

# Einführung

## 1. Einführung

Ziel der Aufgabe ist es, Eclipse etwas kennen zu lernen und zu wissen, wie man ein neues Java-Projekt in Eclipse anlegt.

### 1.1 Information: Neues Projekt anlegen

- File → New → Java Project (oder: File → New → Project... → Java → Java Project)
- Projektname eingeben und mit *Finish* bestätigen
- Paket anlegen mit File → New → Package
- Paketname eingeben und mit *Finish* bestätigen
- Neue Klasse in Paket anlegen mit File → New → Class
- Klassenname eingeben
- Bei *Which method stubs would you like to create?* die `main()`-Methode auswählen und mit *Finish* bestätigen.

### 1.2 Information: Kompilieren und Ausführen

- Run → Run As → Java Application
- oder direkt in der Menüleiste den grünen Button benutzen

### 1.3 Konventionen für den Unterricht

- **Workspace auf Netzlaufwerk „H:“!**
- Wir legen für die ersten Wochen ein Projekt **Einfuehrung** an.
- In diesem Projekt werden einzelne Pakete für die Arbeitsblätter angelegt. (**21Einfuehrung**)
- In diesem Paket sollen die Aufgaben als einzelne Klassen angelegt werden (z. B. **Aufgabe1a**)
- (Jeder Quellcode soll auch kommentiert werden, das hilft beim späteren Durchlesen dem Verständnis!)

## 2. Ein erstes Programm

Die `main`-Methode wird beim Programmstart aufgerufen, d. h. die Befehle in dieser Methode werden nacheinander abgearbeitet.

Mit dem Befehl

```
System.out.println("Hallo_Welt");
```

Listing 1: Ausgeben von Daten auf der Konsole

wird die Zeichenkette `Hallo Welt` auf der Konsole ausgegeben.

Lege also wie oben beschrieben ein neues Projekt `Einfuehrung` mit einem Paket `21Einfuehrung` an und erstelle darin eine Klasse `Aufgabe2`. Achte bei der Erzeugung der Klasse darauf, dass diese eine `main`-Methode enthält!

Schreibe also zunächst diesen Befehl in die `main`-Methode (d. h. innerhalb der geschweiften Klammern) und führe das Programm aus.

Neben dem Befehl `System.out.println()`; gibt es auch noch den Befehl `System.out.print()`; Beschreibe kurz den Unterschied der beiden Befehle:

---

---

---

### 3. Variablen

Mithilfe von sogenannten Variablen können wir Werte zwischenspeichern. Eine Variable `a` können wir mithilfe des Befehls

```
int ersteVariable;
```

erzeugen. Um dieser Variablen beispielsweise den Wert `5` zuzuweisen schreiben wir

```
ersteVariable = 5;
```

Dieser Variablen haben wir dabei den Namen `ersteVariable` gegeben. Nachdem wir diese Variable erzeugt und einen Wert festgelegt haben können wir diese danach im Quelltext verwenden und schreiben dabei statt dem Wert den Namen dieser Variablen. Wollen wir beispielsweise den *Wert dieser Variablen* auf der Konsole ausgeben schreiben wir den Befehl

```
System.out.println(ersteVariable);
```

Das Wort `int` gibt dabei den sogenannten Datentyp an, d. h. welche Art von Werten können in der Variable. `int` bedeutet dabei, dass wir in der Variablen `ersteVariable` *ganze Zahlen* speichern können. Recherchiere im Internet, welche anderen Datentypen es noch gibt:

---

---

---

---

---

---

---

---

---

---