Primitive types, variables.

Working with console.

If-else statement

Contents

- The Java language
- Setting up working environment
- First java program
- Primitives and variables
- Basic operations
- Statements
- Working with the console
- If-else statement and blocks

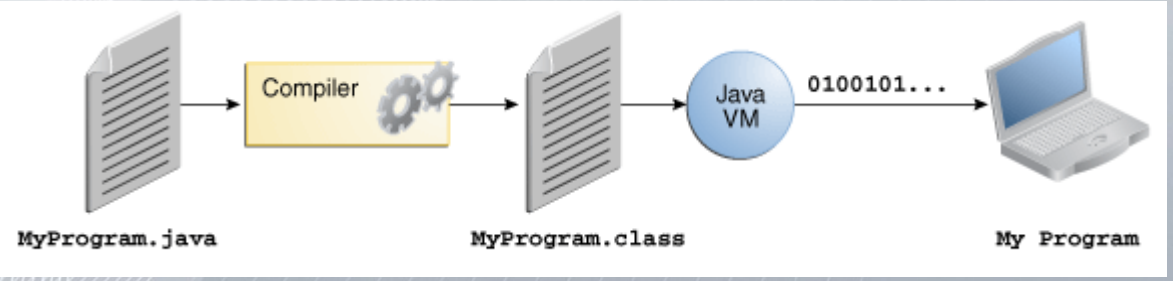
Java language

- What is java as language
  - Developed in 1995 by James Gosling
  - Very widely used programming language
  - Suitable for desktop, web, office applications...
  - Object Oriented language
  - Uses C-like syntax
  - Java is platform independent (programs run on JVM)
- Java runtime environment (JRE)
- Programmers use JDK

Java compiler

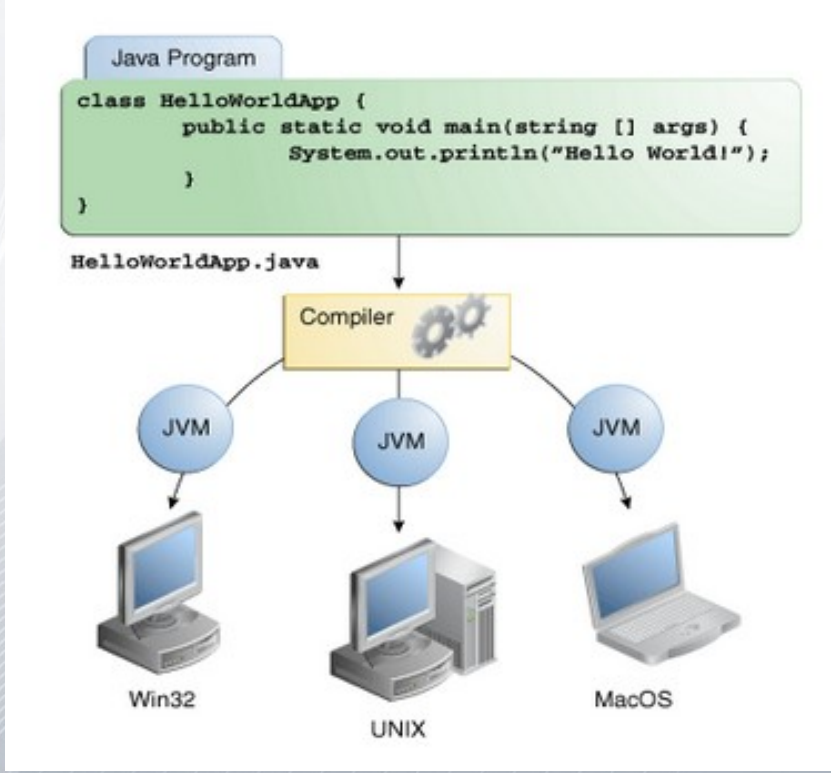
- Java source code is human readable code

  in .java files
- Compilation
- .class file does not contain code that is native to your processor. It instead contains bytecodes
- Java virtual machine



Platform-independent

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Windows, Linux, Mac OS ...



First steps in Java

- Installing JDK
- Installing Eclipse IDE (www.eclipse.org)
- My first class
  - All java classes start with capital letter
  - Classes' names do not include spaces
  - Each class is a file. File and class name are the same
  - .class and .java
  - Java is case sensitive

My first program

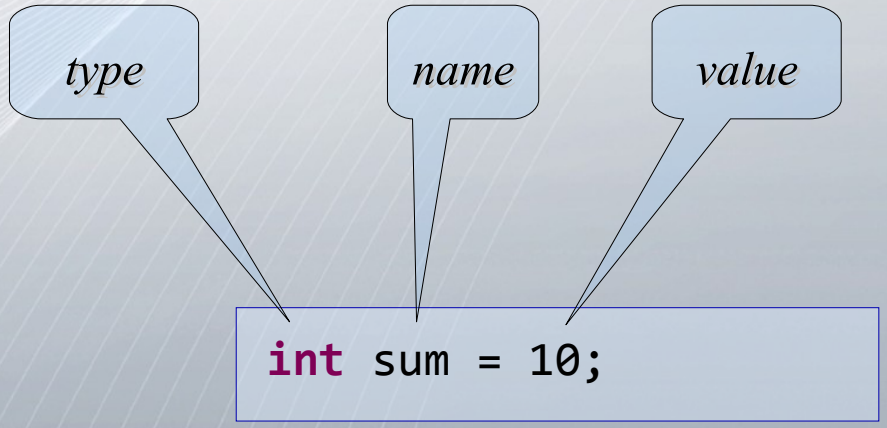
- *main* method – entry point for each java program
- System.out.println();
- HelloWorld program
- What is console?

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

Variables

- Variables in java
  - It's purpose is to hold information
  - Have an unique name
  - Have a type
  - Have a value (can be changed)

- Declaring variable



Primitive types in Java

- Primitives are basic java type
- Primitives can be used with basic operations
- Primitives' values can be assigned to variables

- Primitive types in java
  - byte, short, int, long
  - float, double
  - boolean
  - char

Numeric types

- Numeric types are **byte, short, long, int, double, float**
- **byte** – 8b (-128 : 127)

```
byte b = 100;
```

- **short** – 16b (-32768 : 32767)

```
short s = 10000;
```

- **int** – from integer, 32b

```
int i = 10000;
```

Numeric types

- **long** – 64b

  long l = 100;

  l is added as a sufix to indicate long type

- **float** - precision to 32b

  float f = 3.14f;

  f is added as a sufix to indicate float type

- **double** – precision to 64b

  double d= 3.14;

char and boolean

- **char** is used for 16b unicode character

  Char values are embedded in ''

  char ch = 'c';

- **boolean** has two values - true or false

  boolean bool = false;

Primitives' default values

Data type	Default value
byte	0
short	0
int	0
long	0
float	0.0
double	0.0
char	'\u0000'
boolean	false

Other data types

- Strings
- Reference types

We'll talk about them later in the course!

Operators

- Aritmetic +, -, *, /, %
- Logical &&, ||
- Assignment =, +=,-=, *=, /=
- Equality ==, !=
- Differences between / and %

Try using some of them and print the result in console

Reading from console

Using Scanner

```
Scanner sc = new Scanner(System.in);
```

Read user input with sc.nextXXX();

```
sc.nextInt();
sc.nextDouble();
sc.nextLong();
```

Control flow

- Control flow is the way a program goes – execution of predifined statements
- Control flow may differ each time in dependance of conditions – either input data, or predifined conditions by the programer(i.e – time and so on)
- During the program execution decisions are being met – the program flow branches

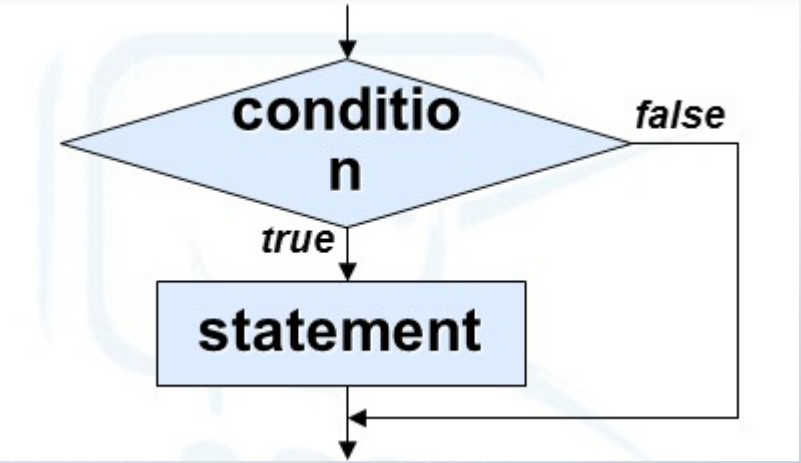
Conditional Statement

- Logical NOT !
- Logical AND &&
- Logical OR ||

A	B	A || B	A && B	! A
false	false	false	false	true
true	false	True	false	false
false	true	true	false	true
true	true	true	true	false

if-else statement

```
If (condition) {
    statement
}
```




```
if (condition) {
    executionA
} else {
    executionB
}
```

if-else statement

- If can exist without else

  But

- Else can't exist without if
- Nested if-else statement

```
double a = 7.5;

if (a < 0) {
    System.out.println("a is smaller than 0");
} else {
    if (a == 0) {
        System.out.println("a is 0");
    } else {
        System.out.println("a is bigger than 0");
    }
}
```

Blocks

A block is a group of zero or more statements between balanced braces and can be used anywhere a single statement is allowed

```
if (a > 10) {
    System.out.println("a is " + a);
    System.out.println("a is bigger than 10");
} else {
    System.out.println("a is not bigger than 10");
}
```

Always format your code! Do not write code like this:

```
if (a > 10) {
System.out.println("a is " + a);
System.out.println("a is bigger than 10");}
else {System.out.println("a is not bigger than 10");
}
```

Mistake

```
int a = 7;

if (a > 10); {
    System.out.println("a is " + a);
    System.out.println("a is bigger than 10");
}
```

In this case println statements will be executed no matter the condition!

```
int a = 7;

if (a > 10);

{
    System.out.println("a is " + a);
    System.out.println("a is bigger than 10");
}
```

Summary

- Startup
- Variables
- Primitive types
- Operators
- Working with the console
- If-else statement and blocks