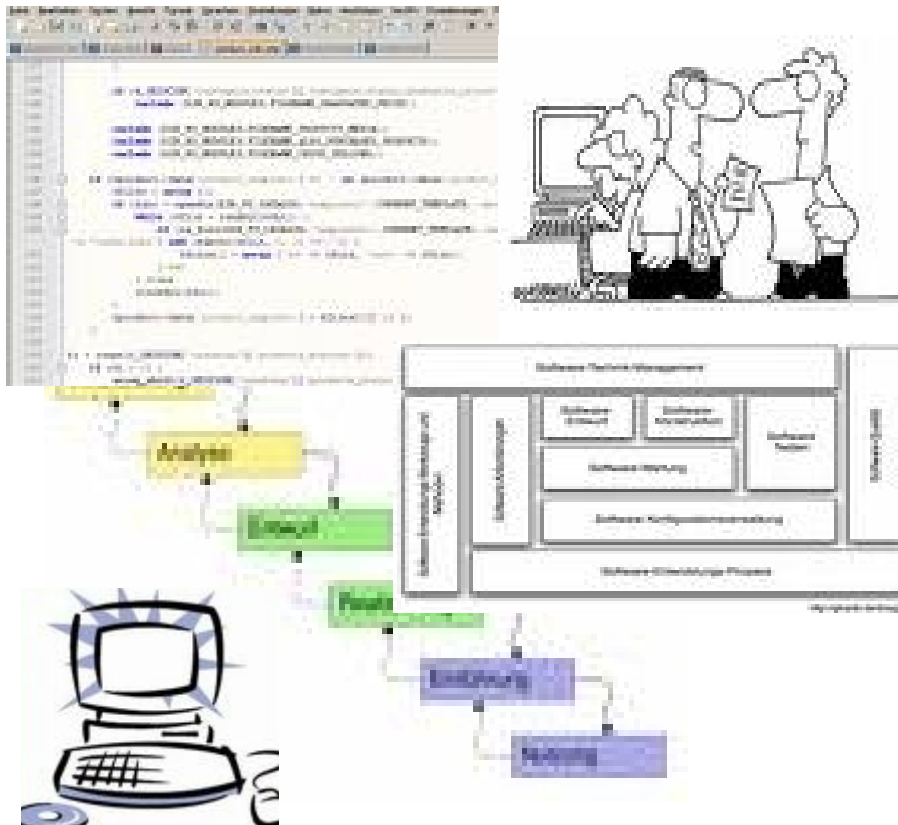


Anwendungsentwicklung Einführung



Inhalt

- Theorie (I)
 - Programmierung
- Praxis
 - Hallo-Welt-Beispiel
 - Konsolenausgabe & Konsoleneingabe
 - Datentypen & Variablen
 - Operatoren & Zuweisungen
- Theorie (II)
 - Programmierwerkzeuge
 - Programmiersprachen
 - Vorgehensmodelle
 - Entwicklungsparadigmen & Methoden
- Software-Arten & Software-Qualität



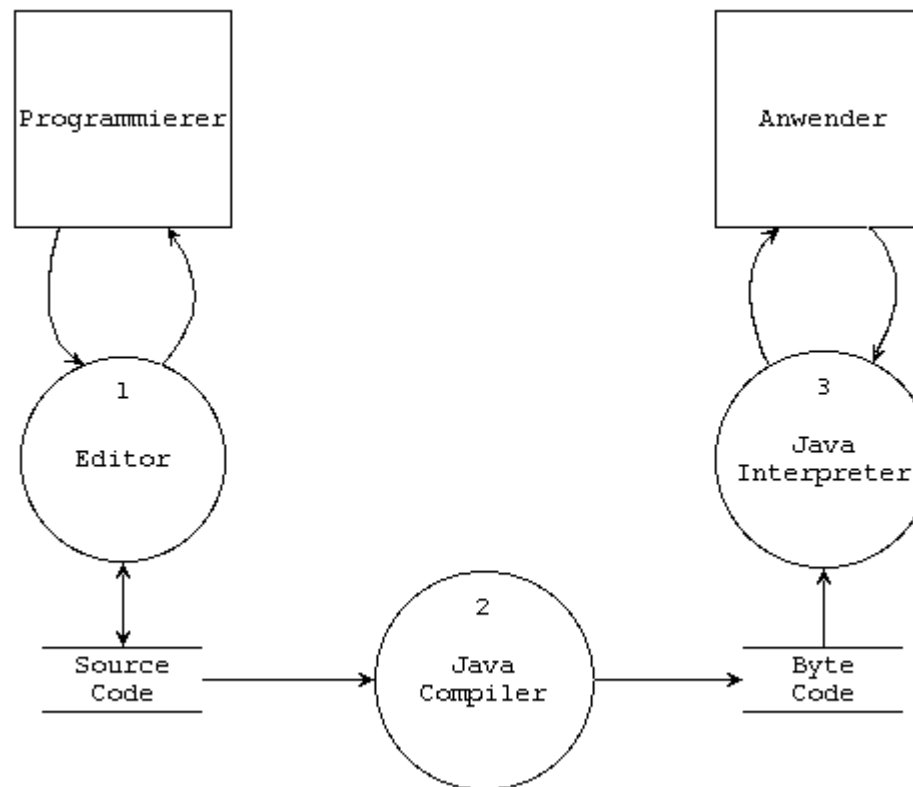
Hallo Welt – Source Code



```
//  
//  Beispiel_ls1_01  -   Hallo Welt Beispiel  
//  
  
public class bsp_01 {  
  
    public static void main(String[] args) {  
        System.out.println("Hallo Welt!");  
    }  
  
}
```

Beispiel_ls1_01

Java-Programmierung



Hallo Welt – Vorgehensweise



- Integrierte Entwicklungsumgebung (IDE): **Eclipse**
- **Vorgehensweise:**
 1. Neues Projekt anlegen
 2. Neue Klasse mit main-Funktion anlegen
 3. Main-Funktion programmieren
 4. Programm testen

Beispiel 2 - Reihenwiderstand



```
import java.util.*;

public class Beispiel_02 {

    public static void main(String[] args) {
        double r1 = 0;
        double r2 = 0;
        double r3 = 0;
        Scanner input = new Scanner(System.in);

        // Widerstandswerte einlesen
        System.out.print("Widerstand 1 (Ohm): ");
        r1 = input.nextDouble();
        System.out.print("Widerstand 2 (Ohm): ");
        r2 = input.nextDouble();

        // Gesamtwiderstand berechnen
        r3 = r1 + r2;

        // Gesamtwiderstand ausgeben
        System.out.println("Der Gesamtwiderstand beträgt " + r3 + " Ohm");
    }
}
```

Beispiel_02

Hinweise (I)



Kommentare

- Kommentare verbessern das Verständnis von Programmen
- Kommentare können mit `//` beginnen und gehen dann bis zum Zeilenende
- Mehrzeilige Kommentare beginnen mit `/*` und enden mit `*/`

Vereinfacht;
Details folgen später

Klassen und main-Funktion

- In Java als rein objekt-orientierter Sprache muss alles in Form von Klassen realisiert werden.
- Klassen werden mit den Schlüsselwörtern `public class <Klassenname>` angelegt.
- Klassen und Funktionen werden mit geschweiften Klammern begrenzt.
- Ist in einer Klasse, eine Funktion mit Namen `public static void main` vorhanden, so wird diese zuerst aufgerufen.

Hinweise (II)



Anweisungen

- Anweisungen werden immer mit einem Semikolon ; abgeschlossen

Konsolenausgabe

- Mit der Anweisung `System.out.println("Hallo Welt!");` können Konsolenausgaben realisiert werden.
- `println` ist eine Funktion des Objekts `System.out`. Der auszugebende Text steht in Hochkommata.

Einfache Datentypen



Art	Typ	Inhalt	Wertebereich
Wahrheitswert	boolean	1 Bit	true, false
Zeichen	char	16 Bit	Unicode-Zeichen 0...65535
Ganze Zahl	byte	8-Bit-Zahl	-128 ... +127
	short	16-Bit-Zahl	-32.768 ... +32.767
	int	32-Bit-Zahl	-2.147.483.648 ... +2.147.483.648
	long	64-Bit-Zahl	-9.223.372.036.854.775.808 ... +9.223.372.036.854.775.807
Fließkomma- zahl	float	32-Bit-Zahl (Kommazahl)	$\pm 3.0282347E+38$... $\pm 1.40239846E-45$
	double	64-Bit-Zahl (Kommazahl)	$\pm 1.7976931348623157E+308$... $\pm 4.9406564581246544E-324$

Variablen



- **Variablen** sind Speicherplätze (Behälter), in denen sich ein Programm einen Wert merken kann.
- Während der Laufzeit können Variablen ihren Wert ändern.
- Variablen müssen vor der Nutzung definiert (deklariert) werden.
- Bei einer **Variablendeklaration** muss für eine Variable der Variablenname und der Datentyp angegeben werden. Optional kann noch ein Startwert angegeben werden.
- Mit dem Schlüsselwort **final** können **Konstanten** definiert werden. Diese dürfen ihren Wert nicht mehr ändern.

```
double    r1;  
long      a, b, c;  
  
boolean    b1 = true;  
float      x = 3.7982;  
float      y = 6.8E3;  
char       c = 'x';  
  
final byte GELB = 1;
```

Operatoren und Zuweisungen



Mathematische Operatoren

Zeichen	Anwendung	Bedeutung
+	$x + y$	Addition
-	$x - y$	Subtraktion
*	$x * y$	Multiplikation
/	x / y	Division
%	$x \% y$	Modulo (Rest nach einer ganzzahligen Division)

Logische Operatoren

Zeichen	Anwendung	Bedeutung
&&	$x \&\& y$	UND; y wird nicht ausgewertet, wenn $x == \text{false}$
&	$x \& y$	UND; x und y werden ausgewertet
	$x y$	ODER; y wird nicht ausgewertet, wenn $x == \text{true}$
	$x y$	ODER; x und y werden ausgewertet
!	$!x$	NICHT (Negation)
^	$x \wedge y$	Exklusives ODER (Entweder oder :-)

Zuweisungsoperatoren

Zeichen	Anwendung	Bedeutung
=	$x = y$	Variable x auf den Wert von y setzen

Zeichen	Anwendung	Äquivalent zu:
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$

Zeichen	Anwendung	Bedeutung
++	x++	x auslesen; dann um 1 erhöhen (Postinkrement)
	++x	x um 1 erhöhen; dann auslesen (Preinkrement)
--	x--	x auslesen; dann um 1 erniedrigen (Postdekrement)
	--x	x um 1 erniedrigen; dann auslesen (Predekrement)

Quelle:
<https://people.oh14.de/dave/java/operat.shtml>

Konsoleneingabe



```
import java.util.*;

public class Beispiel_02 {

    public static void main(String[] args) {
        ...
        Scanner input = new Scanner(System.in);    // Konsoleneingabe Varianten
        String str;
        System.out.print(" Text eingeben: ");      // Texteingabe
        str = input.nextLine();
        System.out.println("Text: " + str);
        System.out.print(" Kommazahl eingeben: "); // Gleitkommazahleingabe
        x = input.nextDouble();
        System.out.println("Kommazahl: " + x);
        int i;
        System.out.print(" Ganzzahl eingeben: "); // Ganzzahleingabe
        i = input.nextInt();
        System.out.println("Ganzzahl: " + i);
        boolean b;
        System.out.print(" Boolschen Wert eingeben: "); // Boolscher Wert
        b = input.nextBoolean();
        System.out.println("Ganzzahl: " + b);
    }
}
```

Beispiel_Is1_03

Informationsquellen



- **Java**, insbesondere die **Klassenbibliotheken**
 - <http://docs.oracle.com/en/java/>
- **Java Runtime Umgebung**
 - www.java.com
- **Eclipse**
 - <https://eclipse.org/>