



Object-Oriented Analysis & Java

Lecture 1

Florentin Bekier



Reminder: course schedule

Monday	Wednesday	Friday
10:40 – 12:10	10:40 – 12:40	9:00 – 11:00
Object-Oriented Analysis (UML)	Software Engineering Project	Object Oriented Programming (Java)
Nga Nguyen	Nga Nguyen / Florentin Bekier	Florentin Bekier
Paper exam	Project (defense, report)	Programming exam
	6 groups of 3 students	



Introduction



Object-Oriented Programming

- Programming paradigm based on the concept of *objects*
 - Fields (attributes or properties)
 - Procedures (methods)
- More intuitive, inspired by the real world
- Easier to reuse pieces of code
- Examples: Java, C++, C#, ...



Java key features

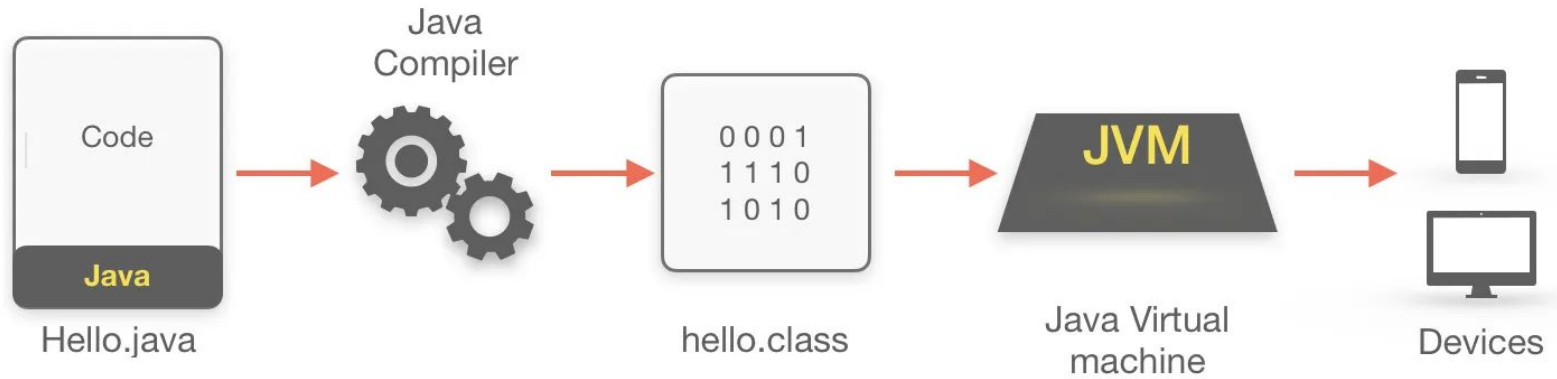
- Simple and familiar
 - Based on C++
- Object-Oriented
 - Standard libraries
- Safe
 - Strongly typed
- *Write once, run anywhere*



Java key features

- Secure
 - Sandboxed
- Distributed
 - Applets, servlets, RMI, Corba
- Multi-threaded

How Java works





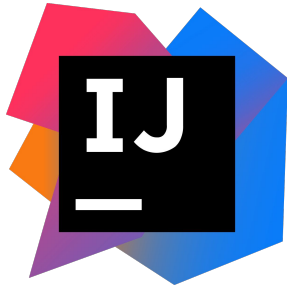
What do you need to code?

- Text editor (emacs, gEdit, VIM, ...)
- JDK (Java Development Kit)
- Automation tools: Ant, Makefile, ...

- All-in-one solution: IDE (Integrated Development Environment)
 - Eclipse, IntelliJ, Netbeans, ...



Installing IntelliJ



Student form: <https://www.jetbrains.com/shop/eform/students>

Download: <https://www.jetbrains.com/idea/>



Our first program

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



Our first program

- One class per file: same name for the class and the file
- Main method: beginning of the program execution
 - Public
 - Doesn't return any value (`void`)
 - Takes one parameter: array of `String` corresponding to the command-line attributes

Basic syntax



Comments

```
/*  
Block comment  
*/
```

```
// Inline comment
```



Javadoc

```
/**  
 * My first class  
 * @author John Doe  
 * @version 1.0  
 */  
public class HelloWorld {  
    ...  
}
```



Primitive types

- `byte`: Integer from +127 to -128 (8 bits)
- `char`: all Unicode characters (16 bits)
- `short`: Integer from +32,767 to -32,768 (16 bits)
- `int`: Integer from +2,147,483,647 to -2,147,483,648 (32 bits)
- `long`: Integer from +9,223,372,036,854,775,807 to -9,223,372,036,854,775,808 (64 bits)
- `float`: Floating-point number (32 bits)
- `double`: Floating-point number (64 bits)
- `boolean`: `true` or `false`



Other types

- Reference type:
 - Reference to an in-memory object
 - Can be `null` if it's not referencing anything
 - Example: `String`, `Integer`
- In Java, arrays are also reference types



Naming convention

- Camel-case for attributes and methods: `exampleOfVariable`
- Pascal-case for classes: `ExampleOfClass`
- Packages are in lowercase: `java.lang`



Declaring variables

```
int i; // No default value
boolean error = false;
String myText = "My name is John";
```



Operators

- Basic arithmetic : + - * / %
- Composed affectation : += -= *= /= etc.
- Increment/decrement : ++ --
- Comparison : == != > >= < <=
- Logic operators: && || !
- Concatenate strings: +



Control flow

```
if (condition) {  
    // Statement(s)  
} else if (condition) {  
    // Statement(s)  
} else {  
    // Statement(s)  
}
```

```
while (condition) {  
    // Statement(s)  
}  
  
do {  
    // Statement(s)  
} while (condition);
```



Control flow

```
switch (condition) {  
  case value1:  
    // Statement(s)  
    break;  
  case value2:  
    // Statement(s)  
    break;  
}
```

```
for (initialization;  
  termination condition;  
  increment/decrement) {  
    // Statement(s)  
}
```