

Java Grundlagen

Einstieg in die Grundlagen der Programmierung



[1]

Stand: **10.09.2019**
Autor: **M. Völkl**

1. Grundbegriffe der Programmierung



Abbildung: Island of Java [5]

Eine Programmiersprache ist eine Sprache, die zur Formulierung von Computerprogrammen verwendet wird. Die Anweisungen (Befehle) können dabei entweder in Maschinensprache geschrieben sein, der vom Prozessor ausgeführt wird, oder in Form eines abstrakteren, für Menschen besser lesbaren Quelltextes angegeben, der automatisiert in Maschinencode übersetzt werden kann.

Programmiersprachen sollen also die Programmierung nicht nur ermöglichen, sondern auch erleichtern.

Die Maschinensprache wird in der Regel von einem Assembler (aus Assemblercode) oder Compiler (aus dem Quelltext einer höheren Programmiersprache) generiert. Direkt in Maschinensprache wird praktisch nicht (mehr) programmiert. Im Gegensatz zum Programmcode der Assemblersprache oder von Hochsprachen ist der binäre Maschinencode für den

Menschen kaum lesbar – außer für Experten, die ihn byteweise interpretieren können und dabei meist spezielle Programme benutzen. Fast ausschließlich wird der Maschinencode dabei binär oder hexadezimal dargestellt – zum Beispiel in sog. Dumps, die den Inhalt des Hauptspeichers zeigen, wenn ein Programm „abgestürzt“ ist und der Programmierer mit diesem Hilfsmittel die Fehlerursache analysieren will.

Immer mehr maschinennahe Programmierung – die Domäne des Assemblers – kann heute fast vollständig durch höhere Programmiersprachen abgedeckt werden. Auch steht der Möglichkeit der Erstellung effizienter Programme die erschwerte Wartbarkeit von Assemblerprogrammen gegenüber. Für optimalen Code wird immer mehr Kontextwissen benötigt.

Auf der anderen Seite benutzen die meisten Compiler für höhere Programmiersprachen nur einen kleinen Teil des Befehlssatzes der CPU, während dem Assemblerprogrammierer der vollständige Befehlssatz zur Verfügung steht, sodass er in manchen Situationen effizientere Befehle benutzen kann, die dem reinen Hochsprachenprogrammierer nicht zugänglich sind. Die Anwendung kann sich dann auf die Situationen beschränken, in denen es aus funktionalen oder Effizienzgründen notwendig oder nützlich ist, maschinennah zu programmieren.

Eine höhere Programmiersprache ist dabei eine, die die Abfassung eines Computerprogramms in einer abstrakten Sprache ermöglicht (die so zwar für Menschen, aber nicht unmittelbar für Computer verständlich ist).

Fortran gilt als die erste jemals tatsächlich realisierte höhere Programmiersprache.

Die ersten höheren Sprachen wurden zuerst in Assemblersprachen bzw. Maschinencode übersetzt, um dann ausgeführt werden zu können. Einige modernere Sprachen werden heute erst in weniger hohe Sprachen übersetzt, aus denen selbst wiederum relativ effizienter Maschinencode gewonnen werden kann. Die Programme, mit denen solche Übersetzungen ausgeführt werden, heißen Compiler.

Ein Compiler ist also ein Computerprogramm, das ein in einer Quellsprache geschriebenes Programm in ein semantisch äquivalentes Programm einer Zielsprache umwandelt. Das Übersetzen wird auch als Kompilierung bezeichnet.

Daneben kann ein Programm in einer höheren Programmiersprache auch interpretiert werden. Dabei wird das Programm nicht vorab in Maschinencode übersetzt, sondern während seiner Laufzeit führt ein Interpreter die Anweisungen aus. Ein Interpreter übersetzt ein Programm nicht in die Zielsprache, sondern führt es Schritt für Schritt direkt aus.

In der Literatur findet man oft den Begriff der Objektorientierung.

Die objektorientierte Programmierung ist eine Grundidee, Daten und Funktionen, die auf diese Daten angewandt werden können, möglichst eng in einem sogenannten Objekt zusammenzufassen und nach außen hin zu kapseln, so dass Methoden fremder Objekte diese Daten nicht versehentlich manipulieren können. Zu den verbreitetsten objektorientierten Programmiersprachen zählen unter anderem C#, C++ und Java.

Unter der grafischen Benutzeroberfläche ist eine Software-Komponente zu verstehen, die dem Benutzer eines Computers die Interaktion mit der Maschine über grafische Symbole erlaubt. Synonyme Bezeichnungen sind die Abkürzung GUI (engl. „Graphical User Interface“) und dessen wörtliche Übersetzung grafische Benutzerschnittstelle.

Der Editor (von engl. to edit) ist ein Computerprogramm zur Erstellung und Bearbeitung von Daten.

Demgegenüber dient ein Viewer (deutsch: „Betrachter“) nur dem reinen Betrachten von Daten und arbeitet in der Regel nur im Lesezugriff.

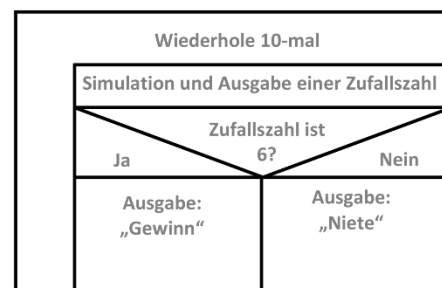
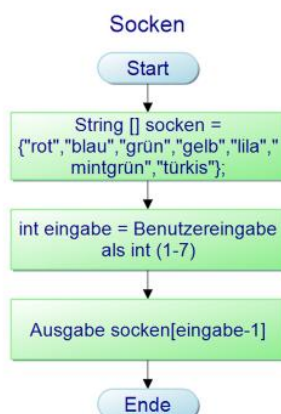
Bevor die eigentliche Programmierung beginnt, empfiehlt es sich oftmals ein Nassi-Shneidermann-Diagramm vor der Erstellung eines Codes zu entwerfen.

Ein Nassi-Shneiderman-Diagramm ist ein Diagrammtyp zur Darstellung von Programmentwürfen im Rahmen der Methode der strukturierten Programmierung und ist in der DIN 66261 genormt. Da Nassi-Shneiderman-Diagramme Programmstrukturen darstellen, werden sie auch als Struktogramme bezeichnet.

Die Methode zerlegt das Gesamtproblem, das man mit dem gewünschten Algorithmus lösen will, in immer kleinere Teilprobleme – bis schließlich nur noch elementare Grundstrukturen wie Sequenzen und Kontrollstrukturen zur Lösung des Problems übrig bleiben. Diese können dann durch ein Nassi-Shneiderman-Diagramm (DIN 66261) oder einen Programmablaufplan (DIN 66001) visualisiert werden.

Auftraggeber können die Gesamtheit ihrer Forderungen an die Leistungen eines Auftragnehmers durch ein Lastenheft beschreiben.

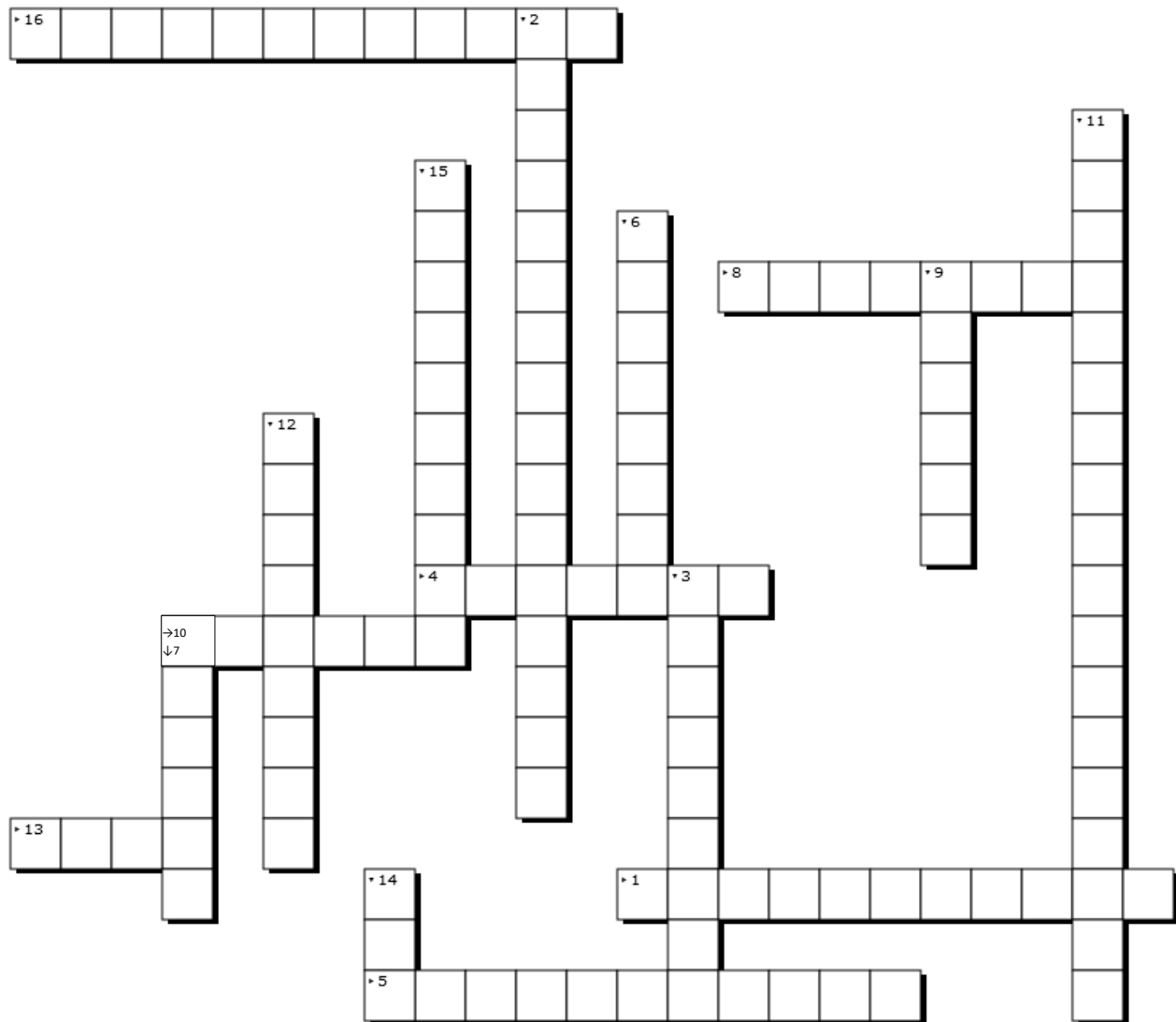
Das Lastenheft kann der Auftraggeber in einer Ausschreibung verwenden und an mehrere mögliche Auftragnehmer verschicken. Mögliche Auftragnehmer erstellen auf Grundlage des Lastenheftes ein Pflichtenheft, welches in konkreterer Form beschreibt, wie der Auftragnehmer die Anforderungen im Lastenheft zu lösen gedenkt. Der Auftraggeber wählt dann aus den Vorschlägen den für ihn geeigneten aus.



» Finden Sie das gesuchte Wort für das Kreuzworträtsel

1. Er ist die Bezeichnung für eine genau definierte Verarbeitungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen. Wer ist er?
2. Nachstehend ist ein Quelltext einer Programmiersprache abgebildet. Um welche Programmiersprache handelt es sich?
0000 1010 0011 0100 0000 1010 0011 0100
3. Englisch(er) Montagearbeiter macht eine Computersprache der CPU verständlich. Wie heißt dieser?
4. Eine technisch-wissenschaftliche Programmiersprache wird in Sprachenbäumen immer an oberster Stelle geführt. Wie heißt diese „Mutter“ der Programmiersprachen aus der sich so viele andere Programmiersprachen ableiten?
5. / 6. Im Bundestag halten zwei ausländische Politiker Reden vor den Abgeordneten. Der eine lässt seine Rede vorab ins Deutsche übersetzen und den Abgeordneten austeilen. Der zweite hält seine Rede ohne festes Manuskript. Für ihn wird ein Simultandolmetscher engagiert, der die Rede direkt den Abgeordneten übersetzt. Zur Überführung eines Quelltextes einer höheren Programmiersprache in eine dem Computer verständliche Form gibt es zwei verschiedene Übersetzungstools. Diese funktionieren ähnlich wie die beiden beschriebenen Übersetzungsvarianten der Politiker
7. Unter ihr versteht man den formalen Aufbau (das Regelwerk) einer Sprache. Wie heißt sie?
8. Ein Programm kann eine Iteration besitzen, die kein Ende findet. Was ist eine Iteration?
9. Mit ihm, in ihm werden Programme geschrieben?
10. Programmablauf zur Automatisierung von Routine-Aufgaben?
11. Smalltalk, C++, C# gehören zu einer bestimmten Art von Programmiersprache. Welches Konzept von Programmiersprachen wird hier verwirklicht?
12. Vom Vater auf den Sohn werden Eigenschaften und Verhaltensweisen übertragen. Wie heißt dieser Vorgang?
13. Indonesische Insel, Kaffee oder Programmiersprache? Wie heißt der gemeinsame Name?
14. Macht den Umgang mit einem Programm komfortabler (Abkürzung).
15. Dieses Dokument bietet die klar formulierte Aufgabenbeschreibung durch den Kunden. Wie wird es bezeichnet?
16. Wie heißt ein grafisches Hilfsmittel für die Planung eines strukturierten Programmablaufs?

© Zm



fisi_einstieg_Vk_20190911.docx

2. Compiler und Interpreter

Auszug aus [2]:

Compiler und Interpreter sind Implementierungsformen von Software. Generell geht es beim Compilieren und Interpretieren darum, den Quelltext, der mit einer höheren Programmiersprache (zum Beispiel C++, C# oder Java) geschrieben wurde, in Maschinenbefehle umzusetzen. Das bedeutet, die lesbaren Programmierbefehle müssen in weniger komplexe Instruktionen übersetzt werden, damit der Prozessor diese ausführen kann."

Compiler

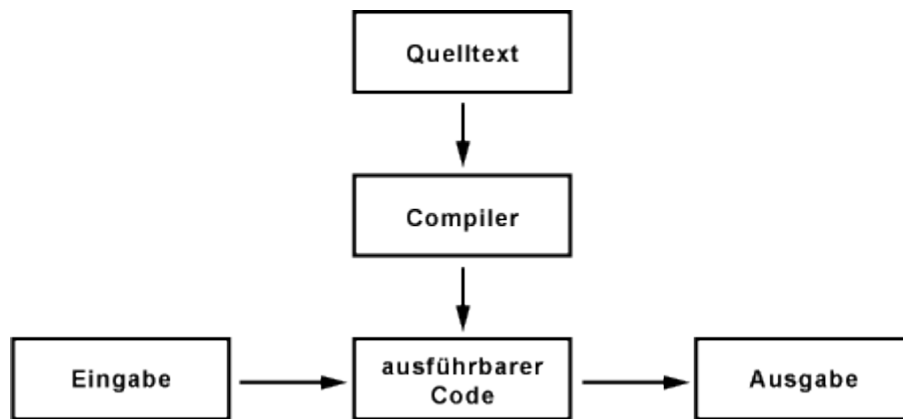


Abbildung Compiler: [2]

Der Compiler ist ein Programm, der aus dem Quelltext das eigentliche Programm erstellt. Der Quelltext könnte zum Beispiel in C oder C++ geschrieben sein. Einzelne Anweisungen aus dem Quelltext werden in Folgen von Maschinenanweisungen übersetzt. Nach dem Kompilieren wird das Programm erzeugt, in dem sich ausführbarer Code befindet. Bei der Ausführung des Programms werden diese Anweisungen "direkt" vom Prozessor ausgeführt. Jede Maschinenanweisung entspricht dabei einer festgelegten Bitfolge. Innerhalb des Prozessors wird diese Bitfolge verarbeitet. Der ausführbare Code verarbeitet auch Eingaben und erzeugt die Ausgabe.

Sobald ein neues Betriebssystem oder ein neuer Prozessor zum Einsatz kommt, muss der Quelltext neu kompiliert werden. In der Praxis ist es so, dass Prozessoren und Betriebssysteme einen Kompatibilitätsmodus haben, so dass alte Programme auch auf neuen Plattformen laufen. So müssen kompilierte Programme nicht immer neu kompiliert werden. Aber die Kompatibilität hat natürlich ihre Grenzen.

Typische Programmiersprachen mit Compiler sind Pascal, Modula, COBOL, Fortran, C und C++.

Interpreter

Die Programmiersprache wandelt den Quelltext in einen von der Hardware unabhängigen Bytecode um. Anschließend wird dieser Code vom Interpreter ausgeführt.

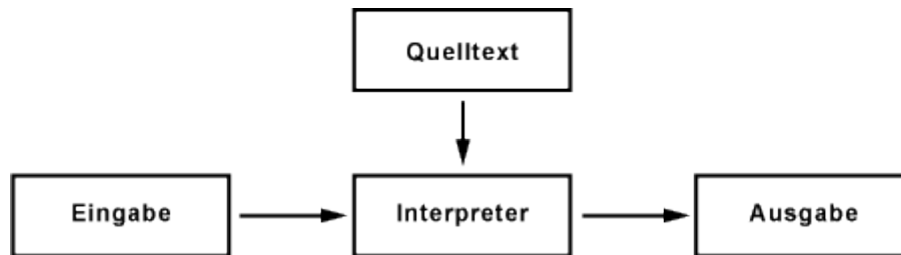


Abbildung Interpreter: [2]

Typische Programmiersprachen mit Interpreter sind BASIC, Smalltalk, LISP und Python.

» Nennen Sie einige Vor- und Nachteile von Compiler und Interpreter!

Compiler

Vorteile:

Nachteile:

Interpreter

Vorteile:

Nachteile:

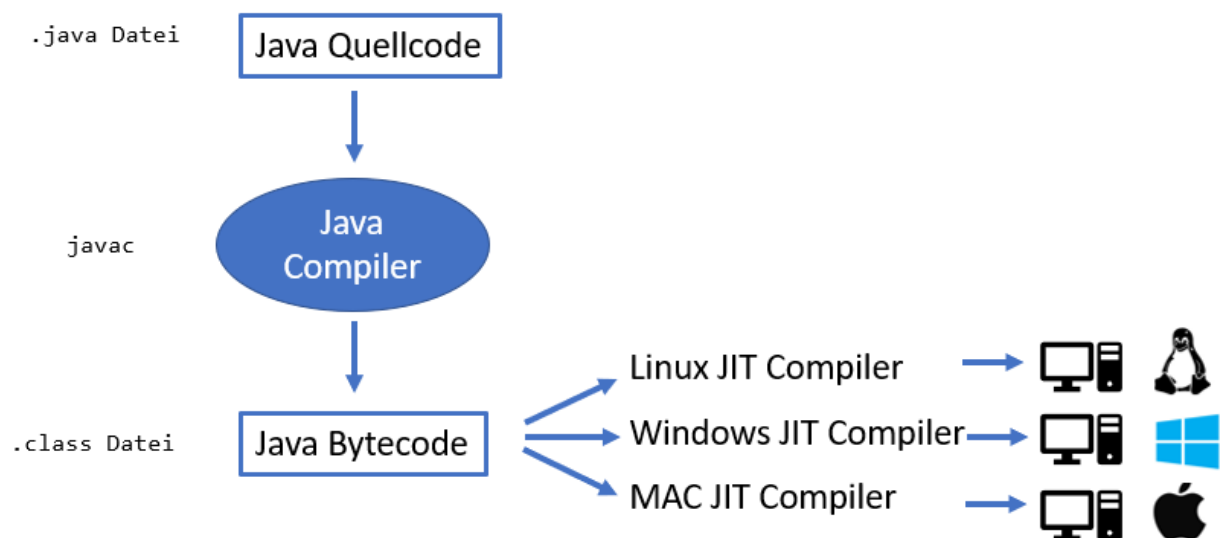
JIT-Compiler (JIT = Just in time)

Just-in-time-Kompilierung ist ein Verfahren, um Programme zur Laufzeit in Maschinencode zu übersetzen. Ziel ist es dabei, die Ausführungsgeschwindigkeit gegenüber einem Interpreter zu steigern.

Wie beim Interpreter wird der Quelltext in eine von der Hardware unabhängigen Bytecode übersetzt. Bei der Ausführung dieser Anweisungen durch den JIT-Compiler werden die Anweisungen in Maschinencode übersetzt. Der JIT-Compiler berücksichtigt dabei die verschiedenen Eigenheiten des Prozessors.

Beim JIT-Compiler wird aus dem Bytecode der Programmiersprache der Maschinencode für den Prozessor generiert, weil das schneller ist, als wenn der Interpreter den Bytecode ausführt. Der Bytecode wird nach der Ausführung verworfen und muss bei nochmaliger Ausführung erneut übersetzt werden.

Typischerweise arbeitet man mit einem JIT-Compiler, wenn es um das Prinzip "write once, run anywhere" geht. Wenn es also darum geht, den Quelltext auf unterschiedlichen Betriebssystemen und Hardware einzusetzen. Eine Software wird einmal geschrieben und ist dann auf verschiedenen Plattformen lauffähig (Portabilität). [2]



3. Das erste Programm

Java-Programme müssen stets nach einem bestimmten Schema angelegt werden. Zu Beginn muss die Klassendeklaration erfolgen:

```

1  class Klassenname
2  {
3
4      (bei Klassennamen ersten Buchstaben immer großschreiben)
5
6  }

```

Java source f nb char : 30 Ln : 4 Col : 1 Sel : 0 Dos\Windows ANSI INS

» Danach folgt die Hauptmethode main. Damit Java "weiß", wo die Ausführung eines Programmes anfangen soll, muss diese Methode in einer Klasse angelegt werden.

Zwischen die beiden geschweiften Klammern der main-Methode kommt Ihr Java-Code.

```

1  class Klassenname
2  {
3      public static void main (String [] args)  Hauptmethode (macht das Programm ausführbar)
4      {
5          System.out.println("Hallo Welt");      Platz für Ihre Anweisungen im Programm
6      }
7
8
9

```

Überblick

» Die folgende Grafik gibt eine kurze Erklärung zu dem Code, der in jedem Ihrer JAVA Programme vorkommen muss.

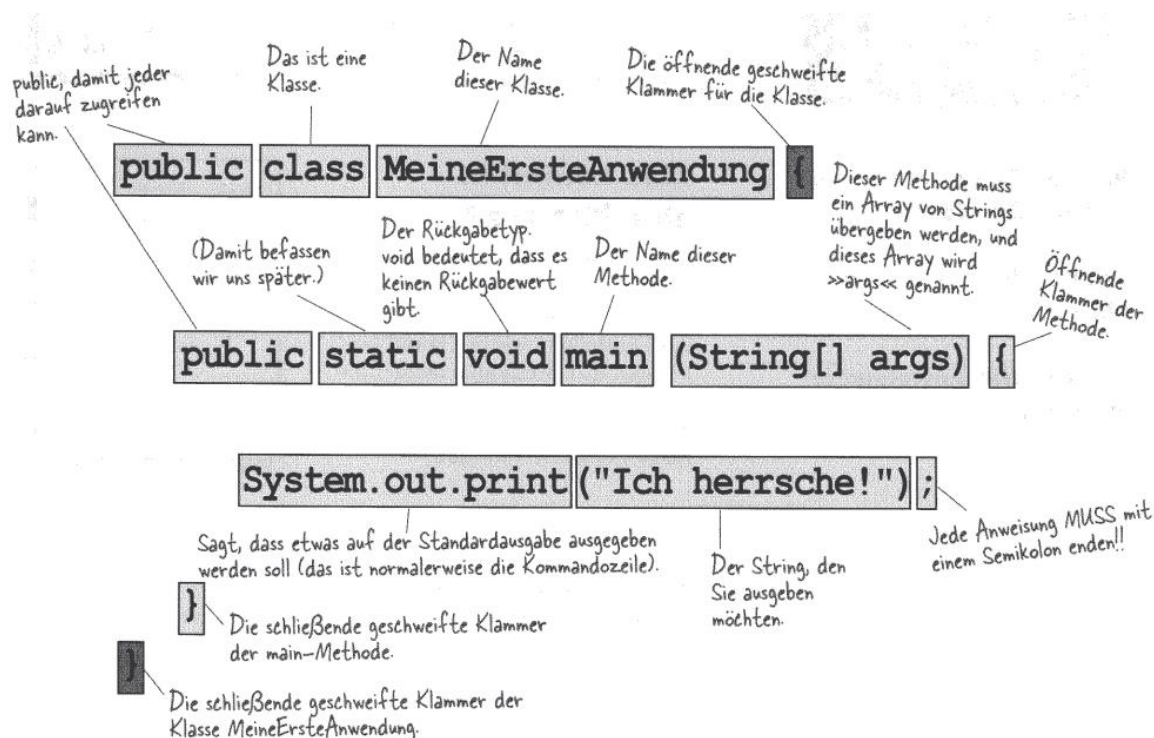


Abbildung: [3]: S.8

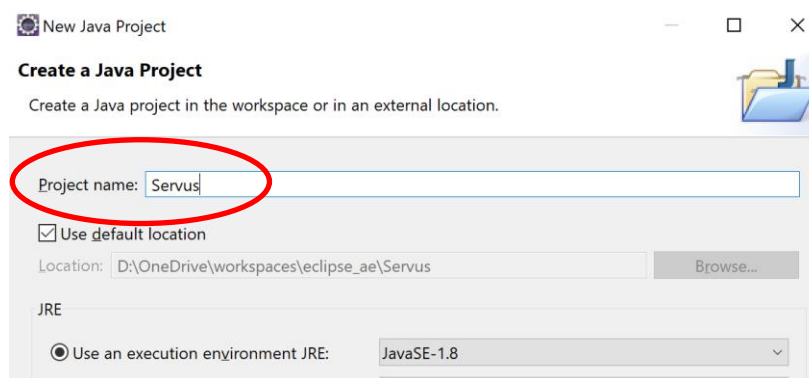
Entwicklungsumgebung Eclipse (für FIAE)

Eclipse ist ein quelloffenes Programmierwerkzeug zur Entwicklung von Software verschiedenster Art. Ursprünglich wurde Eclipse als integrierte Entwicklungsumgebung (IDE) für die Programmiersprache Java genutzt, aber mittlerweile wird es wegen seiner Erweiterbarkeit auch für viele andere Entwicklungsaufgaben eingesetzt.

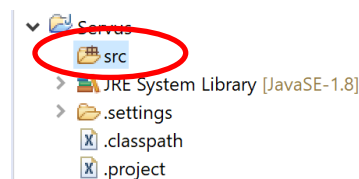
Eclipse vom USB Stick starten

Starten Sie Eclipse von Ihrem USB-Stick (aus dem Startmenü des Sticks oder unter \PortableApps\eclipseAE.) Aufgrund des Virenschanners im Schulnetz kann der erste Start etwas dauern.

Gehen Sie auf File – New – Java Project. Der Projektname soll mit einem Großbuchstaben beginnen.



Klicken Sie mit der rechten Maustaste auf das src im Projekt Servus. Legen Sie mit New – Class eine neue Klasse an.



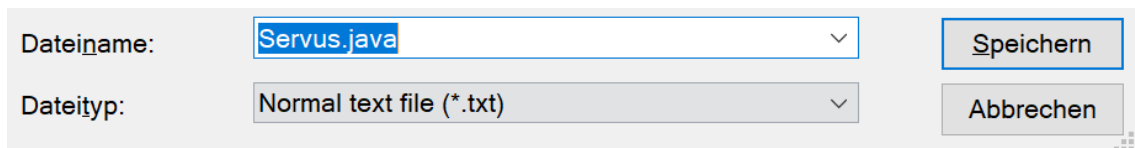
Legen Sie für eine bessere Übersicht für jeden Programmierauftrag ein neues Projekt an!

Arbeiten mit dem Notepad++ (für FISI)

Für einfache Programme können Sie auch mit dem Notepad++ arbeiten, welches auf dem USB Stick in einer portablen Version vorliegt.


Schreiben Sie Ihren Programmcode einfach in das Notepad und speichern Sie es als Java Datei ab (Dateiname.java). Speichern Sie es auf Ihrem Stick unter \Documents\Workspaces\Java

Sie sollten das Programm unter dem gleichen Namen speichern wie Sie Ihre Klasse benannt haben. Bei Klassen schreiben wir den ersten Buchstaben des groß.



Für das Kompilieren und Ausführen benötigen Sie die JavaCMD, welche sich ebenfalls auf dem Stick befindet.

Überprüfen Sie, ob Sie im Verzeichnis mit den gespeicherten Java-Dateien sind.

 Java_cmd

E:\Documents\workspaces\java>

Falls nicht, können Sie mit `cd` (change directory) in den entsprechenden Ordner wechseln.

cd .	springt einen Ordner zurück
cd name	wechselt in den Ordner name
cd \	springt auf das Root-Verzeichnis zurück
E:	wechselt in das Laufwerk E:

Anschließend müssen Sie das Programm kompilieren und anschließend ausführen.

Beispiel:

javac Servus.java	→ Kompilieren des Programms (erstellt eine Servus.class Datei)
java Servus	→ Starten des Java Programms

Arbeiten mit Codiva.io (ohne USB-Stick möglich)

Mit Codiva können Sie online¹ mit einer Java-Entwicklungsumgebung arbeiten. Den folgenden Link am besten als Lesezeichen speichern:

<https://www.codiva.io/java>

Falls Sie möchten, können sich bei Codiva anmelden um all Ihre Programme online zu speichern. Alternativ sichern Sie Ihre Programme lokal, damit Sie mit den Java-Programmen auch lernen können.

Start coding now

Try without login

Quell-Dateien des Projekts

Java Dateien, die für das Projekt angelegt werden.

Projektübersicht / Neues Projekt

Hier können Sie sich Ihre älteren Projekte ansehen und auch neue Projekte erstellen.

Teilen des Projekts

Es wird ein Link erstellt, der einen Zugriff auf den Programm-Code erlaubt.



Im weiteren Verlauf des Jahres werden Sie Eingaben über die Tastatur (Scanner) vornehmen können, diese können Sie in der unteren Leiste eingeben.

Eingabe von Werten

Bei der Arbeit mit dem Scanner können hier Werte eingegeben werden.

¹Die Google Suche im Schulnetz führt aufgrund von Google Sicherheitsstandards oft zu Captcha-Abfragen. Wir empfehlen als Alternative die ökologisch inspirierte Website www.ecosia.de. Etwa 80 % des Einnahmenüberschusses werden an gemeinnützige Naturschutzorganisationen gespendet.





Aufgabenstellung Servus

Erstellen Sie ein erstes Programm mit dem Namen **Servus.java**! Bearbeiten sie anschließend das folgende Kapitel **Einfache Ausgaben in Java**!

4. Einfache Ausgaben in Java

» Beschreiben Sie den Unterschied zwischen `System.out.println("xxx")` und `System.out.print("xxx")`?

» Beschreiben Sie folgende Ausgaben!

`System.out.print("\n");` ?

`System.out.print("\t");` ?

» Welche Ausgabe liefern die Zeilen dargestellten Programmzeilen?

Versuchen Sie die Lösung erst **ohne Programmierung** zu lösen! Überprüfen Sie anschließend Ihre Ergebnisse am Rechner!

Aufgabe 1

```
System.out.println("1")
System.out.print("2")
System.out.print("3")
System.out.println("4")
```

Aufgabe 2

```
System.out.print("1\n");
System.out.print("2");
System.out.print("3\n");
System.out.print("4\n");
```

Aufgabe 3

```
System.out.println("1\n");
System.out.print("2\n");
System.out.print("3\n");
System.out.println("4\n");
```

Aufgabe 4

```
System.out.println("1");
System.out.println("2\n");
System.out.print("3");
System.out.println("4");
```

5. Datentypen

Variablendeklaration und Variableninitialisierung

Bei Java muss jede Variable vor der Verwendung eingeführt werden (Deklaration). Variablen müssen einen **Typ** und einen **Namen** besitzen. Als Initialisierung bezeichnet man die Wertzuweisung zu einer Variablen.

Deklaration: _____

Initialisierung: _____

Elementare Typen

Typ	Bit-Tiefe	Wertebereich
boolean und char		
boolean	JVM-spezifisch	<i>true</i> oder <i>false</i>
char	16 Bit	Unicode Zeichen
numerisch (alle vorzeichenbehaftet)		
<i>Ganzzahlen</i>		
byte	8 Bit	- 128 bis 127
short	16 Bit	- 32768 bis + 32767
int	32 Bit	- 2 147 483 648 bis + 2 147 483 647
long	64 Bit	-9 223.372 036 854 775 808 bis +9 223 372 036 854 775 807
<i>Fließkommazahlen</i>		
float	32 Bit	variabel
double	64 Bit	variabel

Beschreibung JVM:

camelCase



Camel case is the practice of writing compound words or phrases such that each word or abbreviation in the middle of the phrase begins with a capital letter, with no intervening spaces or punctuation. Example: iPhone, eBay

Variablen schreiben wir beginnend mit einem kleinen Buchstaben, Klassen mit einem Großbuchstaben.

Bsp.: boolean headlessChickenMonster = true; ²



[4]

² Die Seegurke Enypniastes eximia, auch "headless chicken monster" genannt, ist eine äußerst seltene Spezies. Sie erreicht Körperlängen von 6 bis 25 cm und halten sich in Meeren in Tiefen zwischen 300 und 6000 Metern auf.

Wert und Variable

```
int x = 24;
byte b = x; // funktioniert nicht!!
```

Begründen Sie, warum das nicht funktioniert!



[3]: S.52

» Folgende Zeilen befinden sich in einer gemeinsamen Methode.

Verbessern Sie falls nötig die Anweisungen!

1. int x = 34.5;	8. float wert = 32.5 ;
2. boolean boo = 20;	9. char test = "a";
3. int g = 17;	10. long bla = 2147483647;
4. int y = g;	11. long blabla = 2147483648 ;
5. byte b = 3;	12. int 4zahl = 129;
6. byte v = b;	13. int erg = 34__12;
7. short n = 32768;	14. double wert = 14,3;

String

Ein String ist eine Zeichenkette. Der Datentyp String hat als Wertebereich die Menge aller Zeichenketten. Strings werden in Anführungszeichen gesetzt; Beispiele für Strings sind "abba", "Hallo Fritz!", "@%#&\$", "x".

Das folgende Programmstück zeigt die Deklaration einer Variablen **s** vom Typ String gefolgt von einer Wertzuweisung und anschließender Ausgabe des Strings.

```
String s ="Hallo!";
System.out.println(s);
```

Aufgabe - Steckbrief

Erstellen Sie ein Programm, mit Variablen für Name, Alter und Geschlecht (m/w). Initialisieren Sie die Variablen mit Werten und geben Sie diese über die Ausgabe am Bildschirm aus! Erweitern Sie das Programm um weitere Werte!

Helfen Sie nach dieser Übung bitte Ihren Mitschülerinnen und Mitschülern, die noch Probleme bei den Aufgabenstellungen haben!

Literaturverzeichnis

- [1] „Public Domain Pictures,“ Bobek Ltd, [Online]. Available: <https://www.publicdomainpictures.net/de/view-image.php?image=286864&picture=frischer-kaffee>. [Zugriff am 05 09 2109].
- [2] „Elektronik Kompendium,“ 04 08 2019. [Online]. Available: <http://www.elektronik-kompendium.de/sites/com/1705231.htm>. [Zugriff am 04 09 20019].
- [3] K. S. & B. Bates, Java von Kopf bis Fuß, O REILLY, 2006.
- [4] W. Commons, „https://commons.wikimedia.org/wiki/File:Enypniastes_eximia_Puerto_Rico.jpg,“ 4 11 2018. [Online]. Available: https://commons.wikimedia.org/wiki/File:Enypniastes_eximia_Puerto_Rico.jpg. [Zugriff am 2019 09 04].
- [5] „needpix.com 2019,“ [Online]. Available: <https://www.needpix.com/photo/946482/merapi-an-active-volcano-game-bubble-mountain-views-from-java-island>. [Zugriff am 05 09 2019].