

Variables and If Else

Tobias Hanf, Maik Göken

October 24, 2022

Learn Programming with Java

Outline

Revision

Getting Started

Variables

Operations

Exercise Block 1

If Else

Exercise Block 2

Revision

Try running the `Hello World` program (with VS Code)

Try running the `Hello World` program (with VS Code)

For whom does it **not** work?



<https://pingo.coactum.de/098124>

Getting Started

Basic Structure of a Java Program

```
1 public class Main {  
2     public static void main(String[] args) {  
3  
4         // Code starts here  
5         <statement>  
6         .  
7         .  
8         .  
9         <statement>  
10  
11     } // end main block  
12  
13 } // end Main block
```

Listing 1: Basic Structure

Comments

Comments will be ignored by the compiler.

Only there to document your code.

```
1 // one line comment
2
3 /*
4    multi-line comment
5 */
6
7 /**
8  * JavaDoc comment
9  *
10 * @author Tobias Hanf
11 **/
```

- Use comments to document your code
- At the beginning: write **What** the code will do
- Later: write clean code and only explain the **Why**
- Helps you to think about your code
- Also helps you to understand your code if you haven't worked on it for a while

Names

- You have to name certain "Things" in Java
- Some restriction apply
- A name must start with:
 - a **letter** (a-z,A-Z) <- preferred
 - or an **_** (Underscore)
 - or a **\$** (Dollar-symbol)
- The rest of the nme can contain:
 - **letters** (a-z,A-Z)
 - **digits** (0-9)
 - **_** (Underscors)
 - **\$** (Dollar-Symbol)
- But cannot be a **reserved** word

Reserved words

1

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

¹https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

Variables

Variables

- Names for memory location
- We have to tell the computer how to interpret the data
- Different kind of variables (in later Units)
- Name **variable** implies that they can change
- Differentiation between **declaration** and **initialization**

Types

Number Types

Type	Size
byte	8 bit
short	16 bit
int	32 bit
long	64 bit

Type	Size
float	32 bit
double	64 bit

Other Types

Type	Values
char	'a', 'G'
String	"Hello, world"

Type	Value
boolean	true, false
void	nothing

Implicit conversion: byte -> short -> int -> long

Declaration and Initialization

```
1 //Declaration
2 <Type> <Name>;
3 int a;
4
5 //Initialization
6 <Name> = <Literal/Variable>;
7 a = 2;
8
9 //Declaration + Initialisation
10 <Type> <Name> = <Literal/Varibale>;
11 int b = 52;
```

Listing 2: Declaration and Initialization

Assignment

```
1 int a; // value of a is undefined
2 int b = 10; // value of b is 10
3
4 a = 20; // value of a is 20
5
6 a = b; // value of a is ?
```

Listing 3: Assignment

Operations

Arithmetic

Operation	Example
+	$3 + 5 == 8$
-	$7 - 2 == 5$
*	$4 * 2 == 8$
/	$7 / 2 == 3$
%(Modulo)	$72 \% 10 == 2$

Comparison

Operation	Mathematical Symbol	Example
>	>	5 > 3
>=	≥	4 >= 4 5 >= 3
<	<	10 < 21
<=	≤	5 <= 5 32 <= 50
==	=	5 == 5 123 == 123
!=	≠	5 != 2 -32 != 32

Logical

Operation	Name	Example
<code>&&</code>	logical AND	<code>a && b && c</code>
<code> </code>	logical OR	<code>a b c</code>
<code>^</code>	logical XOR	<code>a ^ b</code>
<code>!</code>	logical negation	<code>!(a && b)</code>

Operation	Name	Example
~	bitwise negation	~a
&	bitwise AND	a & b
	bitwise OR	a b
^	bitwise XOR	a ^ b
<<	left shift	a << 4
>>	(signed) right shift	a >> b
>>>	(unsigned) right shift	a >>> b

Exercise Block 1

Celsius to Kelvin

1. Download the `TemperatureTemplate` file from the Course Page
2. Create a directory with the name `CelsiusKelvin`
3. Copy the `TemperatureTemplate` file to this directory
4. Open the directory in VS Code
5. Rename the `TemperatureTemplate` to `CelsiusKelvin`
6. Check that class name and package name is `CelsiusKelvin`
7. Research the conversion from Celsius to Kelvin
8. Implement the conversion from Celsius to Kelvin
9. Check your results ($20^{\circ}\text{C} = 293.15\text{K}$)

Celsius to Fahrenheit

1. Repeat steps 2 - 6 from the previous page but with `CelsiusFahrenheit` instead
2. Research the conversion from Celsius to Fahrenheit
3. Implement the conversion
4. Check your results ($20^{\circ}\text{C} = 68^{\circ}\text{F}$)

1. Do the same as on the previous slide but now for the conversion from Fahrenheit to Celsius

Advanced Math

1. Create a new directory and file with the name **AdvancedMath**
2. Create a new class (if not automatically create) with the same name
3. Read the values **x** and **y** from the terminal in the **main** method
4. Calculate the result for the following formula and print it to the terminal

$$z = \frac{xy^2}{107751}(1682y^2 + 29) + \frac{x^3}{107751}(3 - 2x^2) + \frac{832}{107751} \quad (1)$$

$$x = 25, y = 11 \rightarrow z \approx 5533.6677$$

If Else

```
1 if(<condition>) {  
2  
3     <code> // will be execute if condition is true  
4  
5 }
```

Listing 4: If Statement

- Condition must be a boolean expression
 - returning **true** or **false**
- Code will be executed if the **Condition** is **true**
- Code can be (almost) anything
 - nested **ifs** are possible

If Else

```
1 if(<condition>) {  
2  
3     <code> // will be executed if condition is true  
4  
5 } else {  
6  
7     <code> // will be executed if condition is false  
8  
9 }
```

Listing 5: If Statement

- Else **Branch** will be taken if Condition is **false**

If Else-If Else

```
1 if(<condition1>) {  
2     <code> // will be executed if condition1 is true  
3 } else if(<condition2>) {  
4     <code> // will be executed if condition2 is true  
5         // and condition1 was false  
6 } else {  
7     <code> // will be executed if no condition was true  
8 }
```

Listing 6: If Statement

- Can be used instead of multiple If-Else
- Conditions will be checked **in order**

Exercise Block 2

Download the `FizzBuzz`, extract it and open the directory with VS Code.

Implement the `FizzBuzz` game in the `FizzBuzz` file.

The rules are as follows:

- if the input is divisible by 3 -> output `fizz`
- if the input is divisible by 5 -> output `buzz`
- if the input is divisible by 3 and 5 -> output `fizzbuzz`
- else output the input as is

Grade Calculator

Download the `GradeCalculator` file and move it into a directory. Open the directory in VS Code.

The Programm

- Read the total number of points
- And the achieved number of points from the terminal
- Calculate the grade based on the grading table
- Output the resulting grade to the terminal

Grading Table

Percentage	Grade
≥ 93	1.0
88	1.3
81	1.7
76	2.0
71	2.3
64	2.7
59	3.0
54	3.3
47	3.7
42	4.0
< 42	5.0

Quiz

Create a quiz by printing Questions to the terminal, taking an answer from the input and checking it.

The questions can be simple yes or no questions or questions where the user needs to input a number or something entirely different.

Be creative and have fun :^)