

JDK und TextPad

Der TextPad ist ein ziemlich ausgereifter Text-Editor, den man auch als einfache ("noch durchschaubare") Entwicklungsumgebung z.B. für Java-Programme benutzen kann. Wenn man ihn kommerziell verwendet, sollte man ihn kaufen (eine einzelne Lizenz kostet 27,- US\$ \approx 21,- Euro). Man kann ihn aber beliebig lange "ausprobieren", ohne eine Lizenz zu erwerben. Der TextPad läuft (leider) nur unter Windows.

Um den TextPad (auf einem Rechner im SWE-Labor oder auf Ihrem eigenen Windows-Rechner) als Entwicklungsumgebung für Java einzurichten, sind die folgenden 8 Schritte empfehlenswert (die mit einem * gekennzeichneten Schritte sind im SWE-Labor bereits erledigt).

Schritt 1: Den Java Development Kit (JDK) für Java 8 installieren *.

Schritt 2. Das bin-Verzeichnis des JDK in die Umgebungsvariable PATH eintragen *.

Schritt 3: Die Schritte 1 und 2 kurz überprüfen.

Schritt 4: Den TextPad installieren *.

Schritt 5: Die Installation des TextPads (mit einem Hallo-Programm) überprüfen.

Schritt 6: Die Parameter für **Java kompilieren** ändern

Schritt 7: Die Parameter für **Java-Programm ausführen** ändern.

Schritt 8: Die veränderten TextPad-Kommandos überprüfen

Zu jedem dieser Schritte gibt es im folgenden ein paar Tipps und Hinweise.

Schritt 1: Den Java Development Kit (JDK) für Java 8 installieren *

Um Java-Programme entwickeln zu können, benötigen wir einen JDK von Sun/Oracle. Den JDK für Java 8 (**jdk-8u20**, die update number hinter **u** ist evtl. schon größer) kann man von folgender Seite herunterladen:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>,

Dann sollte man diesen JDK z.B. in den Ordner `C:\Programme\Java8\` installieren.

Anmerkung: Mit dem JDK wird automatisch auch ein **JRE** (Java Runtime Environment) installiert, welches nötig ist, um Java-Programme (nicht nur zu *entwickeln* sondern auch) *auszuführen*. Wenn man nur das JRE herunterlädt und installiert (ohne "den JDK drumherum") kann man Java-Programme zwar ausführen, aber nicht entwickeln.

Schritt 2: Das bin-Verzeichnis des JDK in die Umgebungsvariable PATH eintragen *

Wenn Sie den JDK z.B. in den Ordner `C:\Programme\Java8\` installiert haben, sollten Sie den Pfadnamen `C:\Programme\Java8\bin\` in die Umgebungsvariable PATH eintragen.

Unter Windows 7 geht das so:

Start, Systemsteuerung, System, Erweiterte Systemeinstellungen, Umgebungsvariablen...

Wählen Sie die Systemvariable **PATH**, dann **Bearbeiten**.

Die Variable PATH enthält eine Liste von Pfadnamen, die durch je ein Semikolon voneinander getrennt sind. Tragen Sie ganz am Anfang dieser Liste `C:\Programme\Java8\bin;` (mit einem trennenden Semikolon am Ende) ein. **OK** (im Fensterchen *Systemvariable bearbeiten*). **OK** (im Fenster *Umgebungsvariablen*). **OK** (im Fenster *Systemeigenschaften*).

Wenn diese Kurzbeschreibung nicht ausreicht sollten Sie z.B. "die Umgebungsvariable PATH ändern" googeln.

Schritt 3: Die Schritte 1 und 2 kurz überprüfen.

Öffnen Sie eine Kommando-Eingabe (z.B. durch einen Links-Klick auf **Start**, dann ganz unten "cmd.exe" eingeben, Return) und geben Sie darin folgendes Kommando ein:

```
> javac -version
```

Damit fragen Sie den Java-Compiler `javac.exe` nach seiner Version. Die Antwort sollte etwa so aussehen:

```
> javac 1.8.0_20
```

Wenn Sie eine Fehlermeldung bekommen (z.B. Der Befehl "javac" ist entweder falsch geschrieben oder konnte nicht gefunden werden) sollten Sie die Schritte 1 und 2 überprüfen.

Schritt 4: Den TextPad installieren *.

Laden Sie von der Seite <http://www.textpad.com/> die deutsche Version (**User Interface: German**) herunter und installieren Sie ihn.

So können Sie testen, ob der TextPad sich automatisch "mit dem JDK verbunden" hat:

TextPad starten, dann **Konfiguration, Einstellungen, + Extras** (d.h. Sie sollen auf das + vor **Extras** klicken, *nicht* auf das Wort **Extras!**).

Dadurch sollten unter **Extras** die folgenden 3 (Namen für) Java-Kommandos sichtbar werden:

```
-- Java kompilieren  
-- Java Programm starten  
-- Java Applet starten
```

Wenn das nicht der Fall ist, können Sie den TextPad wie folgt mit dem JDK verbinden (natürlich nur, nachdem Sie den JDK installiert haben):

Konfiguration, Einstellungen, Extras (d.h. Sie sollen auf das Wort **Extras** klicken, nicht auf das + davor), **Hinzufügen, Java SDK-Befehle**

Anmerkung: SDK (Software Development Kit) ist eine ältere Bezeichnung für den **JDK** (Java Development Kit).

Schritt 5: Die Installation des TextPads (mit einem Hallo-Programm) überprüfen.

Besorgen Sie sich ein einfaches Java-Hallo-Programm (z.B. die Datei `Hallo01.java` im Archiv `BspJaSp.zip` auf der Seite <http://public.beuth-hochschule.de/~grude/JavaIstEineSprache/welcome.html>

Öffnen Sie diese Datei mit dem TextPad.

Übergeben Sie die Datei dem Ausführer: **Extras, Benutzer-Programme, Java kompilieren** (oder klicken Sie einfach auf den **Hammer 1** oder drücken Sie **Strg+1**)

Die Meldung `Prozess erfolgreich beendet.` sollte in der Programmausgabe erscheinen.

Lassen Sie das Programm ausführen: **Extras, Benutzer-Programme, Java-Programm starten** (oder klicken Sie einfach auf den **Hammer 2** oder drücken Sie **Strg+2**).

In einem DOS-Ausgabefenster sollte eine Meldung wie

```
Hallo Welt!  
Wie geht es?  
erscheinen.
```

Schritt 6: Die Parameter für "Java kompilieren" ändern

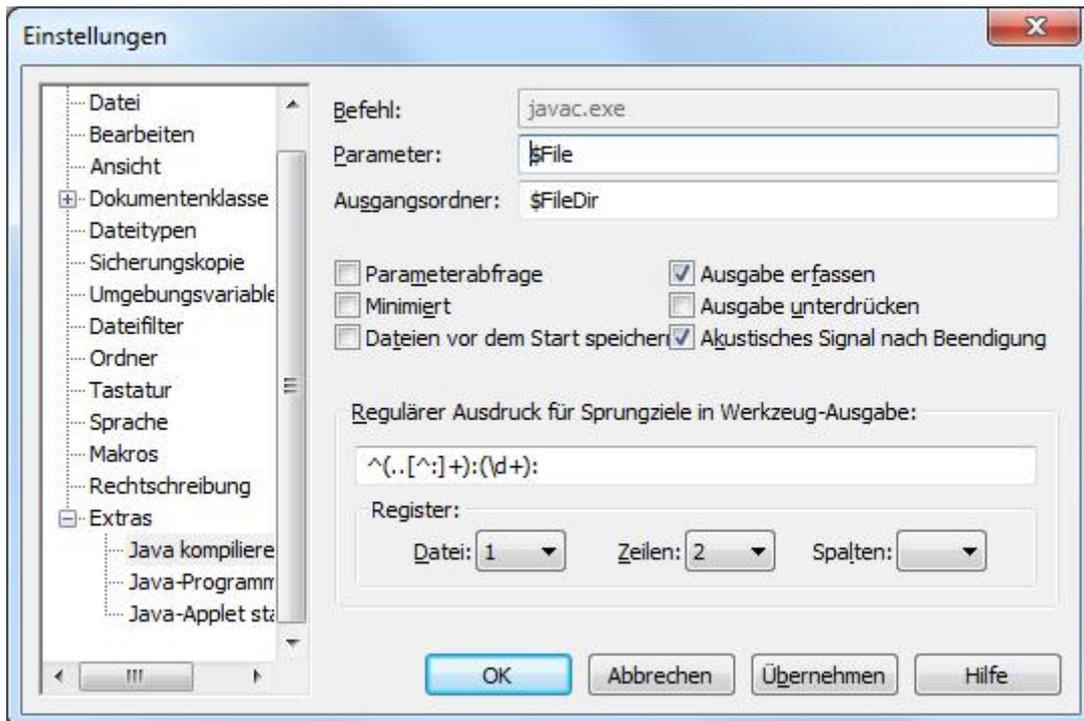
Die Einstellungen, die der TextPad "von Hause aus mitbringt" reichen aus, um einfache Java-Programme zu entwickeln. Um auch höheren Ansprüchen genügen zu können, sollte man einige Einstellungen ändern, z.B. die für das TextPad-Kommando **Java kompilieren**.

Auf einem Rechner im SWE-Labor sollten Sie dazu einen Ordner `z:\klassen` anlegen.

Auf Ihrem eigenen Rechner sollten Sie z.B. einen Ordner `c:\klassen` anlegen und damit alles machen, was hier für den Ordner `z:\klassen` beschrieben wird.

Starten Sie den TextPad, dann **Konfiguration, Einstellungen, + Extras, Java kompilieren**

Dadurch wird ein Einstellungen-Fenster mit Angaben zum **Java kompilieren**-Kommando geöffnet:



Im Eingabefeld **Befehl:** steht, dass der Java-Compiler **javac.exe** aufgerufen wird. Diese Angaben kann man nicht verändern.

Im Eingabefeld **Parameter:** steht `$File`. Dieses Makro bezeichnet den Pfadnamen der aktuellen Datei (z.B. `D:\beispiele\Java\Hallo01.java`). "Aktuell" ist immer die Datei, die der Benutzer zuletzt mit dem TextPad bearbeitet hat oder in die er zuletzt hinein geklickt hat.

Im Eingabefeld **Ausgangsordner:** steht `$FileDir`. Dieses Makro bezeichnet den Pfadnamen des Ordners, in dem die aktuelle Datei steht (z.B. `D:\beispiele\Java\`).

Ersetzen Sie den Inhalt des Eingabefeldes **Parameter** durch folgende Angaben:

```
-d z:\klassen $File
```

Der Java-Compiler `javac.exe` liest normalerweise eine `.java`-Datei ein (z.B. `Hallo01.java`) und gibt eine entsprechende `.class`-Datei (z.B. `Hallo01.class`) aus.

"`-d z:\klassen`" bedeutet: Der Java Compiler soll die von ihm erzeugte `.class`-Datei (nicht in das aktuelle Arbeitsverzeichnis, sondern) in den Ordner `z:\klassen` schreiben.

Schritt 7: Die Parameter für "Java-Programm ausführen" ändern.

Im SWE-Labor ist dieser Schritt nicht nötig (und sollte nicht gemacht werden), da die Labor-Ingenieure (Herr Brüntrup und Herr Shahbaz) schon alle noch nötigen Einstellungen vorgenommen haben.

Auf Ihrem eigenen Rechner müssen Sie noch dafür sorgen, dass der Java-Ausführer (wenn er bestimmte .class-Dateien benötigt) auch an folgenden beiden Stellen danach sucht:

1. Im Ordner `c:\klassen` (in den der Java-Compiler `javac.exe` nach dem Schritt 6 ja alle von ihm erzeugten .class-Dateien ablegt)
2. Im Archiv namens `junit.jar`.

Das Archiv `junit.jar` können Sie sich im SWE-Labor von der Adresse

`c:\Program Files (x86)\java\junit.jar`

z.B. auf einen Stick laden und dann auf Ihrem Rechner irgendwohin kopieren, z.B. an die Stelle

`c:\jars\junit.jar`

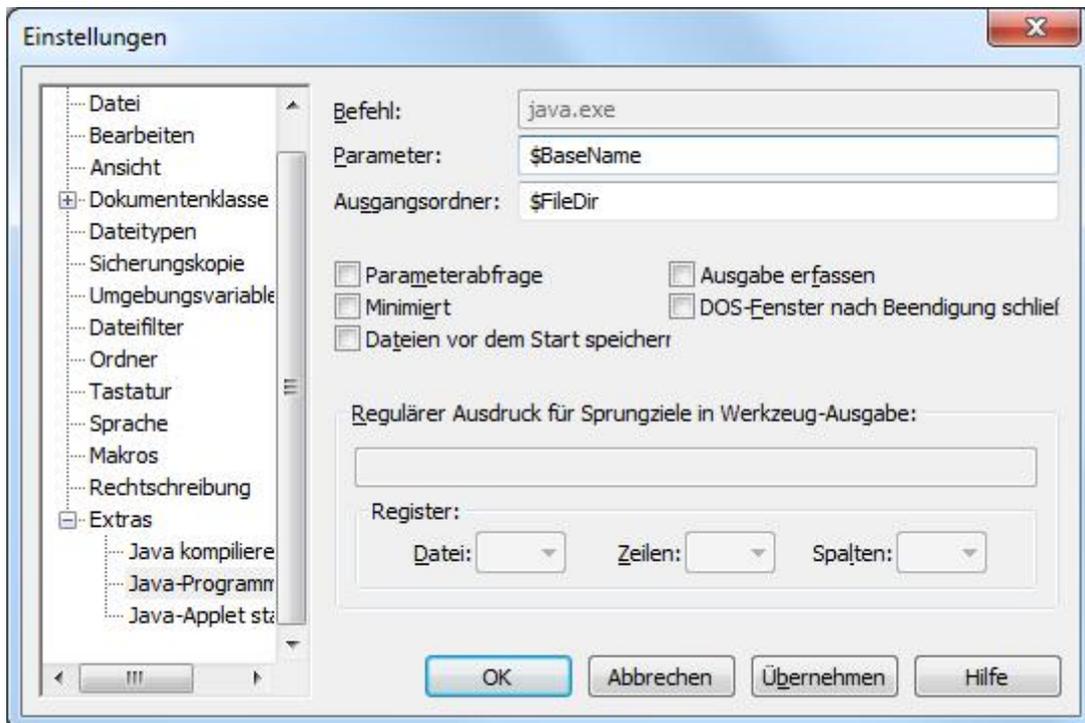
Ab da haben Sie 2 Möglichkeiten:

Möglichkeit 1: Sie tragen die beiden Pfadnamen `c:\klassen` und `c:\jars\junit.jar` in die Umgebungsvariable `CLASSPATH` ein. Das geht ganz ähnlich wie das Eintragen von Pfadnamen in die Umgebungsvariable `PATH` (siehe Schritt 2).

Möglichkeit 2:

Starten Sie den TextPad. Dann **Konfiguration, Einstellungen, + Extras, Java-Programm starten**

Dadurch wird ein Einstellungen-Fenster mit Angaben zum **Java-Programm starten**-Kommando geöffnet:



Im Eingabefeld **Befehl:** steht, dass der Java-Interpreter **java.exe** aufgerufen wird. Diese Angaben kann man nicht verändern.

Im Eingabefeld **Parameter:** steht `$BaseName`. Dieses Makro bezeichnet den Basis-Namen der aktuellen Datei (d.h. den Namen ohne eine Erweiterung wie `.java` oder `.class` etc., z.B. einfach `Hallo01`). "Aktuell" ist immer die Datei, die der Benutzer zuletzt mit dem TextPad bearbeitet hat oder in die er zuletzt hinein geklickt hat.

Im Eingabefeld **Ausgangsordner:** steht `$FileDir`. Dieses Makro bezeichnet den Pfadnamen des Ordners, in dem die aktuelle Datei steht (z.B. `D:\beispiele\Java\`).

Ersetzen Sie den Inhalt des Eingabefeldes **Parameter** durch folgend Angaben:

```
-cp c:\klassen;c:\jars\junit.jar $BaseName
```

"-cp" soll an "Umgebungsvariable CLASSPATH" erinnern.

Der Java-Interpreter `java.exe` erwartet grundsätzlich nur einen Basis-Namen (ohne Erweiterung) und hängt selbst die Erweiterung `.class` daran. Würde man ihm z.B. den Namen `Hallo01.class` übergeben, so würde er nach einer Datei namens `Hallo01.class.class` suchen (und sie höchstwahrscheinlich nicht finden). Deshalb muss als Parameter für den Interpreter `$BaseName` angegeben werden (und nicht `$FileName`, wie im **Schritt 6**).

Schritt 8: Die veränderten TextPad-Kommandos überprüfen

Als erste Prüfung sollten Sie einfach den **Schritt 5** wiederholen. Er sollte ohne Fehlermeldung gelingen und im Ordner `z:\klassen` eine Datei namens `Hallo01.class` hinterlassen.

Für eine "härtere Prüfung" sollten Sie sich das Archiv `BspJaSp.zip` von der Seite <http://public.beuth-hochschule.de/~grude/JavaIstEineSprache/welcome.html>

holen und in einen Ordner namens `BspJaSp` entpacken.

Kopieren Sie aus dem Ordner `BspJaSp` die beiden Dateien `EM.java` und `Hallo08.java` in zwei verschiedene Ordner (z.B. `Ord1` und `Ord2`, wo die Ordner stehen ist egal).

Öffnen Sie die beiden `.java`-Dateien (in den verschiedenen Ordnern) beide mit dem TextPad. Kompilieren Sie zuerst `EM.java`.

Wenn Sie danach `Hallo08.java` erfolgreich kompilieren und ausführen können, haben Sie bei den Schritten 6 und 7 offenbar etwas richtig gemacht (vielleicht sogar alles :-).

Die Dokumentation zu Java 8 findet man auf der Seite:

<http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html>

Von dieser Seite sollten Sie 2 Dateien herunterladen:

Die Dokumentation der Java-8-Standardbibliothek (Datei **jda-8u20-docs-all.zip**, ca. 85 MB) und die Dokumentation von JavaFX-8 (Datei **javafx-8u20-apidocs.zip**, ca. 9 MB).

