

# Übung 1: Einführung in Eclipse

## *Hinweise zum Start von Eclipse*

Zunächst sollen Sie Eclipse als integrierte Entwicklungsumgebung kennen lernen, indem Sie als erstes ein kleines Projekt "Hello World" zum Laufen bringen. Bearbeiten Sie bitte die folgenden Schritte.

1. Starten Sie Eclipse. Sie werden dabei gefragt, wo Ihr Workspace liegen soll. Legen Sie in Ihrem Home-Verzeichnis auf dem Laufwerk N: ein neues Verzeichnis "Workspace" (o. ä.) an. Wählen Sie dieses aus.

2. Auf der Begrüßungsseite können Sie Tutorials oder Beispiele abrufen. Durch Klicken auf das "Workbench"-Icon (rechts) kommen Sie zur üblichen Oberfläche von Eclipse, die allerdings noch nicht mit Inhalt gefüllt ist.

3. Wir wollen nun ein neues Projekt anlegen. Hierfür wählen Sie

**File → New → Project...**

4. In der sich öffnenden Dialogbox wählen Sie als Wizard "Java Project" und klicken auf die Schaltfläche **"Next"**.

Dann geben Sie den Projektnamen "Aufgabe" ein und übernehmen die anderen Standardeinstellungen und betätigen die **"Finish"**-Schaltfläche. Anschließend werden Sie in einer Dialogbox gefragt, ob Sie zur Java-Perspektive wechseln wollen, was Sie mit **"Yes"** beantworten sollten.

5. Nun wird links im Package Explorer das neue Projekt angezeigt.

6. Wählen Sie das Projekt an und erzeugen mit

**File → New → Package**

ein neues Paket für die benötigte Klasse.

7. Geben Sie in der Dialogbox den Namen, z.B. "helloworldpaket" an und betätigen Sie die Schaltfläche **"Finish"**. Paketnamen müssen mit einem Kleinbuchstaben beginnen.

8. Wählen Sie nun im Package Explorer das neue Paket an und erzeugen mit

**File → New → Class**

eine neue Klasse.

9. Geben Sie in der Dialogbox den Namen der Klasse, z.B. "HelloWorld" an. Klassennamen sollten immer mit einem Großbuchstaben beginnen. Der Modifier sollte "public" sein und im unteren Teil sollte "public static void main" angewählt sein, wodurch die Methode für das Hauptprogramm bereits von Eclipse mit erzeugt wird. Schließen Sie den Dialog durch die **"Finish"**-Schaltfläche.

10. Nun erscheint die neue Klasse sowohl im Package Explorer als auch in der großen Arbeitsfläche in der Mitte. Wie Sie leicht erkennen können, wurden die bereits eingegebenen Daten in die Klasse übernommen.

11. Schreiben Sie nun das Hello-World-Programm, indem Sie die folgende Zeile `System.out.println("Hello world!");`

im Hauptprogramm einfügen.

12. Speichern Sie die Datei und starten Sie sie über

**Run → Run As → Java Application.**

In der Konsole im unteren Bereich von Eclipse sehen Sie die gewünschte Ausgabe erscheinen. Falls keine Konsole erscheint, starten Sie diese mit

**Window → Show view → Console.**

13. Nun fügen wir einen bewussten Fehler ein, um zu sehen, wie sich Eclipse dann verhält:

Ändern Sie die Zeile zu

`System.println("Hello world!");`

Sehr rasch zeigt Eclipse am linken Rand an, dass ein Fehler vorliegt. Wenn Sie mit dem Mauszeiger auf dem Symbol verweilen, bekommen Sie genauere Informationen.

Versuchen Sie dennoch das Programm zu übersetzen indem Sie auf den grünen Pfeil klicken bzw.

#### **Run → Run Last Launched**

aufrufen. Brechen Sie den Ausführungsversuch ab. Im Fenster Problems sehen Sie die Fehlermeldungen vom Übersetzungsversuch. Durch Anklicken der jeweiligen Zeile in Problems gelangen Sie zum dazugehörigen Fehler im Quelltext.

#### **Eingabe von verschiedenen Werten**

Importieren Sie die die Eingaberoutinen in `java.util.Scanner` durch

```
import java.util.Scanner;
```

```
import java.util.Locale;
```

vor der Klassendeklaration.

Fügen Sie am Anfang folgende Zeilen ein

```
Scanner input = new Scanner(System.in);
```

und

```
String eingabe = "";
```

#### **Eingabe von Zeichenketten:**

Nun ersetzen Sie die Ausgabe durch

```
System.out.print("Name eingeben > ");
```

```
eingabe = input.next();
```

```
System.out.println("Hallo " + eingabe);
```

Beachten Sie, dass bei der ersten Ausgabe nur `print` steht. Hierdurch wird erreicht, dass kein Zeilenumbruch geschrieben wird.

#### **Eingabe von ganzen Zahlen:**

Fügen Sie am Anfang folgende Zeile hinzu

```
int gzahl = 0;
```

Ersetzen Sie die vorherigen Zeilen durch:

```
System.out.print("Ganze Zahl eingeben > ");
```

```
gzahl = input.nextInt();
```

```
System.out.println("Die Zahl war " + gzahl);
```

#### **Eingabe von Gleitkommazahlen:**

Fügen Sie am Anfang folgende Zeile hinzu

```
double dzahl = 0.0;
```

Ersetzen Sie die vorherigen Zeilen durch

```
System.out.print("Kommazahl eingeben (mit Komma)> ");
```

```
dzahl = input.nextDouble();
```

```
System.out.println("Die Zahl war " + dzahl);
```

Das System besitzt eine sogenannte Lokalisierung. Das bedeutet, dass die für ein Land geltenden Konventionen, z. B. die Zahlendarstellung, bei der Eingabe berücksichtigt werden. Will man bei der Eingabe die Nachkommastellen durch einen Punkt abtrennen (üblich bei Computer-Eingaben), so müssen Sie die Lokalisierung auf ein englischsprachiges Land abändern durch:

```
input.useLocale(Locale.UK);
```

vor der Eingabe mit

```
input.nextDouble().
```

### **Einfache Berechnungen**

Es sollen der Umfang und die Fläche eines Kreises berechnet werden. Hierzu soll die vorher eingegebene Gleitkommazahl den Radius darstellen. Zur Berechnung benötigen wir weitere Variablen am Anfang des Programms

```
double umfang, flaeche;
```

Bitte verwenden Sie bei Variablen keinesfalls Umlaute wie 'ä' sondern nur das im Englischen verwendete Alphabet.

Für die Berechnung benötigen wir die Zahl  $\pi$ . Eine möglichst genaue Approximation findet sich im Paket `java.lang.Math`. Obwohl es eigentlich unnötig ist wollen wir hierzu dieses Paket importieren durch `import java.lang.Math;` ganz am Anfang des Textes.

Nun fügen wir die Berechnung nach der Eingabe des Radius ein

```
umfang = 2.0 * Math.PI * dzahl;  
flaeche = Math.PI * dzahl * dzahl; // Math.PI * Math.pow(dzahl, 2)  
System.out.println("Umfang: " + umfang + " Fläche: " + flaeche);
```

Zwischen Doppelhochkommas können beliebige Zeichen, also auch Umlaute verwendet werden.

### **Optionaler Teil (Eingabe durch BufferedReader)**

Fügen Sie am Anfang folgende Zeilen ein

```
BufferedReader tastatur =  
    new BufferedReader(new InputStreamReader(System.in));
```

und

```
String eingabe = "";
```

*Eingabe von Zeichenketten:*

Nun ersetzen Sie die Ausgabe durch

```
System.out.print("Name eingeben > ");  
try {  
    eingabe = tastatur.readLine();  
} catch (IOException ex) {  
    System.out.println(ex.getMessage());  
}  
System.out.println("Hallo " + eingabe);
```

*Eingabe von ganzen Zahlen:*

Fügen Sie am Anfang folgende Zeile hinzu

```
int gzahl = 0;
```

Ersetzen Sie die vorherigen Zeilen durch:

```
System.out.print("Ganze Zahl eingeben > ");  
try {  
    eingabe = tastatur.readLine();  
} catch (IOException ex) {  
    System.out.println(ex.getMessage());  
}  
try {  
    gzahl = Integer.valueOf(eingabe);  
} catch (NumberFormatException nf) {  
    System.out.println("Falsches Zahlenformat");  
}  
System.out.println("Die Zahl war " + gzahl);
```

### *Eingabe von Gleitkommazahlen:*

Fügen Sie am Anfang folgende Zeile hinzu

```
double dzahl = 0.0;
```

Ersetzen Sie die vorherigen Zeilen durch

```
System.out.print("Kommazahl eingeben (mit Punkt)> ");  
try {  
    eingabe = tastatur.readLine();  
} catch (IOException ex){  
    System.out.println(ex.getMessage());  
}  
try {  
    dzahl = Double.valueOf(eingabe);  
} catch (NumberFormatException nf) {  
    System.out.println("Falsches Zahlenformat");  
}  
System.out.println("Die Zahl war " + dzahl);
```

Die einfachere Klasse "BufferedReader" besitzt keine Lokalisierung. Daher können hier nur Zahlen im englischen Format eingegeben werden.