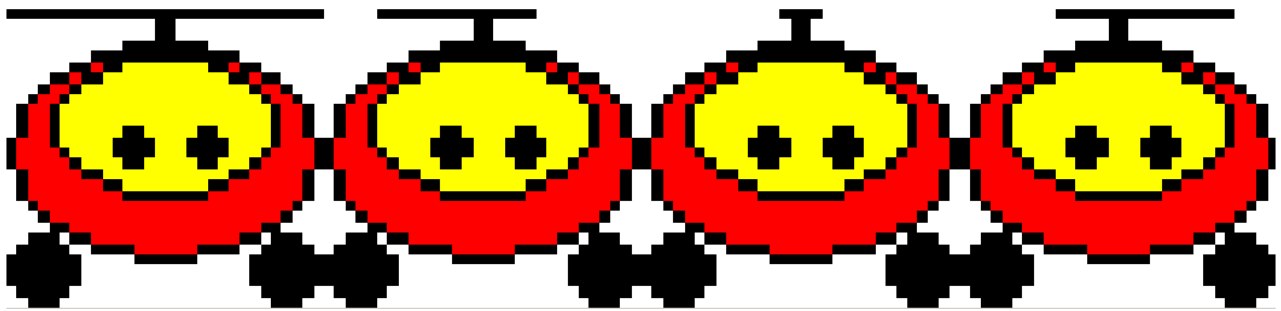


# Spiel mit Programmieren mit Java

für Anfänger

unter Linux Fedora Gnome



Schule  
Auftragnehmer  
Auftraggeber  
Bildungsgang  
Klasse  
Fach  
Bearbeitungszeitraum

Johann-Conrad-Schlaun-Gymnasium  
Dean Schneider  
Herr M. Schmidt  
Berufliches Gymnasium Technik  
GYTU  
Informatik  
18.08.2023 – 05.02.2024

# Inhaltsverzeichnis

Erste schritte	3
1. Wichtige hinweise	3
2. Benötigte Programme installieren	3
3. Projekt erstellen	4
Keywords	6
Datentypen	6
Syntax	7

# Erste schritte

## 1. Wichtige hinweise

Fedora, das Betriebssystem, und alle Programme sind auf Englisch. Wenn man etwas online sucht, sucht man besser auf Englisch, weil es viel mehr Ergebnisse gibt. Die Anleitungen haben ihr System dann auch auf Englisch, was es einfacher macht ihnen zu folgen.

## 2. Benötigte Programme installieren

Programme über den Appstore:

- Office z.B. LibreOffice
- IDE (Entwicklungsumgebung) z.B. NetBeans
- Java Runtime/Development JDK
- UML Diagramme z.B. Dia

Ich benutze NetBeans IDE 19 anstatt von 8.2, weil es für Linux scheinbar keinen 8.2 Download mehr gibt.

```
sudo dnf install ant
```

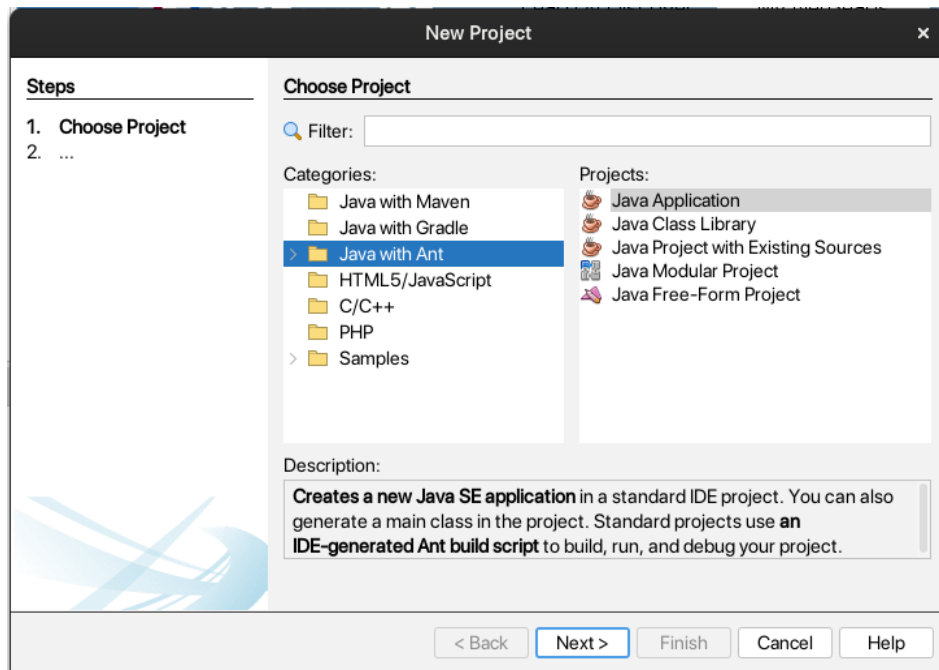
Um aus .java Dateien eine startbare Datei zu machen, verwende ich Ant. Dieser Vorgang wird „Kompilieren“ genannt.

### 3. Projekt erstellen

1.



2.



3. Project Name:           GamePanel  
Project Location:       an einem gutem Ort speichern  
Create main class:       Ja
4. Die .java Datei sollte so aussehen:

```
package gamepanel;

public class GamePanel
{
    // Bei main() beginnt jedes Programm
    public static void main(String[] args)
    {
        /**
         * Hier wird „Hello World“ in der Console ausgegeben
         * ln (line) steht für neue Zeile nach der ausgabe
         */
        System.out.println("Hello World");
    }
}
```

5. Den Code Kompilieren (Startbar machen). Öffne zuerst den Terminal in dem Projektordner.

```
ant -f .
java -jar dist/GamePanel.jar
```

## **Temporär: Hausaufgabe**

Zuhause hab ich das Projekt soweit fertiggestellt das ich nur noch die Kommentare in der Schule fertigstellen muss. In der Schule traten dann wie so oft unerwartete Fehler auf. Zwei Fehler, einen davon konnte ich lösen für den andere hat die Zeit nicht gereicht. Ich habe das Projekt in Github gespeichert und bei dem runterladen auf diesen Laptop gab es bei der Anmeldung Probleme die eine Stunde gedauert haben. Das zweite Problem war das die Java Version nicht übereingestimmt hatte. Ich werde zuhause den rest bis zur nächsten Stunde aufarbeiten und die Dokumentation weiterschreiben.

# Spickzettel

## 1. Keywords

;	Befehl ende – beendet ein Befehl
//	Kommentar – Diese Zeile wird vom Programm ignoriert
/** **/	Kommentar Block – Kommentar über mehrere Zeilen
{}	Block/Scope – In dem Scope erstellte Variablen werden bei dem Ende gelöscht
[]	Array – Liste von Elementen

## 2. Datentypen

Keyword	Bits	Bytes	Beschreibung	Wertebereich
boolean	1	1	wahr oder falsch, true oder false	0 oder 1
byte	8	1	ein Byte	0 bis 255
char	16	2	ein Buchstabe	
String			Array von Buchstaben	
short	16	2	ganze Zahlen	-32768 bis 32767
int	32	4	Integer - positive und negative ganze Zahlen	-2 147 483 648 bis 2 147 483 647
long	64	8	Für besonders große positive und negative ganze Zahlen	$2^{64}$
double	64	8	Zwei ints um eine Kommazahl darzustellen	Geignet für 6 Nachkommastellen
float	32	4	Die gängige Variante um Kommazahlen darzustellen	Geignet für 15 Nachkommastellen

Die Zahlentypen in Java sind standardmässig signed, positiv und negativ.

## 3. Funktionen

Funktionen können Code wiederholungen verhindern.

```
public int foo(int x)
{
}

pub
```

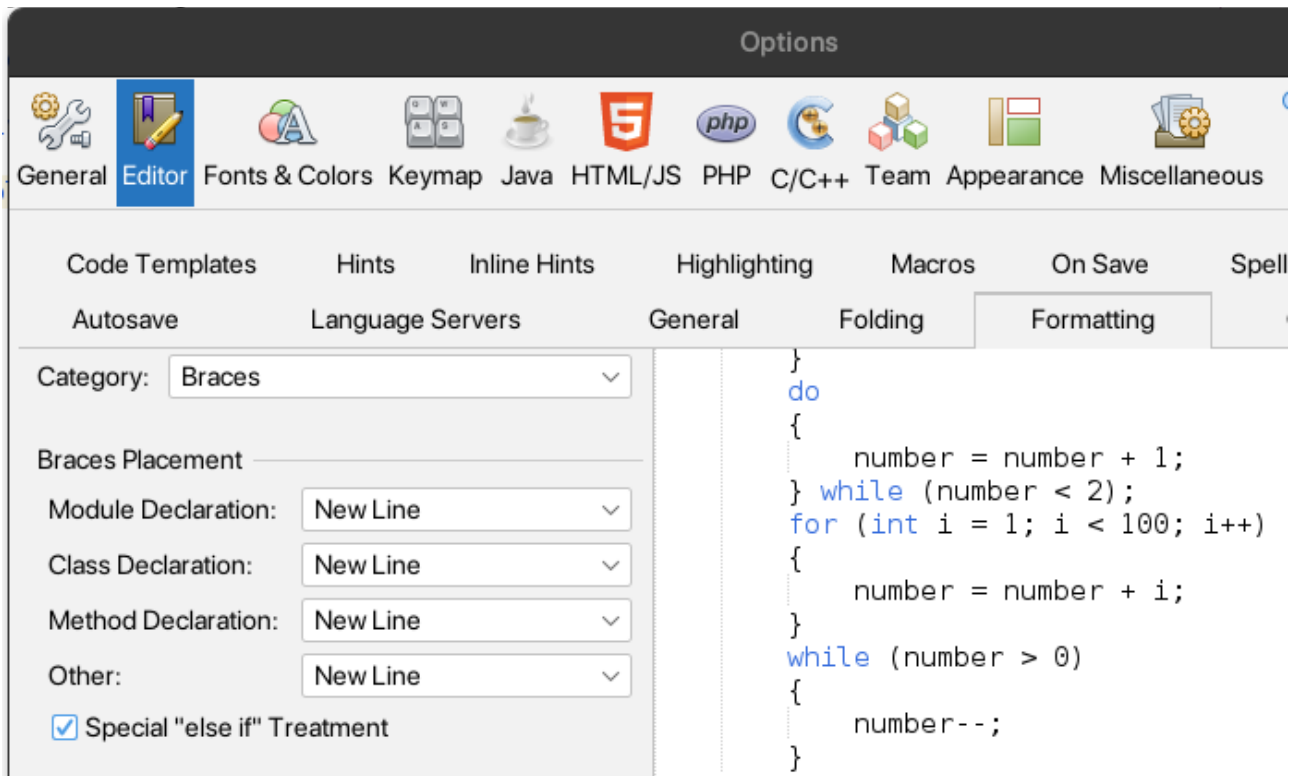
## 4. Classes

## 5. Schleifen

## 6. Threads

# Syntax

Die automatische Code Formatieren kann hier geändert werden:  
Tools -> Options -> Editor -> Formatting



```
public class GamePanel
{
}
```

Curly brackets müssen für einen übersichtlicheren Code untereinander geschrieben werden.  
Klassennamen werden Groß geschrieben werden.