

Einführung in die Programmierung
WS 2018/19

Übungsblatt 0: Einführung

Besprechung: 22.-26.10.18

Mit Ihrer CIP-Kennung steht Ihnen die Rechnerinfrastruktur des CIP-Pools zu Verfügung. Die Übungsaufgaben und Hilfestellungen beziehen sich grundsätzlich auf diese Arbeitsumgebung. Natürlich können Sie auch auf Ihrem eigenen Rechner arbeiten. Bei auftretenden Problemen bemühen Sie sich bitte um folgende Hilfestellung in angegebener Reihenfolge:

1. Fast jedes Problem, das Sie haben, hatten schon 7 Leute vor Ihnen. Nutzen Sie eine Suchmaschine Ihrer Wahl und investieren Sie ein paar Minuten zur Onlinesuche.
2. Befragen Sie Ihren Tutor direkt in der Übungsstunde oder per Email.
3. Schreiben Sie den Übungsassistenten (fromm@dbs.ifi.lmu.de, richter@dbs.ifi.lmu.de) eine Email oder kommen Sie direkt zu uns ins Büro (Oettingenstrasse 67, Raum F110+F104). Wenn Sie vorher einen Termin vereinbaren, dann wird sicherlich einer von uns hier sein.
4. Auch bei Fragen zum Übungs- und Klausurablauf oder Anmeldemodalitäten setzen Sie sich bitte so früh wie möglich mit uns in Verbindung.

Aufgabe 0-1 *Arbeiten im CIP-Pool im Institut für Informatik*

• **Computer, Linux, KDE, ...**

1. Um die Computer in den CIP-Pools im Keller der Oettingenstrasse 67 bzw. in den anliegenden Barackengebäuden benutzen zu können, brauchen Sie die CIP-Kennung.
2. Finden Sie einen freien Computer und loggen Sie sich ein. Zuerst geben Sie Ihre Kennung ein, danach Ihr Passwort. Kurze Zeit später sehen Sie die KDE-Oberfläche. Hier finden Sie alles, was sie brauchen.
3. Normalerweise sollte sich bereits ein Webbrowser geöffnet haben. Falls nicht, dann benutzen Sie das Zahnrad in der unteren linken Ecke, um ein Menü zu öffnen, in dem Sie einen Webbrowser suchen können.
4. Zum Editieren von Dateien brauchen Sie einen Editor (wie der Name es schon sagt). Versuchen Sie, im Menü unten links einen Eintrag „Emacs“ bzw. „XEmacs“ in einem der Untermenüs zu finden und Emacs zu starten. Emacs kann vieles und noch viel mehr. Die wichtigsten Funktionen wie Öffnen und Schließen von Dateien, Speichern und Beenden sind über die Menüs am oberen Rand des Fensters zu erreichen. Sie können genauso jeden anderen beliebigen Editor wie z.B. „Kate“ benutzen.
5. Sie können diesen Editor auch nutzen um Java-Dateien für Ihre Übungsblätter zu erstellen. Hierfür speichern Sie die Datei mit der Endung „.java“ statt „.txt“ ab.

- **Dateien und Verzeichnisse**

Mit Ihrer CIP-Kennung erhalten Sie Speicherplatz auf den Rechnern der CIP-Pools, Ihr sogenanntes Homeverzeichnis. Dies ist unabhängig von dem Rechner, an dem Sie sitzen, Sie können also beim nächsten Einloggen einen anderen Rechner nehmen.

Es gibt zwei Möglichkeiten Ihre Verzeichnisse zu verwalten:

1. Möglichkeit: Verwaltung über ein Terminal

1. Starten Sie eine Terminal-Emulation. In der unteren Bildleiste sehen Sie das Icon eines Bildschirms mit schwarzem Hintergrund. Darauf klicken! Über diese sogenannte Shell können Sie mit dem Computer (genauer gesagt mit dem Betriebssystem Linux) kommunizieren.

2. Lassen Sie sich den Inhalt Ihres Homeverzeichnisses anzeigen.

```
ls
```

3. Erstellen Sie ein Unterverzeichnis. Denken Sie sich dazu einen besonders tollen Namen aus.

```
mkdir Name
```

4. Wechseln Sie von Ihrem Homeverzeichnis in das neue Unterverzeichnis.

```
cd Name
```

Lassen Sie sich dort den Inhalt anzeigen. Nichts da? Kein Wunder.

5. Erstellen Sie dort eine Datei. Denken Sie sich dazu einen zweiten Namen aus.

```
touch Name2
```

Diese Art, eine neue Datei zu erstellen, werden Sie später kaum brauchen, denn jedes normale Anwendungsprogramm erstellt Dateien für Sie. Lassen Sie sich anzeigen, wann die Datei erstellt wurde und wie groß sie ist.

```
ls -al Name2
```

Oftmals wird der Sinn bzw. Typ einer Datei durch eine Endung deutlich gemacht. Löschen Sie die bereits erstellte Datei wieder.

```
rm Name2
```

Erstellen Sie die Datei `Name2.java`.

6. Wechseln Sie in das Verzeichnis über dem aktuellen.

```
cd ..
```

7. Kopieren Sie die Datei aus dem Unterverzeichnis in das aktuelle. Das aktuelle Verzeichnis wird durch einen Punkt gekennzeichnet, während zwei Punkte das darüberliegende Verzeichnis bezeichnen.

```
cp Name/Name2.java .
```

8. Löschen Sie das Unterverzeichnis.

```
rmdir Name
```

Geht nicht. Zuerst müssen Sie Ihre Datei aus dem Unterverzeichnis entfernen. Entweder Sie löschen sie, oder Sie verschieben sie in ein anderes Verzeichnis, z.B. das darüberliegende.

```
cd Name  
mv Name2.java ..
```

Beachten Sie, dass dadurch die gleichnamige Datei im Verzeichnis .. überschrieben wurde. Außerdem läßt sich mv auch zum Umbenennen von Dateien benutzen. Gehen Sie in das Verzeichnis, in dem Name2.java liegt, und denken Sie sich einen weiteren Namen aus.

```
mv Name2.java Name3.java
```

Lassen Sie sich wieder den Inhalt des Verzeichnisses anzeigen.

2. Möglichkeit: Verwaltung über eine Benutzeroberfläche

Anstelle des Terminals können Sie auch alle oben genannten Schritte über den Datei-Manager Konqueror ausführen.

- **Eigene Ordnerstruktur:**

Sie werden im Laufe des Semesters für diese Vorlesung (und vielleicht auch für andere) viele Dateien herunterladen. Wir empfehlen Ihnen, in Ihrem Home-Verzeichnis von Anfang an eine geeignete Verzeichnisstruktur dafür vorzubereiten. Sie können zum Beispiel nach folgendem Schema vorgehen: Ein Verzeichnis EIP enthält ein Unterverzeichnis uebungen in dem wiederum Unterverzeichnisse für die einzelnen Übungen (z.B. u01 bis u14) enthalten sind. Für jedes Übungsblatt (also in jedem Übungsverzeichnis) legen Sie bitte ein Verzeichnis loesung an, in dem Sie die Lösungs-Dateien zu den einzelnen Aufgaben ablegen. Haben Sie alle Lösungen, die Sie abgeben möchten, fertig, dann erstellen Sie ein zip-Archiv aus diesem Verzeichnis. Wenn Sie mit einer shell im Verzeichnis des aktuellen Übungsblattes sind (z.B. EIP/uebungen/u01), erzeugen Sie mit dem Befehl

```
zip loesung01.zip loesung/*
```

ein zip-Archiv namens loesung01.zip, das alle Dateien aus dem Lösungsverzeichnis loesung enthält.

Wichtig: Achten Sie bitte darauf, dass Ihre Lösungsdateien die korrekten, d. h. die in der Angabe geforderten Namen haben, sonst kann Ihre Lösung nicht der richtigen Aufgabe zugeordnet werden. Java-Dateien, die nicht fehlerfrei kompilierbar sind, werden im Allgemeinen nicht korrigiert.

- **Internet und WWW**

Starten Sie einen WWW-Browser wie z.B. Mozilla oder Konqueror. Diese finden Sie entweder im Menü unten links oder als Icons auf Ihrem Desktop.

- **Zugriff von zu Hause aus.**

Mittels SSH können Sie sich von zu Hause auf einem Rechner im CIP-Pool des Instituts für Informatik einloggen. Siehe hierzu

<http://www.rz.ifi.lmu.de/FAQ/Aussenzugriff.faq.html>

Aufgabe 0-2 *Arbeiten unter Windows*

Falls Sie Windows als Betriebssystem nutzen, dann laden Sie sich das JDK (Java Developer Kit) in der aktuellen Version herunter. Achtung, verwechseln Sie das JDK nicht mit der JRE (Java Runtime Environment), dass lediglich die Laufzeitumgebung für Java zur Verfügung stellt. Mit der JDK können Sie Ihren Quellcode in lauffähige Programme kompilieren, mit der JRE führen Sie diese Programme aus. Eine JRE haben Sie vermutlich schon für andere Anwendungen installiert. In der SDK (Software Development Kit) ist beides enthalten.

Testen Sie, ob Ihre Umgebungsvariablen richtig gesetzt sind: Öffnen Sie eine Kommandozeile (z.B. mit `Windows+R` → `cmd`) und nutzen Sie den Befehl `java`. Falls verschiedenste Java-Optionen ausgegeben werden, ist der Pfad zur JRE richtig gesetzt. Benutzen Sie anschließend `javac`. Falls Windows Ihren Befehl nicht finden kann und Sie die nötige Software installiert haben, gehen Sie wie folgt vor:

- Lokalisieren Sie den Installationspfad zu Ihrer Java-Version.
- Öffnen Sie die „Erweiterten Systemeinstellungen“ der Systemsteuerung.
- Klicken Sie auf „Umgebungsvariablen...“.
- Doppelklicken Sie auf die Path-Variable.
- Fügen Sie den Pfad zur `java.exe` und `javac.exe` hinzu, z.B.
C:\ProgramFiles\Java\jdk1.8.1_01\bin
oder C:\ProgramFiles\Java\jdk-11\bin
- Bestätigen Sie alle Änderungen.
- Testen Sie ein weiteres mal, ob eine (neu geöffnete) Konsole die oben genannten Befehle nun kennt.

Aufgabe 0-3 *Der Klassiker*

Diese Aufgabe dient zur Erläuterung der allgemeinen Vorgehensweise bei der Bearbeitung und Abgabe von Übungsaufgaben.

Legen Sie eine Datei `HelloWorld.java` in Ihrem Arbeitsverzeichnis an. Nutzen Sie dann einen Texteditor, um den Inhalt der Datei ändern zu können. Legen Sie wie folgt eine Klassendeklaration `HelloWorld` an. Achten Sie darauf, dass die Schreibweise von Datei und Klasse identisch sein muss. Fügen Sie der Klasse eine statische Methode `main()` hinzu, die bei Aufruf die Textzeile `Hallo, Welt!` in der Konsole ausgibt.

```
public class HelloWorld {
    public static void main(String[] args) {
        // Einfach einen kleinen Text ausgeben
        System.out.println("Hallo, Welt!");
    }
}
```

Kompilieren Sie nun den Code, indem Sie in der Konsole im richtigen Verzeichnis `javac HelloWorld.java` ausführen. Im besten Fall wird eine Datei `HelloWorld.class` erstellt. Ansonsten gibt Ihnen der Compiler eine Auflistung von Fehlern, die ihn an der Kompilierung gehindert haben. Nutzen Sie Ihr Wissen oder suchen Sie online nach einer Lösung Ihres eventuellen Problems. Ein Webbrowser gehört nicht zwingend, aber fast immer zu den Basistools der meisten Programmierer und Sie werden dort viel Zeit mit der Fehlersuche verbringen.

Wurde die Kompilierung ohne Fehler ausgeführt, dann können Sie Ihr Programm mit `java HelloWorld` ausführen.

Ändern Sie nun die Ausgabe in eine Nachricht Ihrer Wahl um. Zum Schluss testen Sie einen mathematischen Ausdruck: `System.out.println("5+3")`. Was passiert, wenn Sie die Anführungszeichen weglassen?

Aufgabe 0-4 *Syntaktische Fehler*

Die Datei `Addition.java` hat folgenden Inhalt:

```
public class Addition
{
    public static void main(String[] args)
    {
        System.out.print("Dieses Programm addiert drei Zahlen. ),
        int x = 3;
        y = 5;
        int z = 1.5;
        System.out.print("Die Summe von " + x + ", " + y + " und " + z " ist: ");
        System.out.println(x+y+z)
    }
}
```

Das Programm enthält sechs *syntaktische* Fehler. Korrigieren Sie alle syntaktischen Fehler, so dass das Programm kompilierbar ist. Die einfachste Möglichkeit ist natürlich, das Programm zu kompilieren, die Fehlermeldungen anzuschauen und den Quellcode entsprechend zu editieren. Die Fehler sind behoben, wenn der Compiler keine mehr meldet. Das richtige Vorgehen hierbei ist, immer den ersten Fehler zu korrigieren, den der Compiler meldet, und dann von neuem zu kompilieren. Der Grund hierfür ist, dass die weiteren Fehler, die der Compiler nach dem ersten Fehler meldet, nicht mehr in jedem Fall verlässlich sind.

Aufgabe 0-5 *Semantische Fehler*

Die Datei `Durchschnitt.java` hat folgenden Inhalt:

```
public class Durchschnitt
{
    public static void main(String[] args)
    {
        int summe = 1;

        int ersteZahl = 3;
        System.out.println("erste Zahl: "+ersteZahl);

        summe = summe + ersteZahl ;

        int zweiteZahl = 2;
        summe = summe + ersteZahl;
        System.out.println("zweite Zahl: "+zweiteZahl);

        double durchschnitt = summe / 2;
        System.out.println("Der Durchschnitt der beiden Zahlen ist: "+durchschnitt);
    }
}
```

Das Programm enthält drei *semantische* (logische) Fehler. Diskutieren Sie, worin diese Fehler bestehen und wie man sie beheben kann. Korrigieren Sie das Programm entsprechend.