

## **Liebe Kursbeleger,**

Zielsetzung des Kurses 1618 ist die Vermittlung grundlegender Konzepte und Begriffe der Objektorientierten Programmierung. Dies geschieht überwiegend am Beispiel der Programmiersprache Java. Um die Einsendeaufgaben zu lösen und praktische Erfahrungen mit der Programmiersprache Java zu sammeln, müssen Sie selbst eigene Java-Programme implementieren, in denen Sie die verschiedenen Konstrukte der Sprache ausprobieren.

Dazu benötigen Sie eine Arbeitsumgebung, die Ihnen erlaubt, Quellcode zu editieren, zu compilieren und schließlich auszuführen. Es gibt prinzipiell eine Vielzahl möglicher Wege, eine solche Arbeitsumgebung zusammenzustellen. Falls Sie bereits mit Java gearbeitet haben, haben Sie vermutlich "Ihren" Weg schon gefunden und es spricht nichts dagegen, die bestehende Umgebung weiter zu verwenden.

Für alle anderen beschreiben wir im Folgenden *eine* mögliche Vorgehensweise, welche aber den Vorteil hat, dass sie unabhängig von bereits auf Ihrem Rechner befindlichen Java-Versionen funktionieren sollte. Die neue Arbeitsumgebung für die Bearbeitung des Kurses bleibt von Ihren sonstigen Programmen getrennt, es wird auf Ihrem Rechner kein (unter Sicherheitsaspekten nicht unproblematisches) Java-Plugin in Ihrem Browser installiert und Sie können sogar die gesamte Arbeitsumgebung auf einfache Weise auf einen anderen Rechner übertragen.

Sofern Sie "unseren" Vorschlag verwenden, halten Sie sich aber bitte *exakt* an die beschriebene Vorgehensweise. Falls dann trotzdem Probleme, auftreten, lassen die sich meist schnell lösen. Scheuen Sie sich bitte nicht, ggf. in der Newsgroup `feu.informatik.kurs.1618.diskussion` im passenden Thread nachzufragen!

## **IDE oder Kommandozeile?**

Um Ihnen einen möglichst zügigen Einstieg in das praktische Arbeiten zu ermöglichen, haben wir uns für die Verwendung der Integrierten Entwicklungsumgebung (IDE) Eclipse entschieden, welche Ihnen die Auseinandersetzung mit einigen problemträchtigen Details vorerst erspart. Sobald Sie sich ein wenig mit Java vertraut gemacht haben, empfehlen wir Ihnen dennoch, einmal durch das "Tal der Tränen" zu gehen und ein einfaches Java-Programm im Texteditor des Betriebssystems zu schreiben, auf der Kommandozeile mit dem Compiler des JDK zu kompilieren und auszuführen.

## **Welches Java und woher?**

Die derzeit aktuellste Java-Version ist Java 11. Der im Kurs 1618 beschriebene Stoff behandelt aber keine Java-Sprachmittel, die erst nach Java 8 neu hinzugekommen sind. Wir empfehlen deswegen die Verwendung von Java 8. Da wir Eclipse verwenden und da diese IDE einen eigenen Compiler mitbringt, würde es prinzipiell genügen, lediglich die Java-Laufzeitumgebung (Java Runtime Environment - JRE) zu installieren. Wir werden dennoch das komplette Java Development Kit (JDK) verwenden, u.a. weil dieses zusätzliche Werkzeuge enthält, welche Sie spätestens für das Programmierpraktikum ohnehin benötigen werden.

## **32 oder 64 Bit?**

Die aktuellsten Eclipse-Versionen werden nur noch für 64-Bit-Systeme bereitgestellt. Wir verwenden daher auch alle anderen Bestandteile der Arbeitsumgebung in der 64-Bit-Version. Sollten Sie darauf angewiesen sein, den Kurs mit einem 32-Bit-System zu bearbeiten, fragen Sie bitte in der 1618-Newsgruppe im passenden Thread nach.

## **Für Linux-Anwender**

Die gängigen Linux-Distributionen bringen eigene Paketverwaltungen mit, auf die wir hier nicht im Einzelnen eingehen können. Die folgende Beschreibung richtet sich daher in erster Linie an Verwender von Microsoft-Windows. Sie können aber prinzipiell auch auf Ihrem Linux-System so vorgehen, dass Sie Eclipse und das JDK – unter Umgehung der Paketverwaltung – als komprimierte Tar-Archive (tar.gz) herunterladen, analog zu der untenstehenden Beschreibung für Windows in einen Ordner z.B. in Ihrem Home-Verzeichnis entpacken und dann von dort aus Eclipse starten. Sollten Sie nicht auf die Unterstützung der Paketverwaltung verzichten wollen, sind evtl. folgende Links hilfreich:

<https://wiki.ubuntuusers.de/Eclipse>

<https://wiki.ubuntuusers.de/Java/Installation#OpenJDK>

<http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-debian.html>

## Es geht los... Downloads

Zunächst müssen Sie einige Dateien auf Ihren Rechner herunterladen:

- Java SE Development Kit 8

Das JDK bekommen Sie über die Webseite

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

Sie finden dort unter „Java SE 8uxxx“ (wobei xxx für eine fortlaufende Versionsnummerierung steht) einen Download-Knopf für das JDK. Dieser bringt Sie zu einer Seite auf der Sie die Datei `jdk-8uxxx-windows-x64.exe` herunterladen können. Vor dem Download müssen Sie dort mit dem Radio-Button weiter oben bestätigen, dass Sie die Lizenzvereinbarung akzeptieren.

- Java SE 8 Documentation

Diese bekommen Sie ebenfalls über die Webseite

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Sie finden dort weiter unten unter "Additional Resources" einen Link "Java SE 8 Documentation" mit einem Download-Knopf. Dieser führt Sie zu einer Seite, auf der Sie – wieder nach Akzeptieren der Lizenzvereinbarung – die Datei `jdk-8uxxx-docs-all.zip` herunterladen können.

- Eclipse IDE for Java Developers

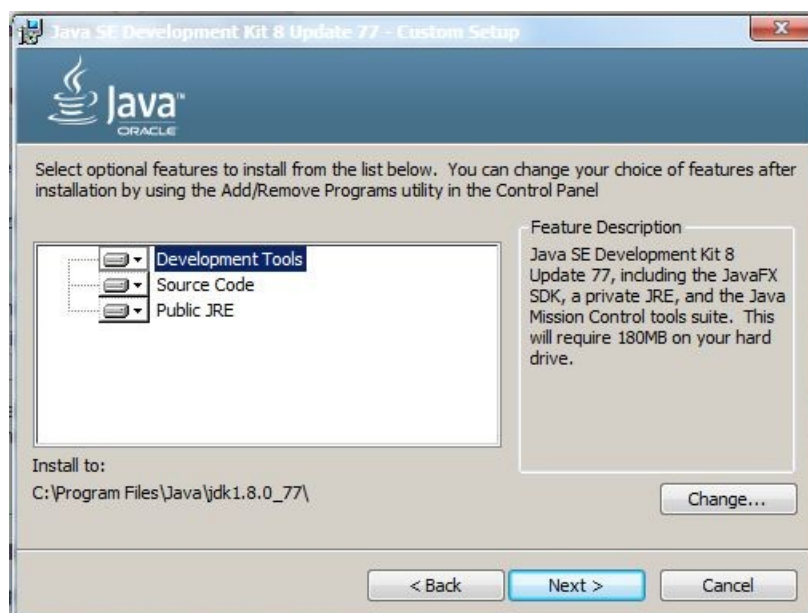
Suchen Sie auf der Seite <http://eclipse.org/downloads/eclipse-packages/> den Link zur "Eclipse IDE for Java Developers" (nicht verwechseln mit der "Eclipse IDE for Java EE Developers") in der 64-Bit-Version für Windows. Dieser lädt (Stand 12/2018) die Datei `eclipse-java-2018-12-R-win32-x86_64.zip` herunter.

- Vorkonfigurierter Workspace

Eclipse speichert Einstellungen und vom Benutzer angelegte Projekte in einem sog. Workspace. Über <http://feu.mpaap.de/eclipse/workspace.zip> erhalten Sie einen von uns vorkonfigurierten Eclipse-Workspace, in dem einige sinnvolle Einstellungen bereits getroffen wurden, als Datei `workspace.zip`.

## Installation

1. Legen Sie auf Ihrem Rechner ein Verzeichnis an, in dem wir dann alle Bestandteile der Arbeitsumgebung unterbringen werden. Dazu werden etwa 800 MB Platz benötigt. Das Verzeichnis dürfen Sie nicht *irgendwo* anlegen, sondern es muss ein Ort sein, an dem auch ein Programm ohne Administrator-Rechte schreiben darf. Das heißt insbesondere, dass Sie es auf dem Wurzel-Laufwerk (üblicherweise C:) nur unterhalb Ihres Benutzerverzeichnisses anlegen dürfen. Im Folgenden gehen wir davon aus, dass dieses Verzeichnis "kurs1618" heißt.
2. Legen Sie in diesem Verzeichnis ein Unterverzeichnis namens "jdk1.8" an. Führen Sie nun die Installationsdatei des JDK (jdk-8uxxx-windows-x64.exe) aus. Wenn Sie den folgenden Dialog sehen...



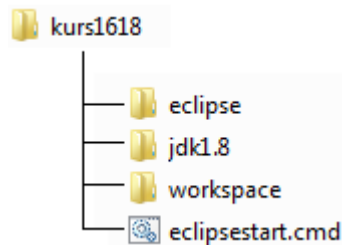
... ändern Sie bitte als Erstes über "Change..." das Installationsverzeichnis der „Development Tools“ auf das zuvor angelegte Verzeichnis "jdk1.8" im Ordner "kurs1618". Anschließend ändern Sie die Einstellung für das "Public JRE" so, dass dieses **nicht** installiert wird ("This feature will not be available"). Mit Klick auf "Next" erfolgt die Installation.

3. Verschieben Sie nun die Zip-Datei mit der Java SE 8 Documentation (jdk-8uxxx-docs-all.zip) in das Verzeichnis "jdk1.8" (nicht auspacken!) und benennen Sie sie in docs-all.zip um.
4. Entpacken Sie die Datei eclipse-java-2018-12-win32.zip und verschieben Sie das enthaltene Verzeichnis "eclipse" in Ihr Verzeichnis "kurs1618".
5. Entpacken Sie die Datei workspace.zip und verschieben Sie das enthaltene Verzeichnis "workspace" in Ihr Verzeichnis "kurs1618".

6. Legen Sie im Verzeichnis "kurs1618" eine neue Textdatei an und schreiben Sie folgendes hinein:

```
start .\eclipse\eclipse.exe -vm ".\jdk1.8\jre\bin\javaw.exe" -data ".\workspace"
```

Ändern Sie nun den Namen der Textdatei auf eclipsestart.cmd. Wenn Sie möchten, legen Sie sich eine Verknüpfung zu dieser Datei auf Ihrem Desktop an. Sie sollten nun in dem Verzeichnis, das bei Ihnen unserem Verzeichnis „kurs1618“ entspricht, folgende Struktur haben:



7. Wenn Sie nun die Datei eclipsestart.cmd ausführen, sollte Eclipse starten. Falls Sie beim Start eine Fehlermeldung erhalten, die besagt, dass der Workspace mit einer älteren Eclipse-Version erstellt wurde und deshalb konvertiert werden muss, erlauben Sie bitte die Konvertierung. Sie befinden sich nun im von uns vorkonfigurierten Workspace. Diesen erkennen Sie daran, dass dort bereits ein Projekt namens "HelloWorld" angelegt wurde.
8. Zum Abschluss binden Sie jetzt noch die "Java SE 8 Documentation" in Eclipse ein, damit Sie für Klassen der Java-Standard-Klassenbibliothek die API-Dokumentation auch dann direkt aus Eclipse anzeigen lassen können, wenn keine Online-Verbindung besteht. Dies erledigen Sie direkt aus Eclipse heraus. Unter "Window - Preferences - Java - Installed JREs " markieren Sie das dort als "Default" eingetragene jdk1.8 und klicken Sie auf "Edit". In dem sich öffnenden Dialogfenster "Edit JRE" findet sich unten ein Bereich mit den von Ihrem JRE verwendeten Bibliotheken. Markieren Sie dort die Bibliothek rt.jar (enthält die Klassen der Standard-Klassenbibliothek) und klicken Sie auf "Javadoc Location". Wählen Sie den Radio-Button "Javadoc in Archive" an und suchen Sie über "Browse" als "Archive-Path" den Pfad zur Datei "kurs1618\jdk1.8\docs-all.zip". Wenn Sie anschließend bei "Path within archive" auf "Browse" klicken, sollte automatisch das Verzeichnis "/docs/api" innerhalb der Zip-Datei vorgewählt sein. Bestätigen Sie dies mit "OK" und schließen Sie anschließend den Dialog ebenfalls mit "OK". Sie sollten sich nun wieder im Dialog "Edit JRE" befinden. Schließen Sie diesen mit "Finish" und verlassen Sie mit "OK" den Preferences-Dialog. Wenn Sie die Einstellung korrekt vorgenommen haben, können Sie nun im Java-Editor von Eclipse den Cursor auf ein Element der Standard-Klassenbibliothek setzen (z.B. auf das "System" in Zeile 3 der Klasse "Hello" im Projekt "HelloWorld") und dann mit der Tastenkombination SHIFT + F2 die

API-Doku zu diesem Element im Browser öffnen (beim ersten Mal dauert das einen Moment). Sollten Sie eine Anfrage Ihrer Firewall bekommen, ob Eclipse weiter geblockt werden soll, schalten Sie Eclipse bitte frei: Eclipse verwendet zur Anzeige der API-Doku einen integrierten Mini-Webserver.

9. Sollten Sie Ihre komplette Arbeitsumgebung auf einen anderen Windows-Rechner übertragen wollen (z.B. um einem Kommilitonen zu helfen) können Sie einfach das Verzeichnis "kurs1618" auf den Zielrechner kopieren. Sie müssen dann lediglich Schritt 8 erneut durchführen.

## **Erste Schritte mit Java und Eclipse**

Wer als Java-Anfänger zum ersten Mal Eclipse startet, ist sicher etwas überfordert von der Vielzahl der Bedienelemente. Zum Glück kann man sich diese aber schrittweise erschließen und sich zu Anfang auf das Wesentlichste beschränken.

Wenn Sie Eclipse mit dem vorkonfigurierten Workspace gestartet haben, sehen Sie im sog. Package-Explorer (oben links) ein von uns bereits angelegtes Java-Projekt namens "HelloWorld". In diesem befindet sich ein Ordner "src" für die zum Projekt gehörigen Java Quellcode-Dateien. In unserem Projekt ist dies genau eine Datei. Diese heißt Hello.java und liegt in einem "Package" names "de.mpaap.test". Was es genau mit Packages auf sich hat, lernen Sie später im Kurs. Für jetzt reicht es, zu wissen, dass sich Quellcode-Dateien (und damit die von ihnen definierten Java-Klassen) in solchen Packages befinden.

Den Quellcode der Klasse Hello sehen Sie im Editor-Fenster. Die Klasse hat eine Methode namens "main" mit einer ganz bestimmten Signatur, was bedeutet, dass die Klasse Hello als Einstiegspunkt für die Ausführung eines Programms verwendet werden kann. Probieren Sie das am Besten direkt einmal aus. Sie starten das Programm, indem Sie in der Bedienelemente-Leiste auf den grünen Knopf mit dem weißen, nach rechts zeigenden Dreieck klicken. Die main-Methode der Klasse Hello wird ausgeführt und in der Console oberhalb des Editorfensters erscheint die erwartete Ausgabe.

Für Ihre weiteren Experimente mit Java werden Sie eigene Java-Projekte mit eigenen Packages und eigenen Klassen anlegen wollen. Um zu sehen, wie das geht, schauen Sie sich am Besten unter <http://feu.mpaap.de/eclipse/filme.html> die Flash-Animation "Anlegen eines Projekts, Compilieren, Ausführen" an.

Des öfteren wird es beim Bearbeiten der Einsendeaufgaben nötig sein, dem Programm beim Start zusätzliche Parameter mitzugeben. Dazu können Sie in Eclipse einstellen, dass beim Start Ihres Programms ein Eingabefenster erscheint, in das Sie die Parameter eingeben können. Wie Sie das machen, sehen Sie auf der o.g. Seite im AVI-Film "Parameter übergeben beim Ausführen eines Programms".

Am Besten probieren Sie beides direkt aus, indem Sie die Klasse aus dem AVI-Film, in der das Aufsummieren von beim Programmstart übergebenen Parametern vorgeführt wird, einmal selbst nachbauen und testen.