Java

Control Statements and OOP

Tobias Hanf, Manik Khurana

25. Oktober 2021

Java-Course

Overview

1. Recalling last session
    - Calculating
    - Text with Strings
2. Input
3. Control Statements
    - Ite
    - for
    - while
4. Methods

Recalling last session

Blocks

```
1        public class Hello {
2            // prints a "Hello World!" on your console
3            public static void main(String[] args) {
4                System.out.println("Hello World!");
5            }
6            }
7
```

Everything between { and } is a *block*.

Blocks may be nested.

Naming of Variables

- The names of variables can begin with any letter or underscore. Usually the name starts with small letter.
- Compound names should use CamelCase.
- Use meaningful names.

```
1       public class Calc {
2           public static void main(String[] args) {
3               int a = 0; // not very meaningful
4               float myFloat = 5.3f; // also not meaningfull
5               int count = 7; // quite a good name
6
7               int rotationCount = 7; // there you go
8           }
9       }
10
```

Primitive data types

Java supports some primitive data types:

boolean	a truth value (either **true** or **false**)
int	a 32 bit integer
long	a 64 bit integer
float	a 32 bit floating point number
double	a 64 bit floating point number
char	an unicode character
void	the empty type (needed in later topics)

Calculating with *int* i

```
1          public class Calc {
2              public static void main(String[] args) {
3                  int a;
4                  a = 7;
5                  System.out.println(a);
6                  a = 8;
7                  System.out.println(a);
8                  a = a + 2;
9                  System.out.println(a);
10             }
11         }
12
```

Calculating with *int* ii

```
1          public class Calc {
2              public static void main(String[] args) {
3                  int a; // declare variable a
4                  a = 7; // assign 7 to variable a
5                  System.out.println(a); // prints: 7
6                  a = 8;
7                  System.out.println(a); // prints: 8
8                  a = a + 2;
9                  System.out.println(a); // prints: 10
10             }
11         }
12
```

After the first assignment the variable is initialized.

Calculating with *int* iii

```
1   public class Calc {
2       public static void main(String[] args) {
3           int a = -9;
4           int b;
5           b = a;
6           System.out.println(a);
7           System.out.println(b);
8           a++;
9           System.out.println(a);
10      }
11  }
12
```

Calculating with *int* iv

```
1   public class Calc {
2       public static void main(String[] args) {
3           int a = -9; // declaration and assignment of a
4           int b; // declaration of b
5           b = a; // assignment of b
6           System.out.println(a); // prints: -9
7           System.out.println(b); // prints: -9
8           a++; // increments a
9           System.out.println(a); // prints: -8
10      }
11  }
12
```

Calculating with *int* v

Some basic mathematical operations:

Addition	`a + b;`
Subtraction	`a - b;`
Multiplication	`a * b;`
Division	`a / b;`
Modulo	`a % b;`
Increment	`a++;`
Decrement	`a--;`

Calculating with *float* i

```
1          public class Calc {
2              public static void main(String[] args) {
3                  float a = 9;
4                  float b = 7.5f;
5                  System.out.println(a); // prints: 9.0
6                  System.out.println(b); // prints: 7.5
7                  System.out.println(a + b); // prints: 16.5
8              }
9          }
10
```

Calculating with *float* ii

```
1        public class Calc {
2            public static void main(String[] args) {
3                float a = 0.1f;
4                float b = 0.2f;
5
6                System.out.println(((a + b) == 0.3));
7            }
8        }
9
```

Calculating with *float* iii

```
1         public class Calc {
2             public static void main(String[] args) {
3                 float a = 0.1f;
4                 float b = 0.2f;
5
6                 System.out.println(((a + b) == 0.3)); // false
7                 System.out.println((a + b));
8             }
9         }
10
```

Float has a limited precision.

This might lead to unexpected results!

Mixing *int* and *float*

```
1          public class Calc {
2              public static void main(String[] args) {
3                  float a = 9.3f;
4                  int b = 3;
5                  System.out.println(a + b); // prints: 12.3
6                  float c = a + b;
7                  System.out.println(c); // prints: 12.3
8              }
9          }
10
```

Java converts from **int** to **float** by default, if necessary.

But not vice versa.

Strings

A String is not a primitive data type but an object.
We discuss objects in detail in the next section.

```
1        public class Calc {
2            public static void main(String[] args) {
3                String hello = "Hello World!";
4                System.out.println(hello); // print: Hello World
    !
5            }
6        }
7
```

Concatenation

```
1         public class Calc {
2             public static void main(String[] args) {
3                 String hello = "Hello";
4                 String world = " World!";
5                 String sentence = hello + world;
6                 System.out.println(sentence);
7                 System.out.println(hello + " World!");
8             }
9         }
10
```

You can concatenate Strings using the +. Both printed lines look the same.

Strings and Numbers

```
1         public class Calc {
2             public static void main(String[] args) {
3                 int factorA = 3;
4                 int factorB = 7;
5                 int product = factorA * factorB;
6                 String answer =
7                 factorA + " * " + factorB + " = " + product;
8                 System.out.println(answer); // prints: 3 * 7 =
    21
9             }
10        }
11
```

Upon concatenation, primitive types will be replaced by their current value as *String*.

Conclusion

Datatypes

- int, long
- float, double
- String

Hello World example

Input

Taking input from *the User* i

To take input from the user we use the Scanner class. For this we need to import the java.util.Scanner package.

```
1         import java.util.Scanner;
2         public class Input
3         {
4             public static void main(String[] args)
5             {
6                 //do something
7             }
8         }
9
```

Taking input from *the User* ii

To use the Scanner class we create a new Scanner object.
//not telling you what an object means at this stage

```
1        import java.util.Scanner;
2        public class Input
3        {
4            public static void main(String[] args)
5            {
6            Scanner sc = new Scanner(System.in);
7            System.out.print("Please input a number: ");
8            int a = sc.nextInt();
9            System.out.println("Input number = "+a);
10
11           }
12       }
13
```

Taking input from *the User* iii

To use the Scanner class we create a new Scanner object.

//For educational purposes only

```
1   import java.util.Scanner;
2   public class Input
3   {
4       public static void main(String[] args)
5       {
6       Scanner sc = new Scanner(System.in);
7       System.out.print("Please input a number: ");
8       int a = sc.nextInt();
9       System.out.println("Input number = "+a);
10
11      System.out.print("Please input a decimal number: ");
12      Double b = sc.nextDouble();
13      System.out.println("Input number = "+b);
14      }
15  }
```

Taking input from *the User* iv

```
1   import java.util.Scanner;
2   public class Input
3   {
4       public static void main(String[] args)
5       {
6       Scanner sc = new Scanner(System.in);
7       System.out.print("Please input a number: ");
8       int a = sc.nextInt();
9       System.out.println("Input number = "+a);
10      System.out.print("Please input a decimal number: ");
11      Double b = sc.nextDouble();
12      System.out.println("Input number = "+b);
13          System.out.print("Please input a String: ");
14          String c = sc.nextLine();
15          System.out.println("Input String = "+c);
16      }
17  }
```

Control Statements

Control Statements

- if, else, else if
- for
- while

If Then Else

```
if(condition) {
    // do something if condition is true
} else if(another condition){
    // do if "else if" condition is true
} else {
    // otherwise do this
}
```

If Then Else example

```
1  public class IteExample {
2
3      public static void main(String[] args) {
4          int myNumber = 5;
5
6          if(myNumber == 3) {
7              System.out.println("Strange number");
8          } else if(myNumber == 2) {
9              System.out.println("Unreachable code");
10         } else {
11             System.out.println("Will be printed");
12         }
13     }
14
15 }
```

Conditions?

How to compare things:

- `==` Equal
- `!=` Not Equal
- `>` Greater Than
- `>=` Greater or Equal than

Note: You can concatenate multiple conditions with `&&` (AND) or `||` (OR)

for

```
for(initial value, condition, change) {
    // do code while condition is true
}
```

for example

```
1  public class ForExample {
2
3      public static void main(String[] args) {
4          for(int i = 0; i <= 10; i++) {
5              System.out.print("na ");
6          }
7          System.out.println("BATMAN!");
8      }
9
10 }
```

while

```
while(condition) {
    // do code while condition is true
}
```

while example

```
1  public class WhileExample {
2
3      public static void main(String[] args) {
4          int a = 0;
5          while(a <= 10) {
6              System.out.println(a);
7              a++; // Otherwise you would get an endless loop
8          }
9      }
10
11 }
```

Methods

What is a method?

What is a method?

- piece of code which can be reused

What is a method?

What is a method?

- piece of code which can be reused
- can take input from the outside

What is a method?

What is a method?

- piece of code which can be reused
- can take input from the outside
- can return data to the outside

What is a method?

What is a method?

- piece of code which can be reused
- can take input from the outside
- can return data to the outside

Other names:

- function
- procedure
- subroutine

Why use methods?

Why use methods?

- programmers are lazy $\rightarrow$ do more with less code

Why use methods?

Why use methods?

- programmers are lazy $\rightarrow$ do more with less code
- better structure and less changes

Why use methods?

Why use methods?

- programmers are lazy $\rightarrow$ do more with less code
- better structure and less changes
- reduces errors

Why use methods?

Why use methods?

- programmers are lazy $\rightarrow$ do more with less code
- better structure and less changes
- reduces errors
- important for OOP

Introduction to *methods* i

The most basic method

```
1    static void helloMethod(){
2        System.out.println("Hello, method!");
3    }
4
```

Introduction to *methods* ii

Calling the method in main

```
1     class Hello{
2         public static main(String[] args){
3             helloMethod();
4         }
5         static void helloMethod(){
6             System.out.println("Hello, method!");
7         }
8     }
9
```

Introduction to *methods* iii

Giving data into a function (Parameters)

```
1        static void printHello(String input){
2            System.out.println("Hello, " + input + "!");
3        }
4
```

Introduction to *methods* iv

Returning data from the function (Return)

```
1        static String getHello(String input){
2            String hello = "Hello, " + input + "!";
3            return hello;
4            // return "Hello, " + input + "!";
5        }
6
```

Introduction to *methods* v

DEMO

Exercise

Print all even numbers between 1 and 100/n (take user input)

Exercise

Print all even numbers between 1 and 100/n (take user input)

- with while and if

Exercise

Print all even numbers between 1 and 100/n (take user input)

- with while and if
- with for (without if)

Exercise

Print all even numbers between 1 and 100/n (take user input)

- with while and if
- with for (without if)
- with a function to check evenness


GOOD LUCK AND
HAVE FUN!!!!