

## Java Cheatsheet – 1. Vorlesung

- Compiler-Befehle (Java Programme in der Command Line Ausführen)
  - `javac dateiname.java` (erzeugt bytecode, der in einer `.class` datei gespeichert wird)
  - `java dateiname` (führt das programm aus)
  - erst `javac`-befehl dann `java`-befehl
- Klassenaufbau:
  - `public class klassenname{`
    - Methoden und Attribute(Zahlen oder Zeichen(ketten)) hier
  - `}`
- Methodenaufbau
  - `public (static) Rückgabewert(void/int/String/...) Methodenname(Parameter(int Parameter1, String Param2)`
    - `{`
      - Variablen und Befehle hier
    - `}`
  - Main-Methode immer: `public static void main(String[] args){`
    - Variablen und Befehle hier
  - `}`
- Vergleichsoperatoren und Verknüpfungen(Beispiel siehe bsp1.java)
  - Vergleichsoperatoren: `==`, `<=`, `>=`, `<`, `>` `.equals`(um Strings zu vergleichen)
  - Verknüpfungen für Bedingungen: `||`(Logisches Oder) `&&`(Logisches und)
  - Inkrementor und Dekrementor(Variablen in einer Zeile um eins kleiner oder größer machen)
    - Inkrementor vor Ausführung: `++a`
    - Inkrementor nach Ausführung: `a++`
    - Dekrementor vor Ausführung: `--a`
    - Dekrementor nach Ausführung: `a--`
  - Einer variablen wird mit einem einfachen `=` ein Wert zugewiesen
- Kommentare:
  - `//` Einzeiliger Kommentar
  - `/*` `*/` Beliebig langer Kommentar

## Java Cheatsheet – 2. Vorlesung

- Datentypen:
  - `byte` (kurze Ganzzahl)
  - `short` (mittellange Ganzzahl)
  - `int` (lange Ganzzahl)
  - `long` (sehr lange Ganzzahl)
  - `float` (kurze Kommazahl)
  - `double` (lange Kommazahl)
  - `char` (Zeichen, z.B. `'a'` oder `'?'`, oder `'2'`)
  - `boolean` (Speichert die Werte wahr oder falsch)
  - `String` (Zeichenkette, Referenzdatentyp(Objekt einer Klasse))
- Variablendeklaration
  - Um eine Variable zu verwenden muss man sie deklarieren und initialisieren

- Beispiel: Deklaration und Definition für einfache Datentypen (das gleiche):

```
int myVar;    // deklariert und definiert die Variable myVar;
int i, j, k;  // deklariert und definiert die Variablen i, j, k;
```

- Initialisierung: Festlegung eines initialen Wertes der Variable

```
int counter = 0;  
int i = counter + 1;
```

- Scope
  - Eine Variable ist nur innerhalb eines bestimmten Gültigkeitsbereiches(Scopes) sichtbar
  - Wenn eine Variable innerhalb einer Methode deklariert wurde, ist Sie nur dort sichtbar
  - Wenn eine Variable innerhalb einer Klasse deklariert wurde, ist Sie in der ganzen Klasse sichtbar
- Typecast (Einen Datentyp in einen anderen umwandeln, siehe bsp2)
  - Expliziter Cast: Zieldatentyp in Klammern vor den umzuwandelnden Wert schreiben

```
float f = 3;  
int i = (int)f; // Expliziter Cast mit Angabe des Typen
```