

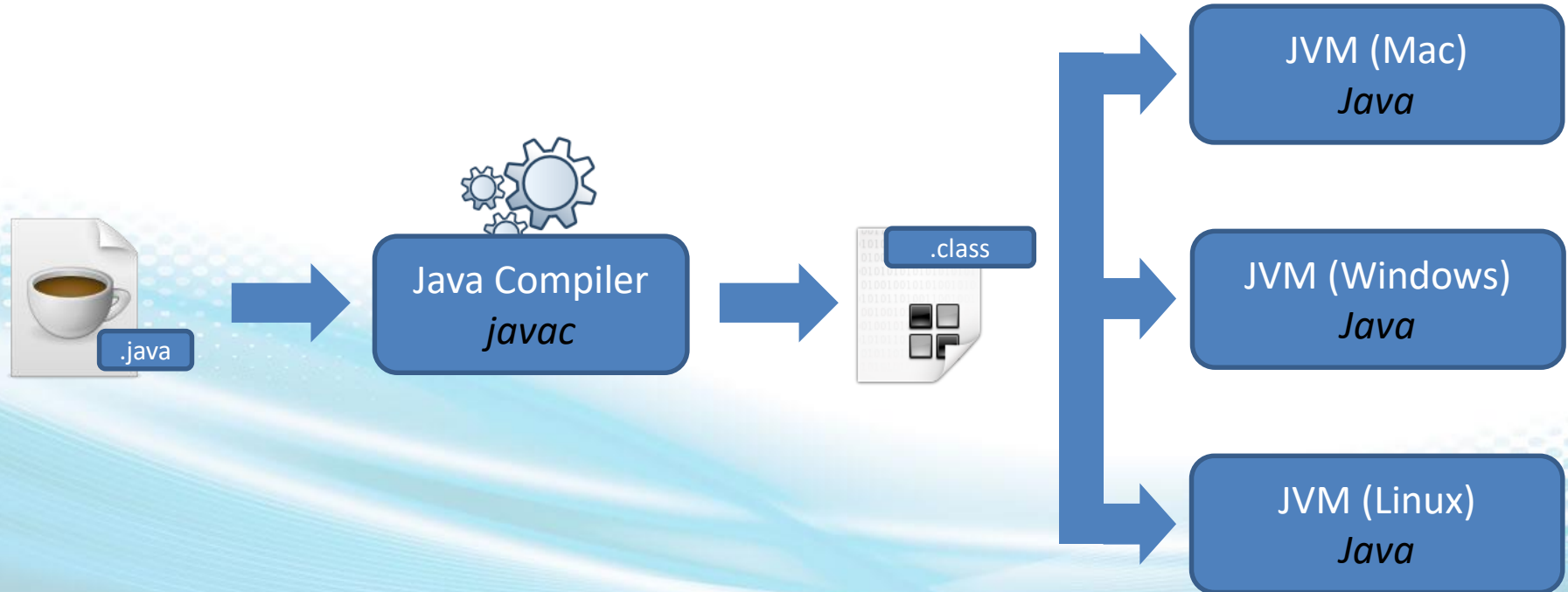
# Java-Grundlagentraining Tag 1

# Agenda

---

- Wie funktioniert Java
- Aufgabe 1 - „Hello World“ im Texteditor
- JRE/JDK Unterschiede
- Aufgabe 2 - Einrichtung IntelliJ IDEA
- Aufgabe 3 - „Hello World“ in der IDE
- Bezeichner / Begrifflichkeiten
- Operatoren
- Datentypen
- Kontrollstrukturen
- JavaDoc
- Hausaufgabe

# Wie funktioniert Java



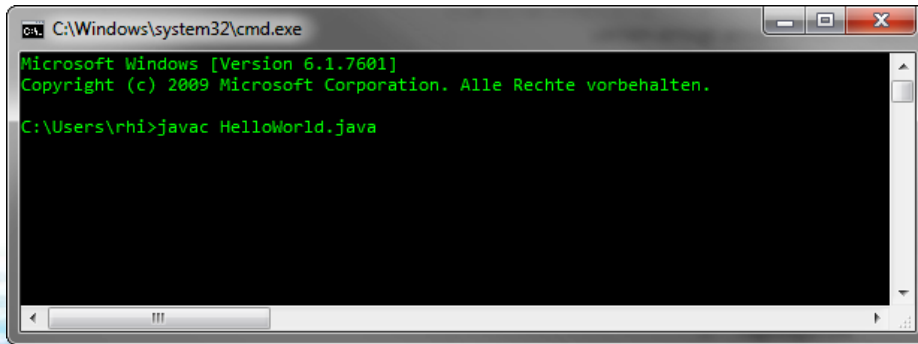
# Aufgabe 1 - „Hello World“ im Texteditor

HelloWorld.java

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



# Aufgabe 1 - „Hello World“ im Texteditor

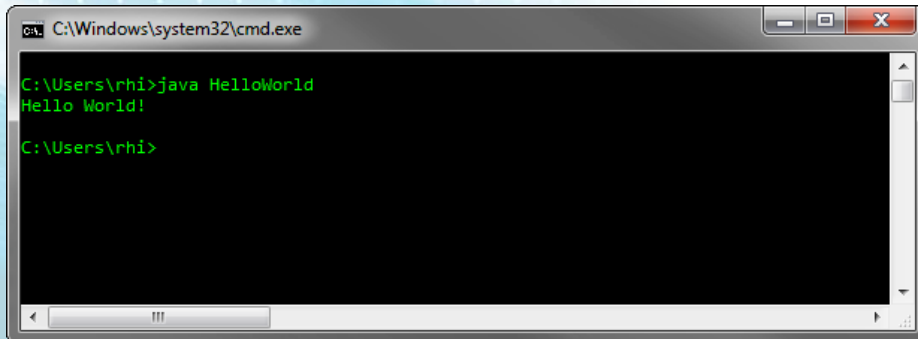


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\rhi>javac HelloWorld.java
```



Java Compiler  
*javac*



```
C:\Windows\system32\cmd.exe

C:\Users\rhi>java HelloWorld
Hello World!

C:\Users\rhi>
```

JVM  
*java*

6 / 20

# JRE/JDK Unterschiede

---

- **JRE (Java Runtime Environment)**
  - für Anwender
  - Beinhaltet die JVM (Java Virtual Machine)
- **JDK (Java Development Kit) auch SDK (Software Development Kit)**
  - für Entwickler
  - Beinhaltet JRE
  - Beinhaltet Compiler
  - Beinhaltet weitere Entwickler-Tools

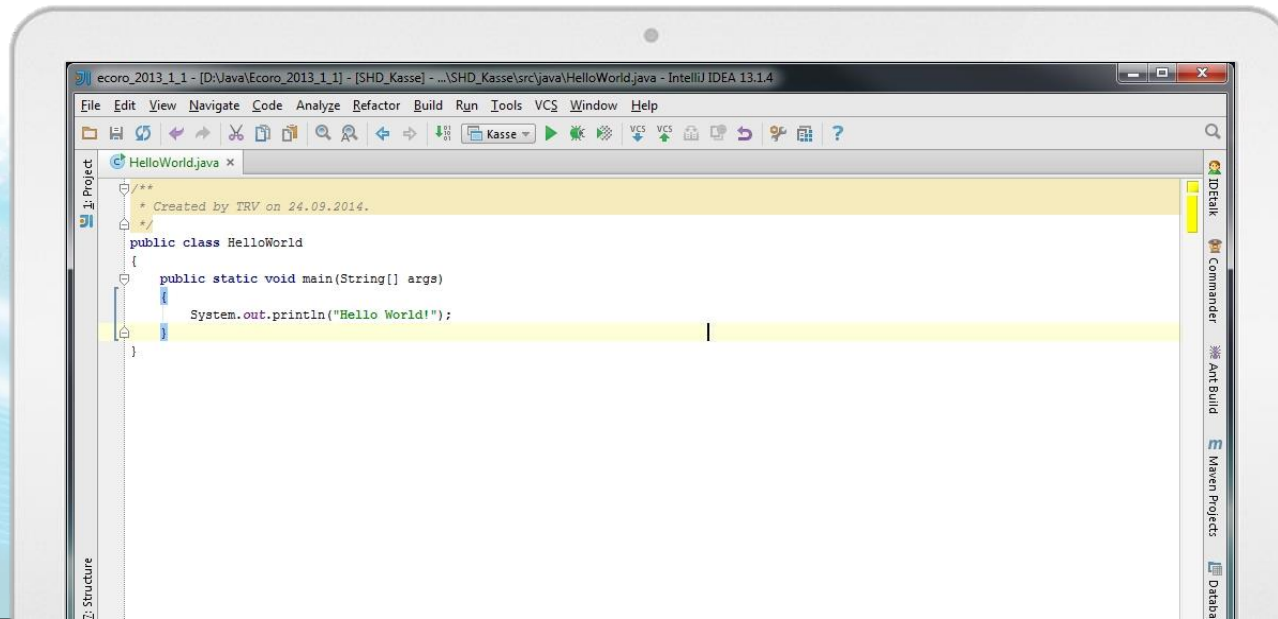


## Aufgabe 2 - Einrichtung IntelliJ IDEA





## Aufgabe 3 - „Hello World“ in der IDE



# Bezeichner / Begrifflichkeiten

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



# Operatoren

Arithmetische Operatoren		Vergleichsoperatoren		Boolesche Operatoren	
+	Addition	==	gleich	!	Negation
-	Subtraktion	!=	ungleich	&&	logisches UND
*	Multiplikation	>	größer als		logisches ODER
/	Division	<	kleiner als	^	exklusiv ODER
%	Modulo (Rest)	>=	größer gleich		
++	Inkrement	<=	kleiner gleich		
--	Dekrement				

# Datentypen

---

Datentypen sind Typen für Variablen.

- Primitive Datentypen
  - char, long, int, short, byte, float, double, boolean
  - Besitzen „Wrapperklassen“
  - Sind nicht nullable
- Komplexe Datentypen
  - Beliebige Klasse
  - Sind nullable

# Kontrollstrukturen

- Bedingungen mit *if / else*

```
boolean isBedingungErfuellt = true;  
if (isBedingungErfuellt)  
{  
    System.out.println("Bedingung ist erfüllt!");  
}  
else  
{  
    System.out.println("Bedingung ist nicht erfüllt!");  
}
```



# Kontrollstrukturen

- Bedingungen mit Bedingungsopertator



```
int zahl = 4;  
String geradeText = (zahl % 2 == 0) ? " gerade" : "ungerade";  
System.out.println("Die Zahl " + zahl + " ist " + geradeText);
```

# Kontrollstrukturen

- Schleifen mit *while* (Kopfgesteuert)

```
int i = 0;
while (i < 10)
{
    System.out.println("Hallo " + i);
    i++;
}
```





# Kontrollstrukturen

- Schleifen mit `do – while` (*Fußgesteuert*)

```
int i = 0;
do
{
    System.out.println("Hallo " + i);
    i++;
}
while (i < 10);
```



# Kontrollstrukturen

- Schleifen mit *for*

```
for (int i = 0; i < 10; i++)  
{  
    System.out.println("Hallo " + i);  
}
```



# Kontrollstrukturen

- Switch-Case

```
String variable = "a";  
switch (variable)  
{  
    case "a" :  
        System.out.println("a wurde eingegeben");  
        break;  
    case "b" :  
        System.out.println("b wurde eingegeben");  
        break;  
    default:  
        System.out.println(variable + " wurde eingegeben");  
}
```



# JavaDoc

---

- JavaDoc ist für die Entwickler-Dokumentation gedacht.
  - Beinhaltet Themen wie: Logik, Variablenbeschreibung, Parameterdefinition, Exceptions, etc.
  - Export über die Entwicklungsumgebung als Webseite möglich.
- Parallel dazu wird es immer eine Programmdokumentation für den Anwender geben, wie das Programm bedient wird.
- Syntax Referenz: <http://www.java-doc.de/>

# Hausaufgabe

---

## **Alle gelernten Inhalte in einem Projekt**

Aufgaben:

Versuche jede Aufgabe mit verschiedenen Kontrollstrukturen zu realisieren

- Gebe jeden Durchlauf einer Schleife (10 Durchläufe) aus. Hebe dabei nur den 5. Durchlauf besonders hervor.
- Gebe die ersten 100 Primzahlen auf der Konsole aus.
- Beende eine Schleife während der Ausführung. (Recherche ‚break‘)
- Überspringe jeden 2. Schleifendurchlauf. (Recherche ‚continue‘)
- Dokumentiere jede Methode mit JavaDoc