

## Java06 – Konsolenausgabe und Berechnungen

Unsere bisherigen BlueJ-Projekte zum Zeichnen von Bildern beruhten auf einer Vorlage, in der die Klassen „Leinwand“, „Kreis“, „Quadrat“, „Dreieck“ und „Person“ bereits vorgegeben waren.

Nun erstellen wir ein neues BlueJ-Projekt „von Grund auf“, mit dem wir Text auf der „Konsole“ (dem Ausgabefenster für Text) ausgeben lassen wollen und auch Berechnungen durchführen wollen.

### Schritt 1 – Neues BlueJ-Projekt erstellen

Öffne zunächst BlueJ und erstelle mit „Projekt“ → „Neues Projekt“ ein neues BlueJ-Projekt mit dem Namen „**Java\_03\_Konsolenausgabe**“ und speichere es in Deinem Home-Laufwerk („Location“ → „Choose“).

### Schritt 2 – Klasse „Ausgabe“ erstellen

Erstelle in Deinem neuen Projekt eine neue Klasse „**Ausgabe**“ und öffne den Quelltext mit einem Doppelklick auf das Klassen-Symbol. Lösche alles zwischen „public class Ausgabe{“ und der letzten geschweiften Klammer ganz unten. Das Ergebnis sollte so aussehen, wie Du es rechts siehst.

```
/**
 * Beschreiben Sie hier die Klasse Ausgabe.
 *
 * @author (Ihr Name)
 * @version (eine Versionsnummer oder ein Datum)
 */
public class Ausgabe {
}
```

### Schritt 3 – Konstruktor erstellen

In unserem Beispiel brauchen wir eigentlich keinen Konstruktor (das zu erzeugende Objekt muss nicht mit Anfangswerten initialisiert werden), sondern lediglich eine Methode (bzw. mehrere). Es ist aber guter Programmierstil, einen Konstruktor zu haben, der dann allerdings nichts tut.

Der Konstruktor besteht aus den folgenden Teilen:

Aufbau eines <b>Konstruktors</b>			
<b>public</b>	<b>Ausgabe</b>	<b>()</b>	<b>{...}</b>
<b>Zugriffsmodifikator</b> (unterschiedliche möglich)	<b>Name des Konstruktors</b> (= Klassenname) Großbuchstabe am Anfang	<b>Parameter-</b> klammern	<b>Rumpf</b> zwischen den beiden geschweiften Klammern

Füge diese Teile nun in deinen Code ein. Was verändert sich dadurch?

### Schritt 4 – Methode zur Ausgabe von Text auf der Konsole

Nun geben wir eine erste Methode ein. Eine Methode besteht – genauso wie ein Konstruktor – aus einem Kopf und aus einem Rumpf. Der Aufbau ist dabei zwar sehr ähnlich wie beim Konstruktor, er unterscheidet sich aber in ein paar entscheidenden Details:

Aufbau einer <b>Methode</b>				
<b>public</b>	<b>void</b>	<b>text1</b>	<b>()</b>	<b>{...}</b>
<b>Zugriffsmodifikator</b> (unterschiedliche möglich)	<b>Ergebnistyp</b>	<b>Name der Methode</b> Kleinbuchstabe am Anfang (Name einzigartig)	<b>Parameter-</b> klammern	<b>Rumpf</b> zwischen den beiden geschweiften Klammern

Es gibt zwei/drei Unterschiede zwischen der Definition eines Konstruktors und der einer Methode. Welche?

Füge nun die Methode (soweit sie hier bisher dargestellt ist) unterhalb des Konstruktors, aber vor der letzten geschweiften Klammer am Ende des Codes ein.

Das eigentliche „Ausprogrammieren“ des Codes findet nun im Rumpf der Methode statt, also zwischen den geschweiften Klammern der Methode.

In unserem Beispiel wollen wir, dass ein Text auf der Konsole ausgegeben wird. Der Befehl hierzu lautet:

```
System.out.println("Einen wunderschönen guten Morgen :)");
```

Vergiss nicht den Strichpunkt am Ende!

Um das Programm nun auszuführen, müssen wir den Code zuerst kompilieren (→ Übersetzen/Compile). Wenn der Übersetzungsvorgang ohne Fehler abgeschlossen ist, sollte unten „*Klasse übersetzt – keine Syntaxfehler*“ stehen und deine Klasse nicht mehr schraffiert dargestellt werden.

Nun kannst Du ein Objekt der Klasse „Ausgabe“ erzeugen und darauf dann die Methode „text1“ aufrufen. Der Text erscheint nun in der sogenannten Konsole, dem Ausgabefenster für Text.

**Aufgabe:** Füge nun selbständig noch zwei weitere Methoden („text2“ und „text3“) hinzu, bei deren Aufruf jeweils ein anderer Text auf der Konsole ausgegeben wird.

Im Konsolenfenster kannst Du unter „Optionen“ die bisherigen Ausgaben löschen, die Methodenaufrufe protokollieren und andere Einstellungen vornehmen. Probiere verschiedene Einstellungen aus!

### Schritt 5 – Das Ergebnis einer Rechenaufgabe ausgeben

Ziel: Ein Nutzer gibt zwei Zahlen ein. Als Ergebnis soll der Wert ihres Produktes zurückgegeben werden.

Erzeuge eine neue Klasse mit dem Namen „Rechnung“, lösche alles innerhalb des Klassenrumpfes und lege dann einen (leeren) Konstruktor zur Klasse an.

Dann erstellen wir die Methode „multiplikation“ mit dem Zugriffsmodifikator „public“ und dem Ergebnistyp „int“. Der Kopf der Methode ist also: `public int multiplikation()`

Als Parameter soll der Benutzer beim Aufruf der Methode zwei Zahlen eingeben. Dies erreicht man durch entsprechende Einträge in den Parameterklammern.

Methodenkopf mit Parametern: `public int multiplikation(int zahl1, int zahl2)`

Im Rumpf der Methode wird die eigentliche Funktion programmiert. Wir möchten, dass das Ergebnis der Multiplikation der vom Nutzer eingegebenen Zahlen zurückgegeben wird.

Methodenrumpf: `return zahl1 * zahl2;`

Übersetze nun die Klasse, erzeuge ein Objekt davon und rufe die Methode „multiplikation“ auf. Teste sie mit verschiedenen Zahlen.

Bestimmt ist Dir aufgefallen, dass das Ergebnis nicht auf der Konsole ausgegeben wird, sondern in einem „Methoden-Ergebnis-Fenster“. Wir wollen nun das Ergebnis mit weiterem Text auf der Konsole ausgeben.

### Schritt 6 – Das Ergebnis einer Rechenaufgabe auf der Konsole ausgeben

Füge die folgende Methode in den Quellcode der Klasse Rechnung unterhalb von der bereits vorhandenen Methode „multiplikation“ ein. (Die hier erscheinenden Zeilenumbrüche bei **\*\*** musst Du löschen!)

```
public void multiplikationAusgabe(int zahl1, int zahl2) {  
    int produkt = zahl1 * zahl2;  
    System.out.println("Das Produkt aus den Zahlen "+zahl1+" und "+zahl2+" hat  
    ** den Wert "+produkt+".");  
}
```

Übersetze nun die Klasse, erzeuge ein Objekt davon und rufe die Methode „multiplikationAusgabe“ auf.

Wir analysieren den Methodenrumpf:

`int produkt` erzeugt eine lokale Variable vom Typ `int` (ganze Zahl), die als Wert den Produktwert `zahl1 * zahl2` zugewiesen bekommt. Hier wird also zunächst der Produktwert berechnet.

`System.out.println` bewirkt eine Ausgabe auf der Konsole. „`ln`“ bedeutet „line new“ (neue Zeile).

In den runden Klammern nach `println` steht, was ausgegeben werden soll. Fest vorgegebener Text wird in Anführungszeichen gesetzt. Mit dem Pluszeichen lassen sich Text-Teile aneinanderreihen. Das entspricht der Funktion „VERKETTEN“ in Excel. Variablen werden ohne Anführungszeichen eingegeben. Beim Methodenaufruf werden die Variablen durch ihren jeweils aktuellen Wert ersetzt. Das Ergebnis der Multiplikation, also der Wert des Produktes, wird hier durch das zuvor berechnete `produkt` eingefügt.

**Aufgaben:** Füge nun noch drei weitere Methoden hinzu für die Addition (Ergebnis: `summe`), die Subtraktion (Ergebnis: `differenz`) und die Division (Ergebnis: `quotient`) zweier Zahlen. Die Rechenzeichen hierfür sind „+“, „-“ und „/“.

Teste alle vier Methoden mit verschiedenen Zahlen.

Finde insbesondere heraus, was bei der Division „passiert“!

Füge nun noch eine weitere Methode „xAusgabe“ hinzu für eine Rechnung mit dem Rechenzeichen „%“, also `x = zahl1 % zahl2`. Lasse auf der Konsole Folgendes ausgeben: „Die Rechnung % mit den Zahlen "+zahl1+" und "+zahl2+" ergibt "+x+".“  
Finde heraus, wie das Ergebnis der Rechnung mit dem Rechenzeichen „%“ zustande kommt!