Java Grundlagen 1

Bioinformatik Bachelor
Propädeutikum WS 2019/2020
~~15.~~ 22. Oktober 2019
Leopold Endres

1. Allgemeines

- Wann und wo? Jeden Dienstag im CIP - Pool
  - Gruppe 1 von 10:15 bis 11:45 Uhr
  - Gruppe 2 von 12:00 bis 13:30 Uhr
- Folien und Übungen unter https://bioinformatik-muenchen.com/studium/propaedeutikum-programmierung-in-der-bioinformatik/
- Bitte alle in den Rosalind Kurs einschreiben unter http://rosalind.info/classes/enroll/4efd422b89/
- Weitere Hilfestellung unter w3schools: https://www.w3schools.com/java/default.asp

Outline

1. Allgemeines
2. Was ist Java?
3. Integrated Development Environment
4. Eclipse oder IntelliJ IDEA installieren
5. Mein erstes Programm
6. Primitive Datentypen
7. if-else-Anweisungen

2. Was ist Java?

- Objektorientierte Programmiersprache
- Erschienen im Jahr 1995
- Plattformunabhängig
- Auf über 3 Milliarden Geräten weltweit installiert

3. Integrated Development Environment

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

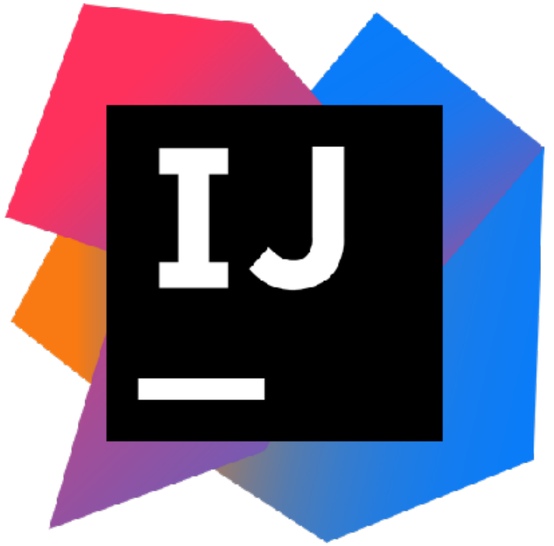
- Kurz “IDE”
- Deutsch: integrierte Entwicklungsumgebung
- Was ist eine IDE?
  - Sammlung wichtigster Tools (Texteditor, Debugger, Compiler, Interpreter) für die Entwicklung von Software
  - Unterschiedliche IDEs für jede Programmiersprache
  - Beliebte IDEs für Java: Eclipse, IntelliJ IDEA

4. Eclipse oder IntelliJ IDEA installieren

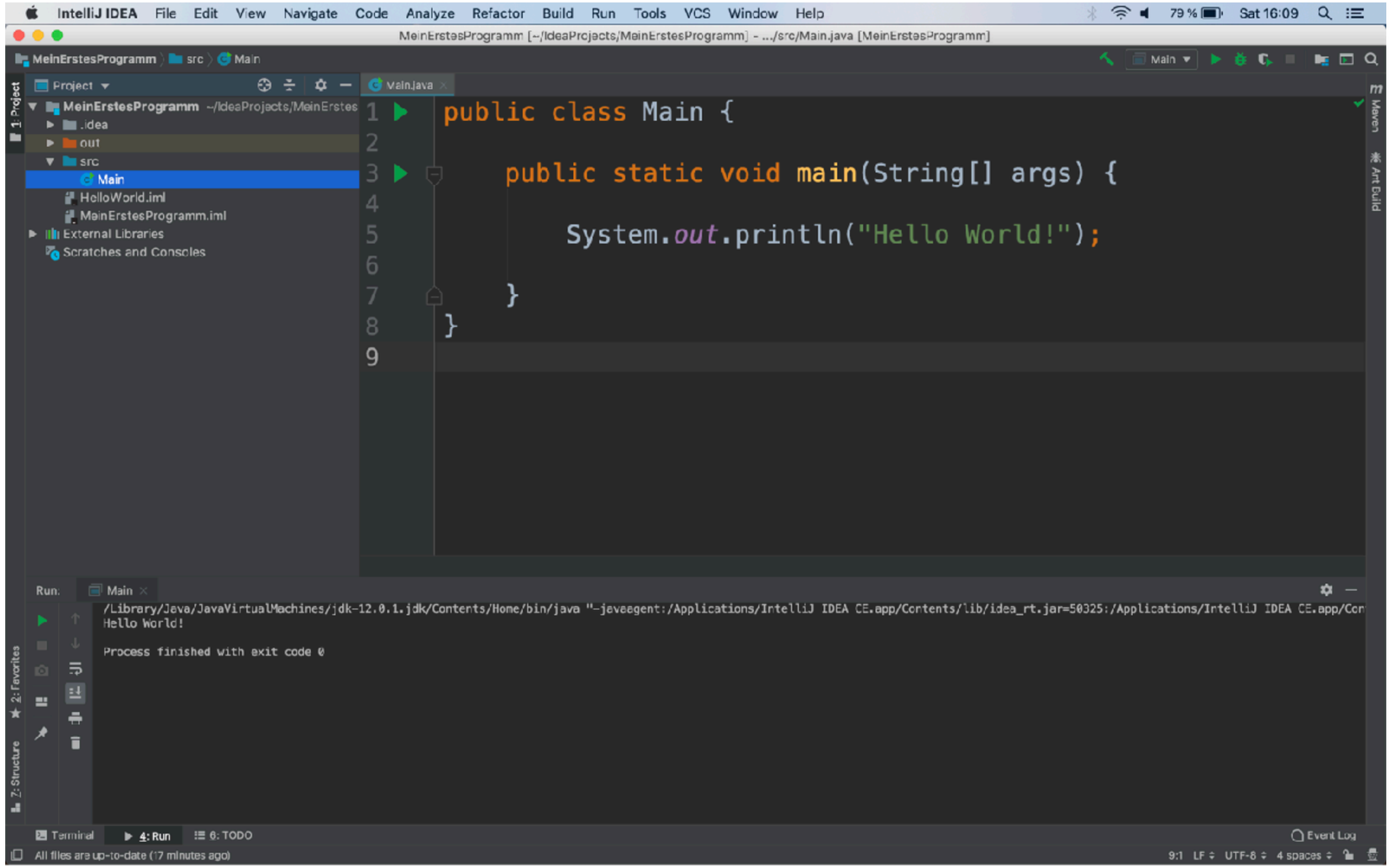
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

1. Java Development Kit (JDK):
   https://www.oracle.com/technetwork/java/javase/downloads/index.html

2. Download Eclipse: https://www.eclipse.org/downloads/ oder
   IntelliJ IDEA: https://www.jetbrains.com/idea/download/



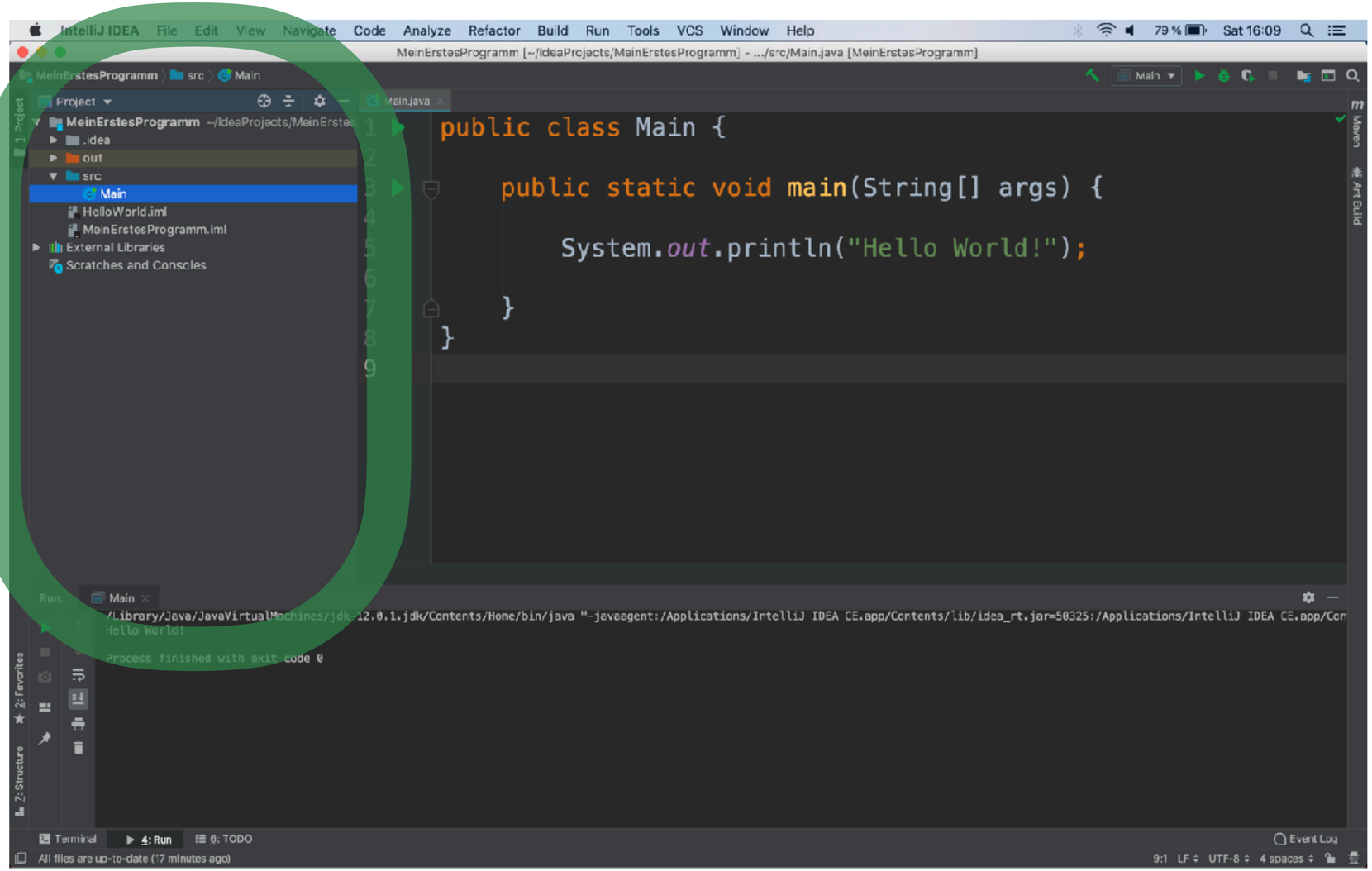
5. Mein erstes Programm



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

6. Programm mit Klick auf ▶ ausführen

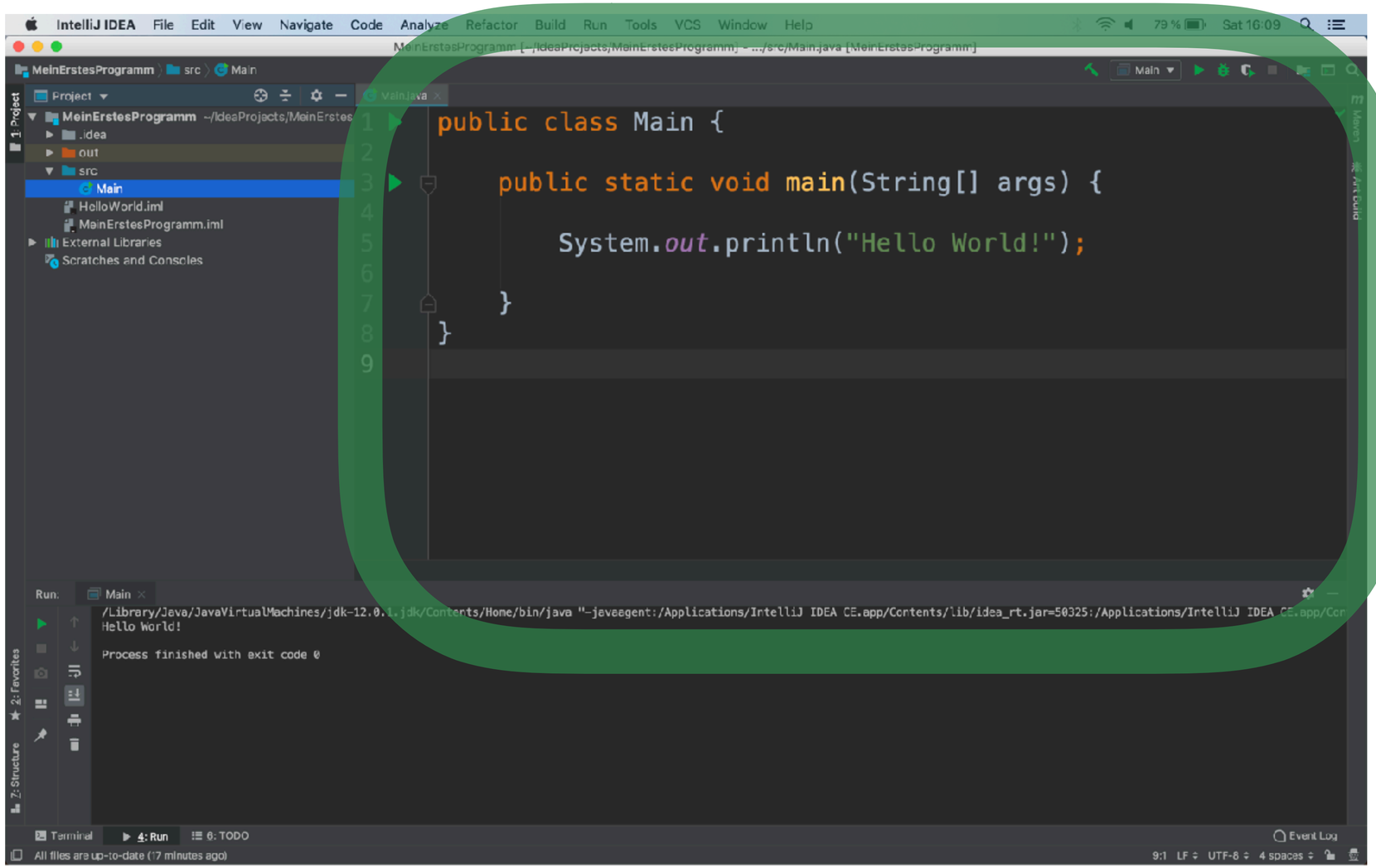
IntelliJ IDEA



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Projekte, Ordner und Dateien

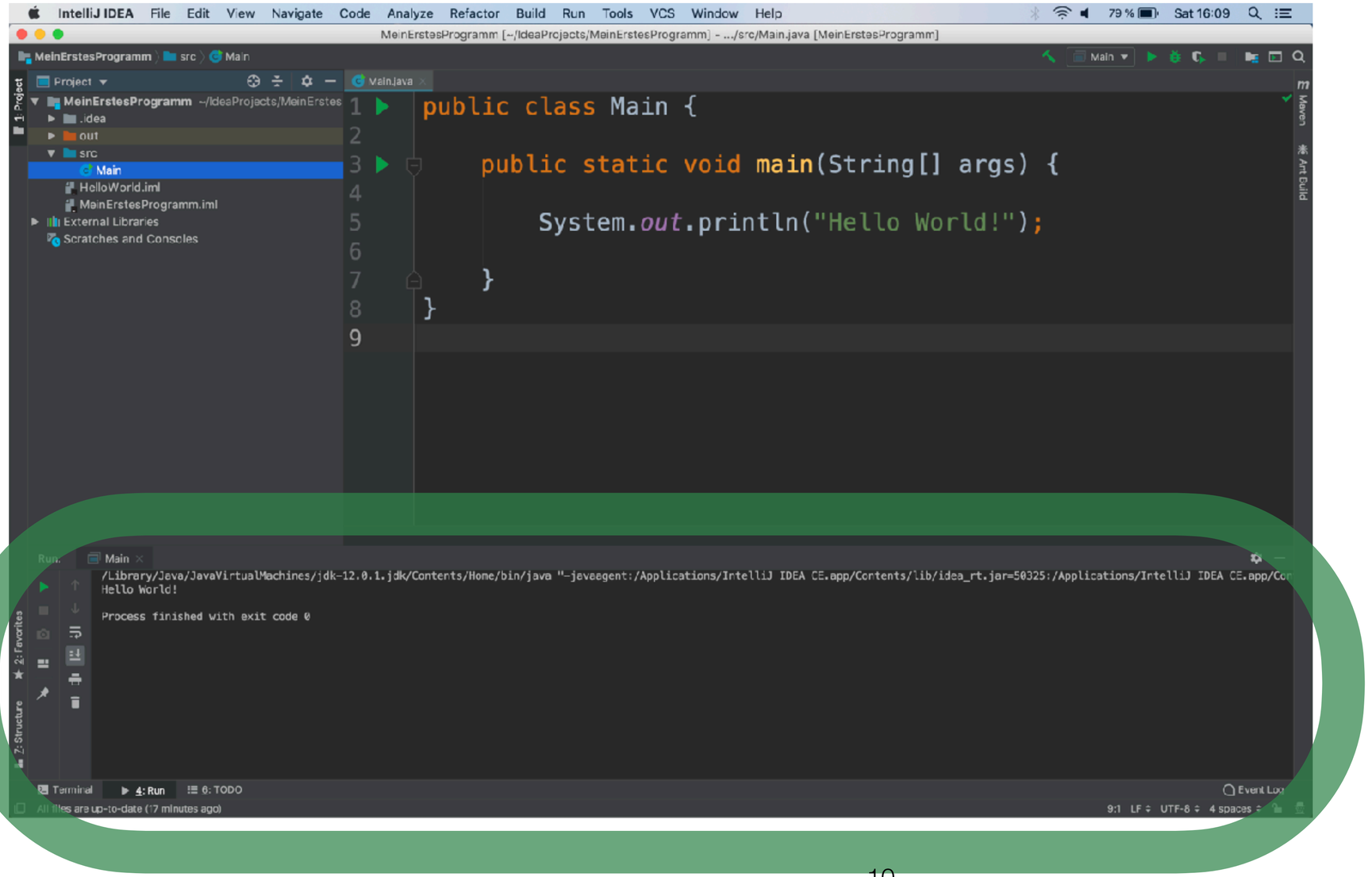
IntelliJ IDEA



Texteditor

IntelliJ IDEA

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Konsole

Mein erstes Programm

Main.java

```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Klasse

Main.java

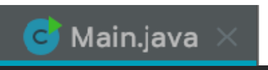
```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Jedes Java Programm beginnt mit einer Klasse.

Jede Zeile Code muss innerhalb einer Klasse sein.

Die Klasse muss den gleichen Namen haben wie die Datei.

“main” - Methode




```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Jedes Java Programm hat eine “main”-Methode.

Code innerhalb der “main”-Methode wird ausgeführt.

Kommentare

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main.java

```java
public class Main {
    // Das ist ein  einzeiliges Kommentar

    /* Das ist ein
    Mehrzeiliges Kommentar*/
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

Kommentare werden nicht ausgeführt.

Einzeilige Kommentare mit //

Mehrzeilige Kommentare beginnen mit /*
und enden mit */

System.*out*.println();

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main.java

```java
public class Main {
    // Das ist ein Kommentar
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

```java
System.out.println("Hello World!");
```

Gibt den Text “Hello World!” auf der Konsole aus.

Jeder Java-Befehl endet mit ;

6. Primitive Datentypen

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- In Java gibt es unterschiedliche Arten von Datentypen:
  - String - speichert Text wie “Hallo”, beginnt und endet immer mit doppelten Anführungszeichen “”
  - int - speichert ganze Zahlen ohne Dezimalen wie 42 or -44
  - double - speichert Gleitkommazahlen wie 4.66
  - char - speichert einzelne *characters* wie ‘v’ oder ‘K’, umgeben von einfachen Anführungszeichen ‘’
  - boolean - speichert einen Wahrheitswert, true oder false

6. Primitive Datentypen

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main.java

```java
public class Main {

    public static void main(String[] args) {

        String name = "Max Mustermann";
        int alter = 19;
        double groesse = 1.85;
        char lieblingsbuchstabe = 'b';
        boolean immatrikuliert = true;

    }
}
```

Um ein Variable zu erstellen, muss man den Datentype spezifizieren und der Variable einen Werte zuweisen.

Syntax:
[Datentyp] [Name der Variable] = [Wert der Variable]

Beispiel:
```java
int alter = 19;
```

6. Primitive Datentypen

Main.java

```java
public class Main {

    public static void main(String[] args) {

        String name = "Max Mustermann";
        int alter = 19;
        double groesse = 1.85;
        char lieblingsbuchstabe = 'b';
        boolean immatrikuliert = true;
        System.out.println("Alter: " + alter);
    }
}
```

Um eine Variable auf der Konsole auszugeben, verwende die System.out.println() - Methode

Syntax:
System.out.println([Variable])

Beispiel:
`System.out.println("Alter: " + alter);`

Console:
Alter: 19

7. *if-else*-Anweisungen

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Java unterstützt folgende logische Vergleichsoperatoren:

Operator	Bedeutung
==	gleich
!=	ungleich
<	kleiner
<=	kleiner oder gleich
>	größer
>=	größer oder gleich

7. *if-else*-Anweisungen

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Expression is true.

```
int test = 5;

if (test < 10)
{
  // codes
}

// codes after if
```

Expression is false.

```
int test = 5;

if (test > 10)
{
  // codes
}

// codes after if
```

7. *if-else*-Anweisungen

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Expression is true.

```
int test = 5;

if (test < 10)
{
   // body of if
}
else
{
   // body of else
}
```

Expression is false.

```
int test = 5;

if (test > 10)
{
  // body of if
}
else
{
  // body of else
}
```

7. *if-else*-Anweisungen

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main.java

```java
public class Main {

    public static void main(String[] args) {

        int alter = 16;

        if(alter < 18){
            System.out.println("Kein Alkohol für dich!");
        }

    }
}
```

Benutze if um einen Code auszuführen, wenn eine Kondition erfüllt ist.

Syntax:
If ([Kondition]) {
  //Code, der ausgeführt wird
}

Beispiel:

```java
if(alter < 18){
    System.out.println("Kein Alkohol für dich!");
}
```

Console:
Kein Alkohol für dich!

7. *if-else*-Anweisungen

Main.java

```java
public class Main {

    public static void main(String[] args) {

        int alter = 18;

        if(alter < 18){
            System.out.println("Kein Alkohol für dich!");
        }else{
            System.out.println("Ready to Go!");
        }

    }
}
```

Benutze else nach if um Code auszuführen, wenn die if Kondition NICHT erfüllt ist.

Syntax:

```
if ([Kondition]) {
  //Code, wenn Kondition wahr
}else{
  //Code, wenn Kondition falsch
}
```

Beispiel:

```java
if(alter < 18){
    System.out.println("Kein Alkohol für dich!");
}else{
    System.out.println("Ready to Go!");
}
```

Console:
Ready to Go!

Fragen?

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN