robocode into Java

New Austrian Coding School

Programming - memo

Almost **everything** in Java is an **object**. Therefore, to do at least **one thing** in Java we need at least **one object**.

What you need to run a Java program

- We write **public class** followed by the class name
- Everything inside the first **curly braces** belongs to that class
- The class needs an action called **main** preceded with the keywords **public static void** followed by the arguments
- The **code** inside its curly braces will be **run**

```
public class IAmADeveloper {
  public static void main(String[] args) {
     System.out.println("I am a developer.");
  }
}
```

Curly braces **{ }** and semicolons **;** are **everywhere** in Java. They express where things **start** and where they **end**. When we **define** something we use **curly braces**. When we **use** something we end the line with a **semicolon**.

Variables

- **Boolean** and **String** stay the same
- **Number** divides itself in several other number types
- **Collections** stay almost the same but need a little bit more of explanation. We will use **List** instead

```
Boolean isReady = true;
isReady = false;

Integer size = 5;
size = size + 7;

String name = "Mittens";

Collection<String> names = Arrays.asList("Paws", name);
Collection<Integer> ages = new ArrayList<>();
```

Variable kinds

There are two variable kinds, **primitive** and **objects**:

- The difference is in **performance**
- **Primitives** use the memory as it is and offer **nothing** special
- **Objects** require much more memory but offer plenty of **traits**
- Primitives start with a **small** letter and Objects start with a **capital** one

```
Boolean isReady = true;
boolean isRegistered = false;

Integer size = 5;
int age = 24;

String name = "Mittens";
char letter = 'a';

Collection<Integer> ages = new ArrayList<>();
String[] words = new String[5];
```

Variable type summary

robocode types	Java types

Number	int and float Integer and Float
String	char String
Boolean	boolean Boolean
Collection	array List

Methods

- Start with the **visibility** keyword
- Continue with the **return type**, **void** if it returns nothing
- Continue with the name
- Continue with the **arguments**
- **Definition** between curly braces
- If they return something (not **void**) the last line uses the **return** keyword

```java
public void askForHelp(String message, Integer times){
  // says the emergency message that many times
}

public String reverse(String word) {
  String reversed = "";
  // some more code goes here...
  return reversed;
}
```

Conditionals

They follow the next rules:

- Start with the **if** keyword
- Define the **condition** between **parentheses**
- **Code** related is written between **curly braces**
- If **more logic** is necessary, it could continue with **else** or **else if**

```java
if (isTimeToChange && isWillingToChange) {
  change();
} else if (isTimeToChange && !isWillingToChange){
  considerChanging();
} else {
  dontChange();
}
```

Loops

There are several types of loops:

- Repeat for each
- Repeat an amount of times
- Repeat while

```java
for (String name : names) {
  System.out.println(name);
}

for (int times = 0; times < 10; times++) {
  System.out.println("Alan!");
}

while(!areWeThere()){
  askAreWeThereYet();
}
```

How to say and listen in Java

○ Instead of say and listen we will have to write on the screen and read from the keyboard
○ The say version of Java is the System.out.println()
○ The listen version of Java is handled by the class Scanner
	`System.out.println("Are you wearing a hat?");`
	
	`Scanner scanner = new Scanner(System.in);`
	`String answer = scanner.nextLine();`