

Java - Basics

- Eclipse GUI
- Schlüsselwörter
- Hello World
- Funktionen und Attribute
- Ablaufsteuerung & Schleifen
- Bedingungen & Boolesche Ausdrücke

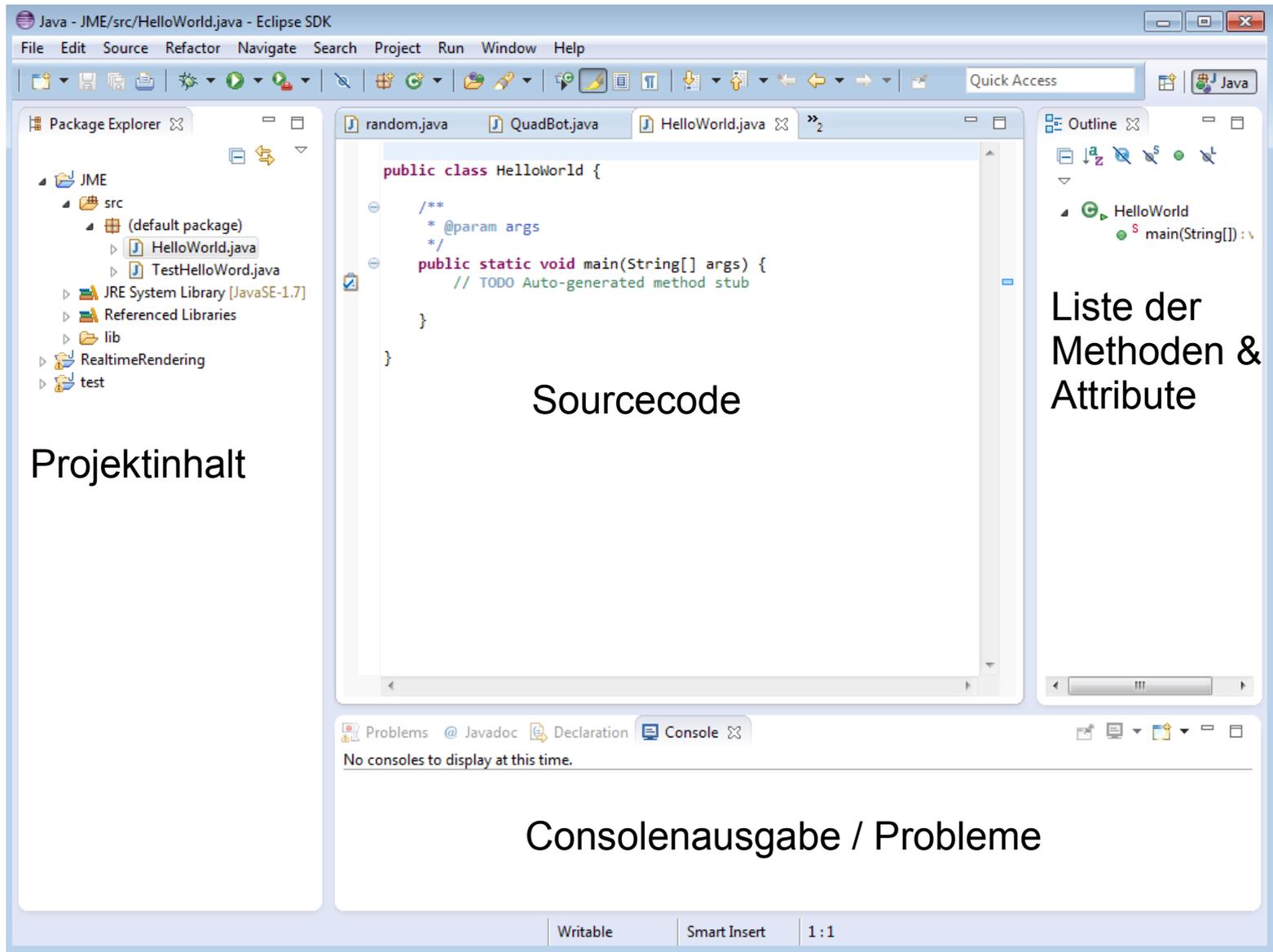
Eclipse GUI

- Workspace als Verzeichnis für alle Projektdateien
- Projekteigenschaften
 - Build-Path, Library (JDK vs JRE)
 - Run-Debug Settings (Program arguments)

Eclipse GUI

- Anlegen einer neuen Java-Datei
 - Rechtsklick auf SRC, New – Class
 - Option auswählen „main“ Methode zu erstellen
 - Name Eingeben
 - Klassenname = Dateiname
- Programm starten mit Run

Eclipse GUI Fenster



»Hello World«

```
class Hello  
{  
    public static void main (String[] args)  
    {  
        System.out.println("Hello World!");  
    }  
}
```

Schlüsselworte & Kommentare

- Schlüsselworte, Wortsymbole
 - Terminale Symbole wie *class*, *public*, *void*, *import*, die eine festgelegte Bedeutung besitzen
 - Dürfen nicht für andere Zwecke benutzt werden
- Kommentare
 - Gute Dokumentation notwendig, da ein Programm von mehreren Personen gelesen wird
 - Kommentare werden vom Compiler überlesen, sie dienen nur dem menschlichen Leser.

Funktionen & Attribute

- Attribut: Zustandsspeicher
- Methode (alias Funktion, Operator): Verhalten & Zustandsänderung
- Funktion Aufbau
 - Kopf
 - Sichtbarkeit (static) Rückgabe Name Parameter
 - `public static void main (String[] args)`
 - Rumpf { }
 - Enthält Algorithmus unter Verwendung der Parameter
 - Kann mit `return <Ausdruck>` enden, wenn Rückgabewerte nicht void ist

Funktionsaufruf

- Funktionsvariablen i und k
 - `func(int i, int k) { <Rumpf> }`
 - Sind innerhalb des Rumpfes verwendbar
 - Bei Aufruf: `func(12,10);` würde i mit 12 und k mit 10 beschrieben

Ausdruck

- Ist
 - Wert `12` oder
 - Funktionsaufruf, der Wert berechnet `Math.sqrt(12)`
 - Oder eine Variable, z.B. `b`
- Klammern steuern Auswertungsreihenfolge
 - `b= func1(func2 () + func3(b)) + 10;`
(ist eine Wertzuweisung der Variablen `b`)

Boolsche Ausdrücke

- Logisches Oder $a \ || \ b$
- Logisches UND $a \ \&\& \ b$
 - durch doppelte Nutzung des Zeichens | und & wird Compiler angewiesen b nicht abzufragen, wenn a bereits zu einem true/false führt

&	0	1
0	0	0
1	0	1

	0	1
0	0	1
1	1	1

Primitive Datentypen

- Reelle Zahl: float, double
- Ganze Zahl: int
- Buchstabe/Zeichen: char
- Boolescher Wert: boolean
 - Können jederzeit angelegt werden `int i=12;`
 - Sind innerhalb der umgebenden Klammern `{ }` nutzbar aber nicht außerhalb

Komplexe Datentypen

- Arrays: eine bestimmte Anzahl von primitiven Datentypen, zugriff über einen Index
 - `int[] zahlen= new int[12];` //erstellt 12 ints, auf die z.B. durch `zahlen[0]` zugegriffen werden kann
- Strings: Sammlung von chars mit vielen Operatoren
 - `String zeichenkette=new String();` oder
 - `String zeichenkette="meineZeichenkette";`

Kontrollfluss

- If: Steuert den Programmpointer auf Basis von Ausdrücken
 - **If** (<boolscher Ausdruck>==<>true oder false>)
 { <Rumpf bei zutreffen > } **else** { <alternative> }
- **While** (<boolscher Ausdruck>) {<Rumpf>}
 - Führt Rumpf aus, bis Ausdruck nicht mehr zutrifft
- **For** (int i=0;i<10;i=i+1) { <Rumpf> }
 - Führt Rumpf aus, bis Variable i die Bedingung <10 nicht mehr erfüllt