# Language Introduction

In this section you will find a number of simple examples, that will introduce the Java programming language. Goal is to cover the most basic features and help you, to start developing your own software as quickly as possible.

## 1. Control

*Controlling what's happening*

This example demonstrate some of Java's features, to make decisions and control your program's flow. Execute the following commands:

```
cd control
mvn clean package
java -jar target/control.jar
```

### Concepts Covered

This example has examples for the following language features:

- if/else -> based on variables or expressions, your program can decide different ways to act
- switch/case -> if you have a multitude of possible choises, based on the value of a variable, this helps you defining them in an easy to understand way.
- loops
  - for -> classic program loop to run from start value (e.g 0) to an end value (like length of an array)
  - for(each) -> Java's way of looping over every element in arrays/collections/...
  - while -> Run a loop for as long a value/expression is true

### Tasks

We want to test different concepts by choosing different Input numbers. Console output should look like this:

```
Select method to test:
(1) Condition
(2) Switch case
(3) For loop
(4) While loop
(default) random choice
```

1. Run app and test with various inputs
2. add a switch case to input method (method name `readingInput`), selecting numbers 1 - 4
3. call the existing methods depending on choice
4. modify readingInput method to run switch case 10 times

5. Write a method, that runs a for loop as often as *input* but breaks if input is divisable by 3 hint break hint modulus
6. instead of running switch case 10 times, let user decide if he wants to run it again

## 2. Data Types

Storing data in variables is essential for any programming language. This example provides help, to understand Java's build-in types and how you use them. Here is an overview of primitive types.

```
cd data
mvn clean package
java -jar target/data.jar
```

### Concepts Covered

This application has examples for the following language features:

- Java's (primitive) datatypes
- Type conversions How to convert types and which rules are applied implicitly
- Operators
- Arrays which allows you, to store multiple things of the same type
- Scopes How scopes define, where variables can be used and where not.

**Please note** As Java is an object oriented programming language, it's type system can be extended and it is in fact the much more interesting part of the language 😃

### Tasks

- Run app
- Modify methods `dataTypes` and `moreDataTypes` method, such that StringBuffer output each value in a seperate line
- Enhance method `calculations` by three arithmetic operators
- Enhance array method with an array of 10 that holds random integersHint for random numbers
  - output random element
  - output integer values and the sum of all values
  - what's happening, using datatype short instead of integer and sum or element value is greater than datatype scope?

## 3. Parameters and Methods

In order to break down software into manageable pieces, handing over parameters is essential. This example project shows you, how to do this on program start and with methods.

```
cd params
mvn clean package
java -jar target/params.jar param1 9 param3
```

### Concepts covered

This example has examples for the following language features:

- Java methods Java's core concept to break code into manageable blocks
- Parameters in Methods How to deal with parameters for methods.
- Overloading Methods On the magic, that multiple methods can have the same name and yet do something different.

### Tasks

1. run the application and explain the methods already written
2. create a method to add 3 numbers given by user input
3. we are doing hiking and bike tours, we want to know the average speed on a tour. Create a method calculating speed by given distance and time. Substract time for breaks.
4. create a method to caculate volume of a cuboid (Quader) from user input
5. create a method to caculate the entire surface of a cuboid

## 3. Object Orientation

The core concept of Java programming language ist object orientation. Hence programming without object orientation in Java is not really possible. You already used predifined Objects like `String`. That's why we give a short introduction for object orientation. More detailed information and complex tasks will be shown in next chapters.

```
cd oop
mvn clean package
java -jar target/data.jar
```

### Concepts covered

- Classes, Objects and Instances
- Attributes
- Methods

### Tasks

- create a class diagram with https://app.diagrams.net/ for class named Cuboid
- add attributes
- add methods for calculating surface and volume
- create class in java code
- create a new cuboid in main class. The attributes should be filled by user input
- calculate and output volume and surface

## 4. Guessing Game

This is a task to repeat learned content. Espacially control flow.

```
cd guessing-game
mvn clean package
java -jar target/guessing-game.jar
```

### Tasks

1. run application and implement missing parts
2. discuss understandability of the code. Can you extract methods?
3. create a class in https://app.diagrams.net/ with attributes and methods
4. Implement class

## 5. Quiz

This is a task to repeat learned content. Espacially multidimensional array and loops.

```
cd quiz
mvn clean package
java -jar target/quiz.jar
```

### Tasks

ChatGPT created the source code for this example.

1. Check if it's running correctly and answer questions
2. Use for each loop instead of for loop.
3. discuss source code. Is it understandable, simple, well documented?
4. Use class instead of multidimensional array
5. Add questions of chapter 1 and let user choose which chapter he wants to play