

Vorkurs zum Modul 63611 „Einführung in die objektorientierte Programmierung“

Liebe Kursbeleger,

Zielsetzung des Moduls 63611 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 weiterzuverwenden.

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 Kurs-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.

Welche Java-Version?

Die derzeit aktuellste Java-Version ist Java 18. Der im Lehrtext 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 aktuelleren Eclipse-Versionen werden nur noch für 64-Bit-Systeme bereitgestellt. Wir verwenden daher auch alle anderen Bestandteile der Arbeitsumgebung in der 64-Bit-Version.

Windows / Linux / MacOS

Die weitere Vorgehensweise unterscheidet sich, je nachdem, welches Betriebssystem Sie verwenden. Wir beschreiben weiter unten zunächst die Vorgehensweise für Windows, dann für Linux, dann für MacOS.

Erste Schritte

Auf der letzten Seite dieses Dokuments finden Sie eine kurze Anleitung für die ersten Schritte mit der neu angelegten Arbeitsumgebung. Diese ist dann wieder für alle drei Betriebssysteme gleich. Diese sollten sie nach der Einrichtung der Arbeitsumgebung lesen und ausprobieren, bevor Sie die ersten Versuche mit Codebeispielen oder Aufgaben aus dem Lehrtext machen.

Anleitung für Windows

Es geht los ... Downloads

Zunächst müssen Sie einige Dateien auf Ihren Rechner herunterladen:

- Java SE Development Kit 8 (das JDK)

Dieses bekommen Sie über <https://adoptium.net/de/temurin/archive/?version=8>. Sie sehen dort verschiedene Versionen des JDK8. Wir verwenden die neueste, also die ganz oben. Klicken Sie dort auf den grauen Knopf in der Zeile „Windows x64“ und der Spalte „Binary“. Es wird eine Datei heruntergeladen, die den Namen „OpenJDK8U-jdk_x64_windows_hotspot_8uxxxxxx.zip“ hat, wobei xxxxxx für eine Versionsnummer steht (zur Zeit 402b06).

- Eclipse IDE for Java Developers

Aufgrund von Versionsproblemen verwenden wir bis auf Weiteres nicht die aktuellste Version von Eclipse, sondern die von Juni 2020. Suchen Sie auf der Seite <https://www.eclipse.org/downloads/packages/release/2020-06/r> den Link zur „Eclipse IDE for Java Developers“ (nicht verwechseln mit der „Eclipse IDE for Enterprise Java Developers“) in der Version für Windows. Dieser lädt die Datei „eclipse-java-2020-06-R-win32-x86_64.zip“ herunter.

- Vorkonfigurierter Workspace

Eclipse speichert Einstellungen und vom Benutzer angelegte Projekte in einem sog. Workspace. Über <https://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

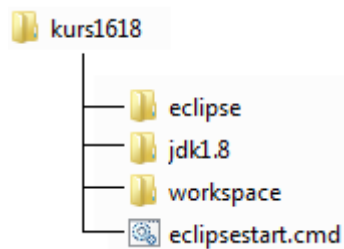
1. Legen Sie auf Ihrem Rechner ein Verzeichnis an, in dem wir dann alle Bestandteile der Arbeitsumgebung unterbringen werden. Dazu werden etwa 600 MB Platz benötigt. Das Verzeichnis dürfen Sie nicht *irgendwo* anlegen, sondern es muss ein Ort sein, an dem auch ein Programm ohne Administrator-Rechte schreiben darf. Das heißt insbesondere, dass Sie es auf dem Wurzel-Laufwerk (üblicherweise C:) nur unterhalb Ihres Benutzerverzeichnisses oder unterhalb von C:\Users\Public anlegen dürfen. Außerdem sollte der Pfad zu diesem Verzeichnis keine Sonderzeichen wie Umlaute oder Buchstaben mit Akzenten enthalten. Im Folgenden gehen wir davon aus, dass dieses Verzeichnis „kurs1618“ heißt.
2. Entpacken Sie die Zip-Datei mit dem JDK. Sie finden darin ein Verzeichnis namens „jdk8uxxxxxx“ (xxxxxx steht wieder für eine Versionsnummer). Verschieben Sie

dieses Verzeichnis in Ihr Verzeichnis „kurs1618“ und benennen Sie es um in „jdk1.8“.

3. Entpacken Sie die Zip-Datei mit der Eclipse-IDE. Sie finden darin ein Verzeichnis namens „eclipse“. Verschieben Sie dieses in Ihr Verzeichnis „kurs1618“.
4. Entpacken Sie die Datei workspace.zip. Sie finden darin ein Verzeichnis namens „workspace“. Verschieben Sie dieses in Ihr Verzeichnis „kurs1618“.
5. 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:



Weil es des öfteren vorkommt: Prüfen Sie zur Sicherheit noch einmal nach, ob Sie nicht versehentlich eine „gedoppelte“ Struktur erzeugt haben: Im Verzeichnis „workspace“ sollte sich *kein* weiteres Verzeichnis „workspace“ befinden, im Verzeichnis „eclipse“ *kein* weiteres Verzeichnis „eclipse“.

6. Sollten Sie später einmal 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.

Hinweis: Unter Windows ist leider auch unter Windows 10 immer noch voreingestellt, dass für bekannte Dateitypen die Namenserverweiterung nicht angezeigt wird. Sollte ein Doppelklick auf die Datei eclipsestart.cmd diese Datei im Texteditor öffnen, statt Eclipse zu starten, prüfen Sie bitte, ob Ihre Datei wirklich eclipsestart.cmd heißt und nicht vielleicht eclipsestart.cmd.txt. Um alle Dateiendungen angezeigt zu bekommen öffnen Sie in der Systemsteuerung die "Explorer-Optionen". Suchen Sie dort im Reiter "Ansicht" nach der Option "Erweiterungen bei bekannten Dateitypen ausblenden" und sorgen Sie dafür, dass diese Option *nicht* aktiviert ist.

Anleitung für Linux

Es geht los ... Downloads

Zunächst müssen Sie einige Dateien auf Ihren Rechner herunterladen:

- Java SE Development Kit 8 (das JDK)

Dieses bekommen Sie über <https://adoptium.net/de/temurin/archive/?version=8>. Sie sehen dort verschiedene Versionen des JDK8. Wir verwenden die neueste, also die ganz oben. Klicken Sie dort auf den grauen Knopf in der Zeile „Linux x64“ und der Spalte „Binary“. Es wird eine Datei heruntergeladen, die den Namen „OpenJDK8U-jdk_x64_linux_hotspot_8uxxxxxx.tar.gz“ hat, wobei xxxxxx für eine Versionsnummer steht (zur Zeit 402b06).

- Eclipse IDE for Java Developers

Aufgrund von Versionsproblemen verwenden wir bis auf Weiteres nicht die aktuellste Version von Eclipse, sondern die von Juni 2020. Suchen Sie auf der Seite <https://www.eclipse.org/downloads/packages/release/2020-06/r> den Link zur „Eclipse IDE for Java Developers“ (nicht verwechseln mit der „Eclipse IDE for Enterprise Java Developers“) in der Version für Linux. Dieser lädt die Datei „eclipse-java-2020-06-R-linux-gtk-x86_64.tar.gz“ herunter.

- Vorkonfigurierter Workspace

Eclipse speichert Einstellungen und vom Benutzer angelegte Projekte in einem sog. Workspace. Über <https://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“.

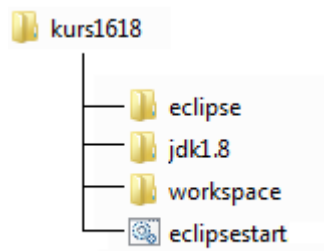
Installation

1. Legen Sie auf Ihrem Rechner ein Verzeichnis an, in dem wir dann alle Bestandteile der Arbeitsumgebung unterbringen werden. Dazu werden etwa 600 MB Platz benötigt. Das Verzeichnis dürfen Sie nicht *irgendwo* anlegen, sondern es muss ein Ort sein, an dem sie als Benutzer Schreibrechte haben. Außerdem sollte der Pfad zu diesem Verzeichnis keine Sonderzeichen wie Umlaute oder Buchstaben mit Akzenten enthalten. Im Folgenden gehen wir davon aus, dass dieses Verzeichnis „kurs1618“ heißt.
2. Entpacken Sie die tar-gz-Datei mit dem JDK. Sie finden darin ein Verzeichnis namens „jdk8uxxxxxx“ (xxxxxx steht wieder für eine Versionsnummer). Verschieben Sie dieses Verzeichnis in Ihr Verzeichnis „kurs1618“ und benennen Sie es um in „jdk1.8“.

3. Entpacken Sie die tar-gz-Datei mit der Eclipse-IDE. Sie finden darin ein Verzeichnis namens „eclipse“. Verschieben Sie dieses in Ihr Verzeichnis „kurs1618“.
4. Entpacken Sie die Datei workspace.zip. Sie finden darin ein Verzeichnis namens „workspace“. Verschieben Sie dieses in Ihr Verzeichnis „kurs1618“.
5. Legen Sie im Verzeichnis „kurs1618“ eine neue Textdatei an und schreiben Sie folgendes hinein:

```
#!/bin/sh
./eclipse/eclipse -vm jdk1.8/jre/bin/java -data workspace -showlocation &
```

Ändern Sie nun den Namen der Textdatei auf eclipsestart und machen Sie sie mittels „chmod +x eclipsestart“ ausführbar. Sie sollten nun in dem Verzeichnis, das bei Ihnen unserem Verzeichnis „kurs1618“ entspricht, folgende Struktur haben:



Weil es des öfteren vorkommt: Prüfen Sie zur Sicherheit noch einmal nach, ob Sie nicht versehentlich eine „gedoppelte“ Struktur erzeugt haben: Im Verzeichnis „workspace“ sollte sich *kein* weiteres Verzeichnis „workspace“ befinden, im Verzeichnis „eclipse“ *kein* weiteres Verzeichnis „eclipse“.

6. Sie können nun im Ordner „kurs1618“ ein Terminal öffnen und mit „./eclipsestart“ Eclipse mit dem von uns vorbereiteten Workspace öffnen. Nachdem Eclipse läuft, können Sie im Terminal „STRG +C“ eingeben und das Terminal schließen.
7. Sollten Sie später einmal 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.

Anleitung für MacOS

Zunächst müssen Sie einige Dateien auf Ihren Rechner herunterladen:

- Eclipse IDE for Java Developers

Laden Sie sich bitte von <https://www.eclipse.org/downloads/packages/> den Installer für die aktuelle Version von Eclipse herunter. Je nach Prozessortyp Ihres Mac wählen Sie die Version x86_64 (Intel-Prozessor) oder AArch64 (ab M1-Prozessor).

- Java SE Development Kit 8 (das JDK)

Dieses bekommen Sie über <https://adoptium.net/de/temurin/archive/?version=8>. Sie sehen dort verschiedene Versionen des JDK8. Wir verwenden die neueste, also die ganz oben. Klicken Sie dort auf den grauen Knopf in der Zeile „macOS x64“ und der Spalte „Binary“. Es wird eine Datei heruntergeladen, die den Namen „OpenJDK8U-jdk_x64_mac_hotspot_8uxxxxxx.tar.gz“ hat, wobei xxxxxx für eine Versionsnummer steht (zur Zeit 402b06).

- Vorkonfigurierter Workspace

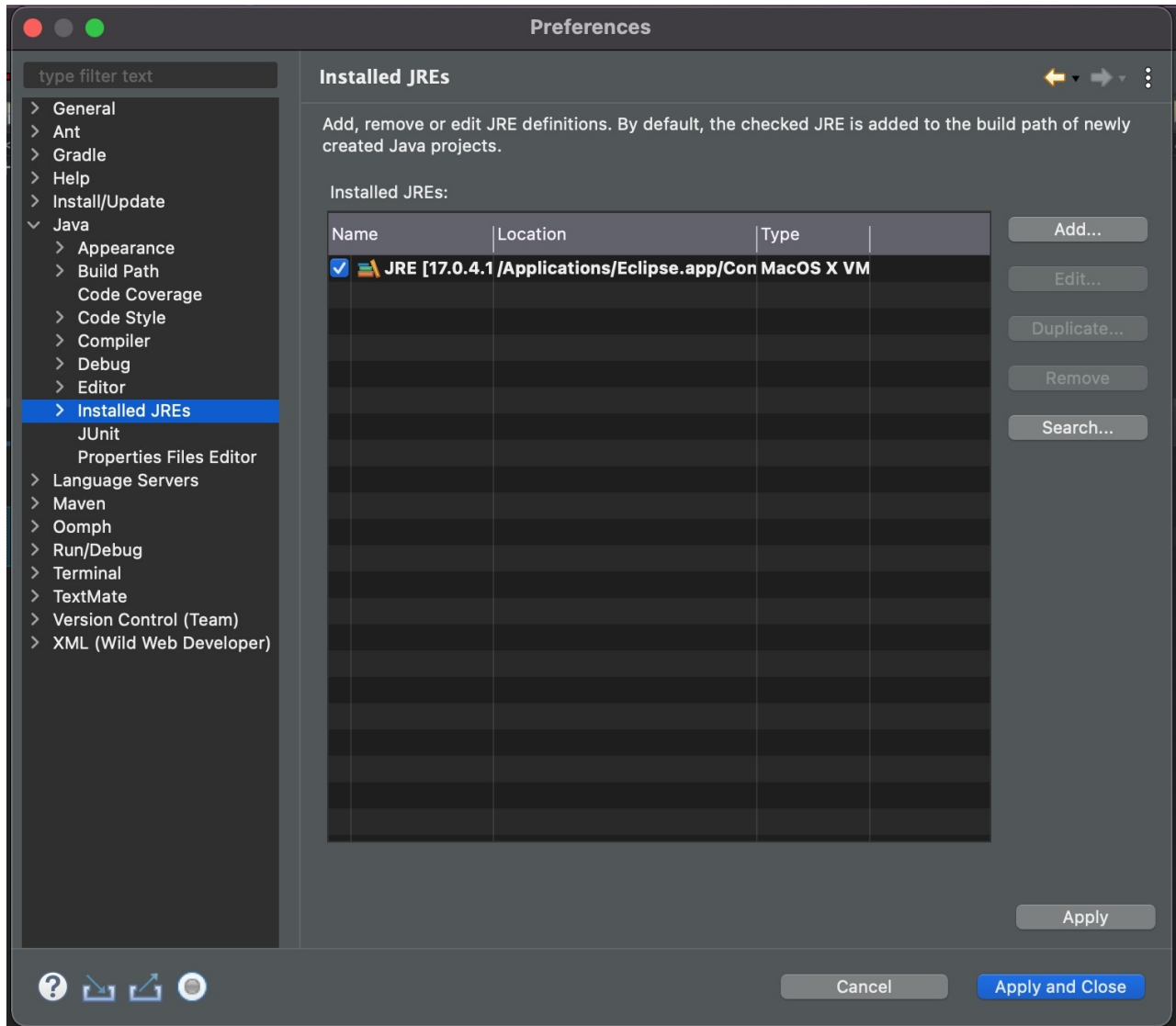
Eclipse speichert Einstellungen und vom Benutzer angelegte Projekte in einem sog. Workspace. Über <https://feu.mpaap.de/eclipse/workspace.zip> erhalten Sie einen von uns vorkonfigurierten Eclipse-Workspace, in dem einige sinnvolle Einstellungen bereits getroffen wurden, als gezippten Ordner „workspace.zip“.

Installation

1. Entpacken Sie die tar-gz-Datei mit dem JDK an einen Ort Ihrer Wahl. Sie finden darin ein Verzeichnis namens „jdk8uxxxxxx“ (xxxxxx steht wieder für eine Versionsnummer).
2. Entpacken Sie die Datei workspace.zip an einen Ort Ihrer Wahl. Sie finden darin ein Verzeichnis namens „workspace“.
3. Installieren Sie das Programm Eclipse in der üblichen Weise.
4. Starten Sie Eclipse. Um nun in den für den Kurs vorgesehenen Workspace zu wechseln, wählen Sie aus der Menüleiste „File – Switch Workspace – Other“ und navigieren Sie im sich öffnenden Dialog per „Browse“-Knopf zu dem Verzeichnis „workspace“, das Sie aus der Datei workspace.zip extrahiert haben. Bestätigen Sie mit dem „Launch“-Knopf. Meldungen, dass der Workspace mit einer früheren Version von Eclipse erstellt wurde, können Sie ignorieren.
5. Eclipse startet nun neu und Sie sollten sich in „unserem“ Workspace befinden, was

Sie daran erkennen, dass dort bereits ein Projekt „HelloWorld“ angelegt ist.

- Um das heruntergeladene Java8-JDK als Standard für die Ausführung Ihrer Programme einzurichten, muss dieses Eclipse zunächst bekannt gemacht werden. Wählen Sie dazu in der Menüleiste „Eclipse – Preferences“ und suchen Sie dann den Eintrag „Java – Installed JREs“. Hier sollte sich bereits ein Eintrag für das mit Eclipse mitgelieferte JRE befinden. Dieses ist durch das davorstehende Häkchen als Standard gesetzt:



- Klicken Sie nun den Knopf „Add“. Wählen Sie „Standard VM“ und drücken Sie „Next“. Klicken Sie neben dem Feld „JRE home“ auf den Knopf „Directory“. Navigieren Sie dann in Ihrem Verzeichnis „jdk8uxxx“ bis zum Ordner „jdk8uxxx/Contents/Home/jre“, und klicken Sie „Open“. Nach einem kurzen Moment tauchen unter „JRE system libraries“ allerlei Einträge auf. Vergeben Sie einen Namen, z.B. „JDK1.8“ und bestätigen Sie dann mit dem Knopf „Finish“. Setzen Sie anschließend das neu registrierte JDK als Standard, indem Sie das Häkchen entsprechend umsetzen.

8. Ob die Registrierung des Java-8-SDK als Default funktioniert hat, können Sie testen, indem Sie ein neues Projekt anlegen und dort das folgende Programm erstellen und ausführen:

```
public class JavaVersionTest {  
    public static void main(String[] args) {  
        System.out.println(System.getProperty("java.version"));  
    }  
}
```

Sie sollten eine Ausgabe der Art "1.8.0_xxx" erhalten, wobei xxx für eine Unterverversions-Nummer steht und nicht relevant ist.

Erste Schritte mit Java und Eclipse

Nachdem Sie die Arbeitsumgebung nach der obenstehenden Anleitung eingerichtet haben, sollten Sie Eclipse im von uns vorbereiteten Workspace starten können. 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.

Links oben im sog. Package-Explorer sehen Sie 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 <https://feu.mpaap.de/eclipse/filme.html> die Flash-Animation bzw. das Video „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.

Hinweis: Wenn Sie später bei der Bearbeitung des Kurses Programme schreiben, für die Sie *mehrere* Klassen benötigen, legen Sie diese bitte jeweils in eine eigene Java-Datei im gleichen Projekt. Wenn Sie die Klassen so erstellen, wie in „Anlegen eines Projekts, Compilieren, Ausführen“ gezeigt, ergibt sich dies automatisch.