

Installation, Konfiguration und Dokumentation

Das Java Development Kit installieren

Die Installationsdateien

Im Folgenden wird die Installation des J2SE Development Kit 9.0, kurz als JDK 9.0 bezeichnet, unter Windows 10 und Linux gezeigt.

Für die Installation des JDK 9.0 benötigen Sie die entsprechende Installationsdatei für das jeweilige Betriebssystem, die Sie unter der URL

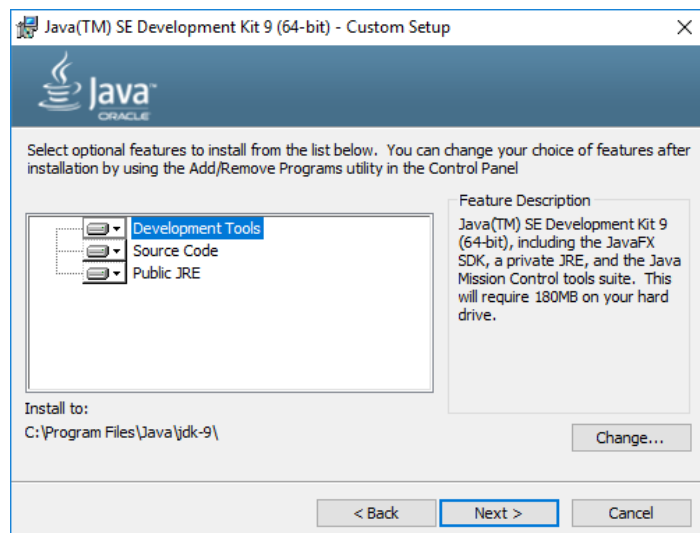
<http://www.oracle.com/technetwork/java/javase/downloads/jdk9-downloads-3848520.html>

downloaden können. Bevor Sie die Dateien herunterladen können, müssen Sie den Lizenzbestimmungen zustimmen.

JDK 9 (x64)	Installationsdatei	Größe
Installationsdateien des Java™ SE 9 JDK für Windows	<code>jdk-9_windows-x64_bin.exe</code>	ca. 358 MB
Installationsdateien des Java™ SE 9 JDK für Linux	<code>jdk-9_linux-x64_bin.tar.gz</code>	ca. 330 MB

Das JDK 9 unter Windows 10 installieren

- ▶ Starten Sie das Installationsprogramm.
- ▶ Deaktivieren Sie im folgenden Dialogfenster die Komponenten, die Sie nicht installieren möchten. Es wird empfohlen, wie vorgegeben alle Komponenten zu installieren:
 - ✓ die eigentliche Entwicklungsumgebung und die integrierte Laufzeitumgebung zur Ausführung von Java-Programmen,
 - ✓ Beispiele zur Demonstration (Quellcode),
 - ✓ einen Großteil des Quellcodes von Klassen der Entwicklungsumgebung.



Komponenten und Installationsverzeichnis festlegen

- ▶ Betätigen Sie die Schaltfläche *Change*, falls Sie das Verzeichnis ändern möchten. Standardmäßig erfolgt die Installation in dem folgenden Verzeichnis:
C:\Programme\Java\jdk-9
- ▶ Betätigen Sie die Schaltfläche *Next*, um die ausgewählten Komponenten zu installieren.
- ▶ Beenden Sie die Installation mit einem Klick auf die Schaltfläche *Close*.

Die Laufzeitumgebung installieren

Eine Laufzeitumgebung Java™ SE Runtime Environment 9 (JRE 9) ist bereits in der Entwicklungsumgebung integriert. Daher benötigen Sie das *Public JRE* nicht zwingend für die Entwicklung (das Erstellen und Ausführen) von Java-Programmen. Wenn Sie die Laufzeitumgebung jedoch zusammen mit Ihren Programmen weitergeben möchten, benötigen Sie aus lizenzrechtlichen Gründen diese spezielle Laufzeitumgebung (*Public JRE*).

Bei dieser Installation können Sie ebenfalls das Installationsverzeichnis anpassen.

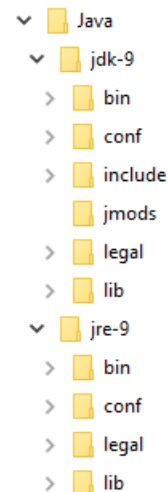
- ▶ Aktivieren Sie das Kontrollkästchen *Zielordner ändern* und betätigen Sie die Schaltfläche *Installieren*, um die Installation der Laufzeitumgebung zu starten.
- ▶ Über die Schaltfläche *Ändern* können Sie im nächsten Dialogfenster einen anderen Zielordner bestimmen. Starten Sie die Installation über die Schaltfläche *Weiter*.
- ▶ Betätigen Sie nach erfolgter Installation die Schaltfläche *Ende*.

Unter Windows 10 benötigen Sie Administratorrechte, um die Installation durchzuführen.

Die Verzeichnisstruktur

Bei der Installation wird im Verzeichnis *c:\Programme* die abgebildete Ordnerstruktur angelegt:

- ✓ Im Ordner *jdk-9* befinden sich neben den Unterordnern Readme-Dateien.
- ✓ Der Unterordner *bin* enthält die Programmdateien.
- ✓ Der Unterordner *demo* enthält – wenn installiert – Beispiel-Java-Programme.
- ✓ Der Unterordner *lib* enthält zusätzliche Bibliotheksdateien und Quellcodedateien zum JDK in gepackter Form (*src.zip*).
- ✓ Der Ordner *jre9* enthält die Laufzeitumgebung.



Verzeichnisstruktur zu Java 9

Das JDK unter Linux installieren

Beachten Sie, dass sich die Installation auf unterschiedlichen Linux-Distributionen anders gestalten kann.

- ▶ Wechseln Sie in das Verzeichnis, in dem Sie das JDK installieren möchten (z. B. in `/usr/local` oder in das eigene HOME-Verzeichnis): Achten Sie darauf, dass Sie die benötigten Rechte besitzen, um das JDK in diesem Verzeichnis zu installieren.

```
cd /usr/local
```

- ▶ Kopieren bzw. verschieben Sie die Installationsdatei (selbstextrahierende Binärdatei) in dieses Verzeichnis, z. B. über die folgenden Befehle:

```
mv /usr/install/jdk-9_linux-x64_bin.tar.gz.
```

Der Punkt am Ende gehört zum Befehl und bedeutet, dass als Zielverzeichnis das aktuelle Verzeichnis (hier `/usr/local`) verwendet wird. Alternativ können Sie den vollständigen Pfad angeben.

Für die Ausführung des Befehls `mv` benötigen Sie die entsprechende Berechtigung.

- ▶ Setzen Sie gegebenenfalls die Ausführungsrechte der Datei über die folgenden Befehle:

```
chmod 544 jdk-9-linux-i586.bin
```

Beim Auflisten der Dateien des aktuellen Verzeichnisses (oder des Verzeichnisses, in dem sich die Installationsdatei befindet) über den Befehl

```
ls -la
```

müssen die Dateirechte jetzt folgendermaßen angezeigt werden:

```
-r-xr--r-- jdk-9-linux-i586.bin
```

- ▶ Das JDK 9 wird nicht installiert, sondern in das Unterverzeichnis `jdk-9` entpackt. Es werden keine Lizenzbestimmungen angezeigt. Führen Sie den folgenden Befehl aus:

```
tar xzf jdk-9-linux-i586.tar.gz
```

- ▶ Löschen Sie die Installationsdatei, wenn Sie sie nicht mehr benötigen, z. B. über die folgenden Befehle:

```
rm jdk-9-linux-i586.tar.gz
```

- ✓ Unter Linux wird zusätzlich zum JDK das JRE im Verzeichnis `./jdk-9/jre` installiert. Während der Installation des JDK wird die Installation des JRE nicht separat angeboten und auch nicht an einem weiteren Ort installiert, wie das unter Windows der Fall ist.
- ✓ Wenn Sie eine Java-Anwendung weitergeben möchten, müssen Sie das JRE nicht zusätzlich downloaden. Ein separater Download des JRE für die Anwendungsweitergabe hat aber den Vorteil, dass das JRE über ein Installationsprogramm installiert werden kann. Ansonsten müssen Sie dies manuell tun.
- ✓ Eine weitere Installationsform für das JDK ist die Verwendung von RPM-Paketen. Diese wird hier nicht näher beschrieben.
- ✓ Installieren Sie das JDK nicht über eine bereits vorhandene Version.

Das JDK konfigurieren



Die Systemvariablen **PATH** und **CLASSPATH**

Für die Vervollständigung Ihrer Installation sind noch zwei Dinge zu tun.

- ✓ Die `PATH`-Variable ist zu setzen. Die Betriebssysteme suchen jeweils in den Verzeichnissen, die in der `PATH`-Angabe angegeben sind, nach Programmen, die Sie ohne Pfadangabe in der Kommandozeile verwenden, z. B. `javac`. Wenn Sie die `PATH`-Variable nicht setzen, müssen Sie immer den vollständigen Pfadnamen angeben:
z. B. `c:\Programme\Java\jdk-9\bin\javac`
- ✓ Die `CLASSPATH`-Variable legt fest, in welchem Verzeichnis der Compiler und der Interpreter nach `*.java`-Dateien (`*.java/*.class`) suchen sollen. Ist der Classpath nicht angegeben, erfolgt die Suche im aktuellen Verzeichnis.

JDK unter Windows 10 konfigurieren

Falls Sie das JDK nicht wie standardmäßig vorgegeben im Verzeichnis `c:\Programme` installiert haben, passen Sie in den folgenden Beschreibungen den Verzeichnisnamen entsprechend an.

- ▶ Öffnen Sie den Systemsteuerung über   - *Systemsteuerung*.
- ▶ Klicken Sie auf die Kategorie *System*.
- ▶ Wählen Sie *Info, Systeminfo* und dann *Erweiterte Systemeinstellungen*.
- ▶ Wechseln Sie in das Register *Erweitert* und klicken Sie auf die Schaltfläche *Umgebungsvariablen*.
- ▶ Klicken Sie doppelt in der Liste *Benutzervariablen für <Benutzername>* auf den Eintrag *Path*.
oder Wählen Sie den Eintrag *Path* und betätigen Sie die Schaltfläche *Bearbeiten*.
Fügen Sie dem aktuellen Wert der `PATH`-Variablen den folgenden Text hinzu:
`; C:\Programme\Java\jdk-9\bin`
Mehrere Einträge trennen Sie durch ein Semikolon.
- ▶ Bestätigen Sie Ihre Angaben zweimal mit *OK*.

oder

- ▶ Sollte kein Eintrag *Path* in der Liste vorhanden sein, klicken Sie auf die Schaltfläche *Neu*.
- ▶ Geben Sie im Eingabefeld *Name der Variablen* den Text *Path* und im Eingabefeld *Wert der Variablen* den folgenden Text ein:
`; C:\Programme\Java\jdk-9\bin`
- ▶ Bestätigen Sie Ihre Angaben zweimal mit *OK*.

Unter Windows 10 müssen Sie Ihr System nicht neu starten. Die Änderungen werden sofort für jede neue geöffnete Konsole verwendet.

Die `PATH`-Variable in der Konsole festlegen

Wenn Sie in der Konsole arbeiten, setzen Sie die `PATH`-Variable folgendermaßen:

- ▶ Geben Sie in der Konsole den folgenden Befehl ein:

```
set Path=%Path%;C:\Programme\Java\jdk-9\bin
```



Beachten Sie, dass diese Einstellung nur für die aktuelle Konsole gültig ist.

JDK unter Linux konfigurieren

In den folgenden Verzeichnisangaben wie beispielsweise `/usr/local/jdk-9/bin` ersetzen Sie `/usr/local` durch die entsprechenden Verzeichnisnamen Ihrer JDK-Installation.

- ▶ Um Ihre `PATH`-Variable zu ändern (diese befindet sich in Ihrer Profildatei), wechseln Sie in Ihr `HOME`-Verzeichnis (z. B. über `cd`).
- ▶ Geben Sie den Befehl


```
ls -la
```

 ein, um die Dateien in diesem Verzeichnis anzuzeigen.
- ▶ Suchen Sie nach einer Datei, die den Text `profile` beinhaltet. Diese enthält spezielle Startinformationen Ihrer Shell (Kommandozeile). Bei der Bash-Shell heißt sie z. B. `.profile` oder `.bash_profile` (mit einem Punkt vor dem Dateinamen). In einigen Fällen enthält diese Datei einen weiteren Verweis auf Ihre Profildatei, z. B. `.bashrc`.
- ▶ Öffnen Sie diese Datei in einem beliebigen Texteditor (`joe`, `vi`, `emacs`).
- ▶ Fügen Sie einer bereits vorhandenen `PATH`-Variablen den folgenden Eintrag hinzu:


```
/usr/local/jdk-9/bin
```

 Mehrere Einträge werden durch einen Doppelpunkt getrennt.
- ▶ Ist kein `PATH`-Eintrag vorhanden (was eigentlich darauf schließen lässt, dass Sie eventuell die falsche Profildatei bearbeiten), können Sie einen Eintrag hinzufügen.


```
PATH=$PATH:/usr/local/jdk-9/bin
```
- ▶ Testen Sie die `PATH`-Variable über das Kommando `which java`. Sie müssen dazu eine neue Shell öffnen, da die Änderung der Profildatei nicht sofort übernommen wird.
- ▶ Im Erfolgsfall sollte der vollständige Pfad zum Aufruf des Programms `java` angezeigt werden. Ansonsten erhalten Sie eine Fehlermeldung


```
which: no java in (... Verzeichnisangaben der PATH-Variablen...).
```

`CLASSPATH` unter Windows und Linux

Beispielsweise beim Ausführen und Kompilieren von Java-Programmen benötigen die jeweiligen Programme Zugriff auf die entsprechenden `*.class`-Dateien. Wo sucht beispielsweise der Compiler nach `*.class`-Dateien? Standardmäßig erfolgt die Suche in den im JDK enthaltenen Verzeichnissen und im aktuellen Verzeichnis.

Mit der Variablen `CLASSPATH` können Sie individuell ein Verzeichnis festlegen. Die über das JDK installierten `*.class`-Dateien werden immer automatisch gefunden, sodass dieser Pfad nicht in den `CLASSPATH` aufgenommen werden muss. Soll die Suche aber weiterhin auch im aktuellen Verzeichnis erfolgen, so müssen Sie die Formulierung für das aktuelle Verzeichnis ebenfalls dem `CLASSPATH` hinzufügen.

Die nachfolgende Beschreibung zeigt, wie Sie das aktuelle Verzeichnis und das Verzeichnis der Übungsdateien in den `CLASSPATH` aufnehmen:

Betriebssystem	CLASSPATH	Erläuterung
Windows	<code>. ; c:\uebung\jav9_bu</code>	Der führende Punkt kennzeichnet das aktuelle Verzeichnis. Mehrere Pfadangaben trennen Sie durch ein Semikolon.
Linux	<code>./home/meinname/Uebung/jav9_bu</code>	Der führende Punkt kennzeichnet auch hier das aktuelle Verzeichnis. Mehrere Pfadangaben sind durch einen Doppelpunkt zu trennen.

- ✓ Das Bearbeiten/Erstellen der `CLASSPATH`-Variablen erfolgt wie das Bearbeiten der `PATH`-Variablen.
- ✓ In diesem Buch wird der Name `CLASSPATH` als Name für den Pfad zu den benötigten Java-Klassen verwendet. Wenn sich Einstellungen auf die `CLASSPATH`-Variable beziehen, wird dies explizit angegeben.

Den Editor TextPad installieren und konfigurieren

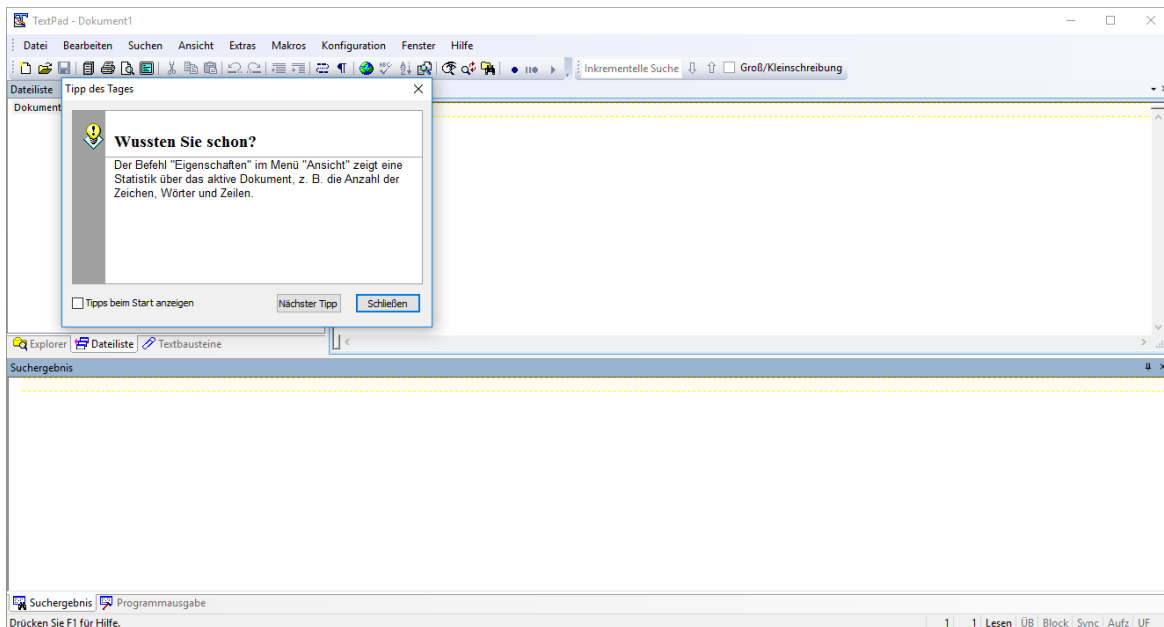
Arbeiten mit einem Texteditor

Um den Java-Quellcode zu schreiben, benötigen Sie einen Texteditor. Texteditoren sind für alle Computer-Plattformen verfügbar, besitzen aber unterschiedliche Funktionen. Ein Editor unterscheidet sich von einer Textverarbeitung oder einem Layout-Programm dadurch, dass er keine – normalerweise unsichtbaren – Formatierungsanweisungen in den Text einfügt, um ihn in Absätze, Listen, Tabellen usw. zu gliedern.

Bei der Erstellung des vorliegenden Buches wurde mit dem Texteditor TextPad Version 7.6.4 (<http://www.textpad.com/>) gearbeitet. Der Texteditor TextPad ist als Shareware erhältlich und leicht zu bedienen. Durch die Verwendung von Dokumentenklassen erkennt TextPad an der Dateinamenerweiterung `*.java` einen Java-Quellcode und wendet zur Darstellung entsprechende Einstellungen an. TextPad erleichtert die Lesbarkeit von Java-Quellcode, da Schlüsselwörter farbig hervorgehoben und fett formatiert werden. Ein automatisches Einrücken der Programmzeilen lässt die Struktur besser erkennen.

Download und Installation von TextPad

- ▶ Laden Sie das Programm TextPad von der Webseite <http://www.textpad.com>.
- ▶ Um TextPad zu installieren, folgen Sie den Schritten des Installationsassistenten.
- ▶ Starten Sie nach erfolgreicher Installation das Programm.



Konfiguration von TextPad

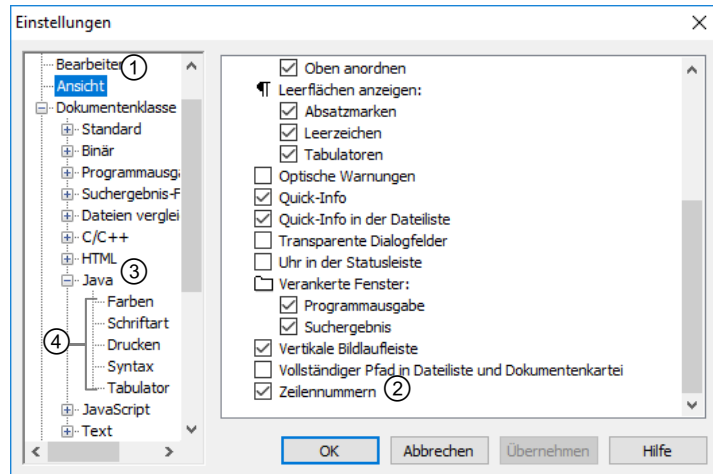
Ein Merkmal des Texteditors TextPad ist die Arbeit mit sogenannten Dokumentenklassen. In einer Dokumentenklasse wird für eine bestimmte Gruppe von Dateien eine Reihe von Attributen definiert. Über die Dateinamenerweiterung und mithilfe von Platzhalterzeichen (Wildcards) kann festgelegt werden, welche Datei zu einer Gruppe gehört (z. B. **.java*). Attribute, die für jede Dokumentenklasse definiert werden können, sind beispielsweise Tabstops, Schriftarten, die Seiteneinrichtung für den Ausdruck und die farbige Syntax-Hervorhebung.

Standardmäßig werden bei der Installation von TextPad automatisch fünf Dokumentenklassen definiert: *Text* (reine Textdateien), *Binärdateien*, *C/C++*, *HTML* und *Java*. Für alle anderen Dateien, die nicht zu einer der festgelegten Dokumentenklasse gehören, gelten die Attribute der Klasse *Standard*.

Einstellungen verändern

Über das Menü *Konfiguration - Einstellungen* können Sie Einstellungen vornehmen, die die Bearbeitung von Java-Quellcode mit dem Texteditor erleichtern:

- ▶ Wählen Sie im Bereich ① die Kategorie *Ansicht*.
In den Einstellungen für die Ansicht können Sie beispielsweise im unteren Bereich die Anzeige der *Zeilennummern* ② einschalten.
- ▶ Wählen Sie die Dokumentenklasse *Java* ③.
In den Einstellungen für die Dokumentenklasse *Java* bzw. in den Unterbereichen ④ können Sie die Darstellung von Java-Quellcode verändern.



Einstellung für die Kategorie ANSICHT in TextPad

- ▶ Bestätigen Sie die Eingaben mit *OK*.

Standarddateityp für TextPad festlegen

Im Menü *Konfiguration - Einstellungen* können Sie festlegen, dass TextPad standardmäßig die Dateinamenerweiterung *.java* verwendet und somit dann auch die Dokumentenklasse *Java* zur Syntaxhervorhebung wählt.

- ▶ Wählen Sie die Kategorie *Datei*.
- ▶ Geben Sie im Eingabefeld *Standard-Dateierweiterung* *java* ein. Geben Sie keinen Punkt ein.
- ▶ Bestätigen Sie die Eingaben mit *OK*.

Einige Einstellungen werden erst nach einem Neustart von TextPad wirksam.

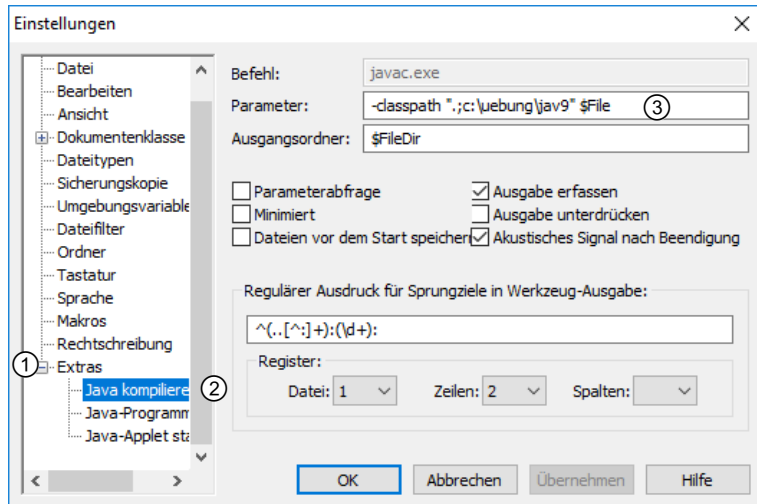
Die CLASSPATH-Variable in TextPad festlegen

Zum Kompilieren und zur Ausführung wird die Angabe benötigt, in welchem Verzeichnis nach weiteren erforderlichen Java-Dateien zu suchen ist. Mit der Variablen CLASSPATH können Sie dieses Verzeichnis festlegen. Falls Sie keinen CLASSPATH angeben, wird das aktuelle Verzeichnis verwendet. Die Classpath-Variable können Sie in TextPad vorgeben.

- ▶ Rufen Sie den Menüpunkt *Konfiguration - Einstellungen* auf.

- ▶ Wählen Sie im Bereich ① die Kategorie *Extras* und den Untereintrag *Java kompilieren* ②.

Im Eingabefeld *Parameter* ist standardmäßig das Makro *\$File* vorgegeben. Es trägt beim Ausführen des Kompilier-Befehls jeweils den kompletten Dateinamen der geöffneten Datei als Parameter ein.



Einstellung in TextPad

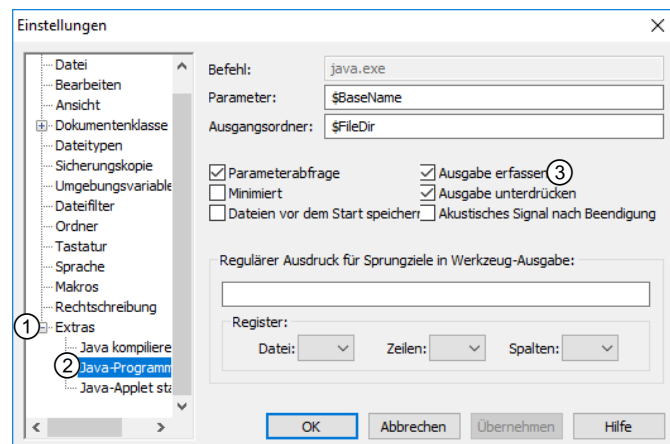
- ▶ Geben Sie im Eingabefeld *Parameter* vor dem Eintrag *\$FILE* den Text *-classpath* und anschließend den Pfad zu dem gewünschten Verzeichnis und ein Leerzeichen ein ③.
- ▶ Schließen Sie die Konfiguration mit *OK* ab.
- ▶ Starten Sie TextPad neu, um die Änderungen wirksam werden zu lassen.

Die Ausgabe umleiten

Wenn Sie Anwendungen mit Java programmieren, die eine Ausgabe über die Konsole vornehmen, dann wird beim Ausführen des Programms ein Konsolenfenster für die Ausgabe geöffnet. Sie haben in TextPad die Möglichkeit, die Ausgabe in ein Fenster innerhalb von TextPad umzuleiten.

- ▶ Rufen Sie den Menüpunkt *Konfiguration - Einstellungen* auf.
- ▶ Wählen Sie im Bereich ① die Kategorie *Extras* und den Untereintrag *Java-Programme starten* ②.

- ▶ Aktivieren Sie das Kontrollfeld *Ausgabe erfassen* ③.
Das Kontrollfeld *Ausgabe unterdrücken* wird eingeblendet und automatisch aktiviert.
- ▶ Schließen Sie die Konfiguration mit *OK* ab.



Die Konsolenausgabe umleiten

Die Ausgabe erfolgt nun direkt in TextPad im Programmausgabefenster ④.

The screenshot shows the TextPad application window. The main editor displays the following Java code:

```

1 class Geometry
2 {
3     public static void main(String[] args)
4     {
5         //Ein Rechteck mit der Breite 1 und der Laenge 1 erzeugen
6         Rectangle specialRect = new Rectangle();
7
8         //Ein Rechteck mit der Breite 3 und der Laenge 5 erzeugen
9         Rectangle anotherRect = new Rectangle(3, 5);
10
11        System.out.printf("Das Rechteck (%d x %d)\n",
12                          specialRect.getWidth(), specialRect.getLength());
13        System.out.printf("hat den Flächeninhalt %d\n",
14                          specialRect.getArea());
15        System.out.printf("Das Rechteck (%d x %d)\n",
16                          anotherRect.getWidth(), anotherRect.getLength());
17        System.out.printf("hat den Flächeninhalt %d\n",
18                          anotherRect.getArea());
19    }
20 }
21

```

The 'Programmausgabe' (Program Output) window at the bottom shows the following output:

```

Das Rechteck (1 x 1)
hat den Flächeninhalt 1
Das Rechteck (3 x 5)
hat den Flächeninhalt 15

```

The output is circled with a blue circle and the number 4, indicating it is the focus of the text.

Die umgeleitete Konsolenausgabe in TextPad

- ! Falls das von Ihnen erstellte Programm eine Tastatureingabe über die Standardeingabe erfordert, müssen Sie die hier beschriebene Ausgabeumleitung ausschalten, indem Sie das Kontrollfeld ③ deaktivieren.

Die Dokumentation zu Java nutzen

Die Dokumentation zu Java einrichten

Zur Programmiersprache Java stellt Oracle als Referenz eine umfangreiche HTML-basierte Dokumentation zur Verfügung. Diese steht komplett im Internet zur Verfügung. Alternativ können Sie die Dokumentationsdateien lokal installieren. Dafür müssen Sie diese separat downloaden und einrichten. Die Dokumentationsdateien liegen in gepackter Form (ZIP) vor und sind für Windows und Linux gleich.

Unter der Adresse <http://www.oracle.com/technetwork/java/javase/documentation/jdk9-doc-downloads-3850606.html> finden Sie den Link zur Downloaddatei. Die Größe beträgt ca. 67 MB.

Entpacken Sie die Datei in ein Verzeichnis, zum Beispiel `c:\Program Files\Java\jdk-9docs`. Die Dokumentation befindet sich anschließend im Unterverzeichnis `docs`. Zum Öffnen rufen Sie die Datei `index.html` auf.

Alternativ können Sie die im Internet unter der folgenden Adresse verfügbare Dokumentation nutzen:

<http://docs.oracle.com/javase/9/index.html>

Die Dokumentation zum JDK 9.0 öffnen

Ausschnitt der Startseite der Dokumentation zum JDK 9.0

- ▶ Öffnen Sie die HTML-Datei *index.htm* über die Webadresse oder in dem Ordner, in dem Sie die Dokumentation installiert haben.
Die Startseite der Dokumentation wird angezeigt.
- ▶ Um Informationen zu den Packages, Klassen, Methoden und Attributen zu erhalten, klicken Sie auf der Startseite auf den Hyperlink *API Documentation* ① (API = Application Programming Interface).

Mit der Java-API-Dokumentation arbeiten

Nach dem Öffnen können die Informationen der Seite unterteilt in Bereiche (*Frames* ①) oder im Listenformat (*No Frames* ②) angezeigt werden.

The screenshot displays the Java SE 9 & JDK 9 API documentation. The left sidebar contains a navigation menu with 'All Packages' (3) and 'All Classes'. The main content area is titled 'Java Platform, Standard Edition & Java Development Kit Version 9 API Specification'. It includes a search bar and navigation tabs for 'FRAMES' (1) and 'NO FRAMES' (2). The 'FRAMES' view shows a table of modules with their descriptions.

Module	Description
java.activation	Defines the JavaBeans Activation Framework (JAF) API.
java.base	Defines the foundational APIs of the Java SE Platform.
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
java.corba	Defines the Java binding of the OMG CORBA APIs, and the RMI-IIOP API.
java.datatransfer	Defines the API for transferring data between and within applications.

Darstellung im Modus *Frames*

- ▶ In der Frameansicht können Sie im linken oberen Fenster zwischen *All Packages* und *All Modules* wechseln ③. Die Möglichkeit der Darstellung nach Modulen beruht auf dem neu in Java 9 eingeführten Modulkonzept.
- ▶ Klicken Sie auf den Namen des Moduls/Packages, zu dem Sie Informationen benötigen. Wechseln Sie beispielsweise zu *All Packages* und blättern Sie nach unten und klicken Sie auf *java.lang*.

In der anschließend eingeblendeten Übersicht finden Sie beispielsweise alle in dem Package enthaltenen Klassen.

- ▶ Klicken Sie auf einen Hyperlink, um z. B. Informationen zu einer Klasse anzeigen zu lassen.

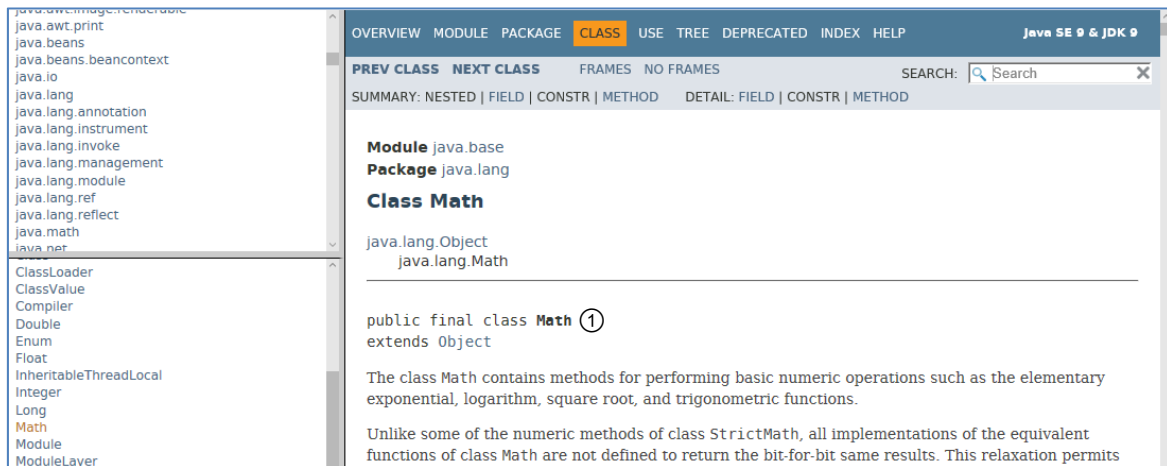
Beispiel

Im Abschnitt *Class Summary* finden Sie beispielsweise die Klasse `Math`, die im Package `java.lang` enthalten ist. Wenn Sie auf den Hyperlink *Math* klicken, werden alle Informationen zur Klasse `Math` angezeigt.

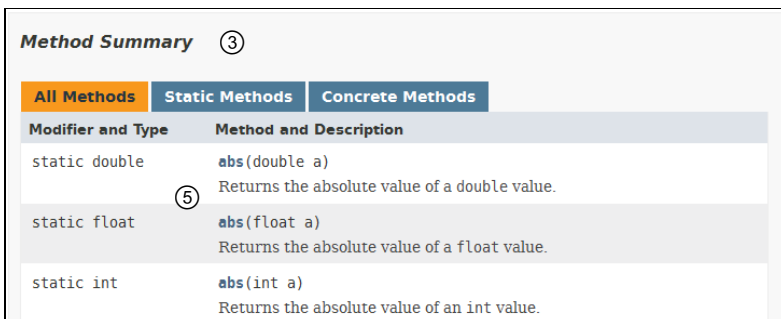
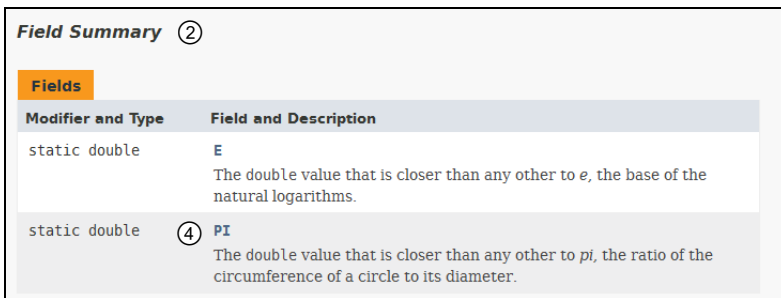
Alternative Navigationsmöglichkeiten

- ✓ Unabhängig davon, welche Informationen im rechten Frame angezeigt werden, können Sie über die Hyperlinks im linken oberen Frame jederzeit zu einem anderen Package/Module wechseln. Im linken unteren Frame werden dazu dann die Inhalte aufgelistet.
- ✓ Über den linken unteren Frame wählen Sie eine Klasse dieses Packages/Moduls aus. Die Informationen zu der Klasse werden dann im rechten Frame angezeigt.

Informationen zu einer Klasse



Nach einer grundlegenden Beschreibung der Klasse ① ist die Seite in mehrere Abschnitte gegliedert. Zwei dieser Bereiche sind *Field Summary* und *Method Summary*.



- ✓ Unter Field Summary ② finden Sie die Erläuterung der Attribute (Fields) der Klasse.
- ✓ Der Abschnitt Method Summary ③ stellt die Methoden mit einer Kurzbeschreibung vor.
- ✓ Die Namen sind jeweils als Hyperlinks eingefügt ④/⑤.

- Klicken Sie auf einen Hyperlink ④/⑤, erhalten Sie Detailinformationen.

Die nebenstehende Abbildung zeigt Informationen zur Methode `sin`, mit der Sie den Sinus eines Winkels berechnen können.

Method Detail

sin

```
public static double sin(double a)
```

Returns the trigonometric sine of an angle. Special cases:

- If the argument is NaN or an infinity, then the result is NaN.
- If the argument is zero, then the result is a zero with the same sign as the argument.

The computed result must be within 1 ulp of the exact result. Results must be semi-monotonic.

Parameters:

a - an angle, in radians.

Returns:

the sine of the argument.

Weitere Möglichkeiten

Index

Über den Hyperlink *Index* im oberen oder unteren Bereich der Seite (im rechten Frame) blenden Sie eine alphabetische Übersicht ein.

Java SE 9 & JDK 9

OVERVIEW MODULE PACKAGE CLASS USE TREE DEPRECATED **INDEX** HELP

PREV LETTER NEXT LETTER FRAMES NO FRAMES SEARCH: Search

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z _

A

- a - Variable in class java.awt.AWTEventMulticaster
A variable in the event chain (listener-a)
- A - javafx.scene.input.KeyCode
Constant for the A key.
- A - Static variable in class java.awt.PageAttributes.MediaType
The MediaType instance for Engineering A, 8 1/2 x 11 in.
- A - Static variable in class javax.print.attribute.standard.MediaSize.Engineering
Specifies the engineering A size, 8.5 inch by 11 inch.
- A - Static variable in class javax.print.attribute.standard.MediaSizeName
A size .
- A - Static variable in class javax.swing.text.html.HTML.Tag
Tag <a>

Die Übersicht enthält alle Elemente wie beispielsweise Klassen, Attribute und Methoden in alphabetischer Reihenfolge.

Tree

Über den Hyperlink *Tree* blenden Sie eine Übersicht ein, die die Klassen in ihrer Zugehörigkeit zu den Packages hierarchisch geordnet auflistet.

Nach Informationen suchen

Neben den Navigationsmöglichkeiten können Sie auch die Suchfunktion Ihres Browsers nutzen, um die gewünschten Informationen zu finden. Auf diese Weise können Sie auch innerhalb der Erläuterungstexte suchen.