls Sie nicht Ihrer Wunschgruppe zugeordnet werden können, wird eine Alternative gemäß Ihrer Präferenzen gewählt.
- Unter Sheets finden Sie die bisherigen Übungsblätter.
- Unter Folien finden Sie die Folien aus der Vorlesung.
- Unter Material finden Sie Zusatzmaterial aus der Vorlesung.
- Mit dem **Show** Button neben jedem Übungsblatt können Sie sich die einzelnen Aufgaben (Tasks) zu dem Blatt anzeigen lassen.
- Die Abgabe des Übungsblatt erfolgt für jeden Task *einzel*n. Mit dem **Show** Button kommen Sie zur Task-Übersicht, auf der Sie vor dem Ablauf der Abgabefrist Abgaben tätigen können.
- Ab Minute 1:05 im YouTube-Video sehen Sie wie eine Abgabe getätigt wird. Achten Sie darauf, dass Sie immer das **main** Verzeichnis zu einer **Zip**-Datei komprimieren und diese Hochladen müssen.
- Nach dem Upload wird ihre Abgabe auf dem Server automatisch getestet. Wir verwenden zwei Arten von Tests, deren Ausgabe Sie im Video bei 1:21 sehen können:

**ClassStructureTest** Ein *ClassStructureTest* überprüft ob Ihre Abgabe alle von uns geforderten Klassen und Methoden enthält und ob diese die von uns vorgegebenen Signaturen tragen. Bei Fehlschlag lesen Sie bitte die Fehlermeldung genau durch und passen Sie ihren Code entsprechend an. Bitte ändern Sie keine Methodennamen. Die von uns bereitgestellten Codevorlagen beinhalten die richtigen Signaturen, falls gegeben.

**ValueTest** Ein *ValueTest* testet das Verhalten und die Ergebnisse der von Ihnen implementierten Funktionen. Der Test wird bestanden und zeigt [ OK ], wenn ihre Funktion für alle Testfälle die erwarteten Werte liefert. Falls nicht, wird eine Fehlermeldung angezeigt, die genauere Informationen enthält. Im Video sehen Sie:

```
[ FAILED ] PlusValueTest:  
  Error 1/1  
    - Tag: failure  
    - Typ: junit.framework.AssertionFailedError  
    - Msg: plus(0, 2) expected: <2> but was <-2>
```

- Die erste Zeile gibt an welcher Test gelaufen ist.
- Die zweite Zeile gibt an wieviele Fehler in dem Test erkannt wurden.
- Die dritte Zeile enthält die Art des Fehlers. Ein fehlgeschlagener Test zeigt hier typischerweise `failure` an.
- Die vierte Zeile gibt Informationen über die interne Funktion des Tests der fehlgeschlagen ist. Im obigen Beispiel handelt es sich um einen `AssertionFailedError`. Dies bedeutet, dass eine von uns formulierte Erwartung (assertion) nicht erfüllt wurde.
- Die letzte und wichtigste Zeile enthält die konkrete Fehlermeldung. Hier wurde die Methode `plus` mit den Eingabewerten 0 und 2 getestet. Der Test erwartet die Ausgabe 2, der Aufruf `plus(0, 2)` hat jedoch -2 ausgegeben. Der Fehler befindet sich also in der `plus` Methode.

Wenn keine Fehler mehr detektiert werden sollten alle Tests [ OK ] ausgeben. Ein ValueTest kann nur erfolgen werden wenn der ClassStructureTest erfolgreich ist.

- Sie können vor der Abgabefrist beliebig viele Abgaben tätigen. Nur ihre letzte Abgabe wird von den Tutoren bewertet.
- Die Ausgabe der Unit-Tests ist *nicht* bindend. Es handelt sich hierbei lediglich um eine optionale Hilfestellung für Sie und die Tutoren. Die Tutoren haben Zugriff auf weitere Testausgaben, sodass triviale Lösungen, die nur den öffentlichen Tests bestehen, ebenfalls detektiert werden.
- Sie können Ihre Punkte und das Feedback Ihres Tutors einsehen, sobald Ihre Abgabe nach der Abgabefrist bewertet wurde. Das Feedback der Tutoren basiert auf Ihrer Implementierung und nicht nur auf der Unit-Tests Ausgabe.
- Mit dem **Rating** unter den **Test Results** bitten wir Sie uns Feedback zu den Aufgaben zu geben.

## Aufgaben

Im Laufe des Kurses Informatik II wird es 12 Übungsblätter mit jeweils 30 erreichbaren Punkten geben. Die Aufgabe 0.2 auf diesem Blatt bietet die Möglichkeit 2 Zusatzpunkte zu erhalten.

### 0.1 Die Arbeitsumgebung (0 Punkte)

In dieser Aufgabe sollen Sie sich mit Ihrer Programmierumgebung für die Programmierung mit Java vertraut machen. Folgende Instruktionen sollen Ihnen bei der Installation der Arbeitsumgebung Eclipse helfen:

- a) Installation unter Ubuntu:

- Installieren Sie das Java SDK mit dem Befehl: `sudo apt install openjdk-11-jdk`. (Sie können die Installation überprüfen indem Sie den Befehl `java -version` ausführen: Dann sollte folgende Anzeige erscheinen: `openjdk version '10.0.2' 2018-07-17`.)

- Downloaden Sie Eclipse für Linux von den Download Links auf

<https://www.eclipse.org/downloads/packages/release/2019-03/r/eclipse-ide-java-developers>

- Entpacken Sie die .zip-Datei in ein gewünschtes Verzeichnis.

b) Installation unter Windows:

- Downloaden Sie das Java SE Development Kit 11 unter:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>

- Installieren Sie das Java SDK in ein gewünschtes Verzeichnis.
- Fügen Sie das `bin` Verzeichnis des jdk-11 der `Path` Variable von Windows hinzu. Eclipse kann ansonsten das jdk nicht finden.

- 1) Im Dateimanager Rechtsklick auf Computer "Dieser PC" und "Eigenschaften" wählen.
- 2) Auf "Erweiterte Systemeinstellungen" klicken.
- 3) Auf "Umgebungsvariable" klicken.
- 4) Auf "Path" und dann "Bearbeiten..." klicken.
- 5) Fügen Sie an das Ende der Zeile ein Semicolon (;) ein und anschließend den Pfad zum `bin` Verzeichnis des jdks.

- Downloaden Sie Eclipse für Windows mittels folgendem Download-Link:

<https://www.eclipse.org/downloads/packages/release/2019-03/r/eclipse-ide-java-developers>

- Entpacken Sie die .zip-Datei in ein gewünschtes Verzeichnis.

c) Installation unter OS-X (ungetestet):

- Laden Sie das Java SE Development Kit 11 herunter:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>

- Installieren Sie das Java SDK in ein gewünschtes Verzeichnis.
- Laden Sie Eclipse für Mac von den "Download Links" herunter:

<https://www.eclipse.org/downloads/packages/release/2019-03/r/eclipse-ide-java-developers>

- Entpacken Sie die Datei in ein gewünschtes Verzeichnis.

## 0.2 Ein erstes Java-Programm (2 Punkte)

In dieser Aufgabe soll ein erstes Java-Programm erstellt und ausgeführt werden.

- Öffnen Sie Eclipse und wählen Sie einen Pfad für den Workspace.
- Gehen Sie auf `File -> New > Java Project` und geben Sie als Projektname `MeinErstesJavaProjekt` ein und klicken Sie auf `Next`.
- Entfernen Sie das Häkchen ganz unten bei `create module-info.java file` und klicken Sie auf `Finish`.
- Links im Package Explorer sehen Sie nun Ihr Projekt. Klicken Sie mit der rechten Maustaste darauf und wählen Sie `New -> Package`.
- Nennen Sie das Package `main`.
- Klicken Sie mit der rechten Maustaste auf das `main` Package und wählen Sie `New -> Class`.

- g) Geben Sie `Hello` als Name der Klasse ein und setzen Sie das Häkchen bei `public static void main(String[] args)`. Behalten Sie alle anderen Einstellungen bei und klicken Sie auf `Finish`.
- h) Eclipse öffnet die frisch angelegte Datei `Hello.java`. Vervollständigen Sie den Code, damit er dem Unteren gleicht:

---

```
1 package main;
2
3 public class Hello {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello World!");
8         System.out.println("6 + 2 = " + plus(6, 2));
9     }
10
11     public static int plus(int a, int b) {
12         return a - b;
13     }
14 }
```

---

Die Bedeutung der einzelnen Zeilen wird im Verlauf der Vorlesung erklärt.

- i) Speichern Sie ihre Änderungen.
- j) Klicken Sie auf Run (grüner Pfeil Button in der oberen Leiste, oder drücken Sie F11).
- k) Nun sollten Sie im unteren Bereich (Console) folgende Ausgabe erhalten:  
`Hello World!`  
`6 + 2 = 4`
- l) Ändern Sie das Programm so ab, dass es anstatt "Hello World!" Ihren Namen ausgibt.
- m) Ihnen ist sicher aufgefallen, dass  $2 + 6$  nicht 4 ergibt. Um dies in Ordnung zu bringen, ersetzen Sie das Minus in Zeile 12 durch ein Plus.
- n) Jetzt sollte das korrekte Ergebnis ausgegeben werden.
- o) Verpacken Sie nun den `main` Ordner als Zip-Datei. Ihre Zip-Datei sollte folgende Struktur haben:

```
<name>.zip
  main
    Hello.java
```

- p) Laden Sie die Datei `<name>.zip` in das **InfoMark** System hoch. Wenn Sie alle Anweisungen richtig befolgt haben, müssten Sie nach dem Upload folgendes Feedback bekommen:

```
[ OK ] PlusValueTest:
[ OK ] HelloClassStructureTest:
```

Dies bedeutet, dass Ihr Programm allen Tests standgehalten hat. Glückwunsch!