Java Training
Introduction

About me

Filipp Stankevich
Software engineer

10 years in software engineering
Design Patterns and Clean Code advocate

Intro

- Contact in Discord
- Materials on Git
- Schedule (2 times a week, ongoing)
- Rules for online classes
- Questions?

https://github.com/MJC-School

https://discord.gg/VZmvgcmjkd

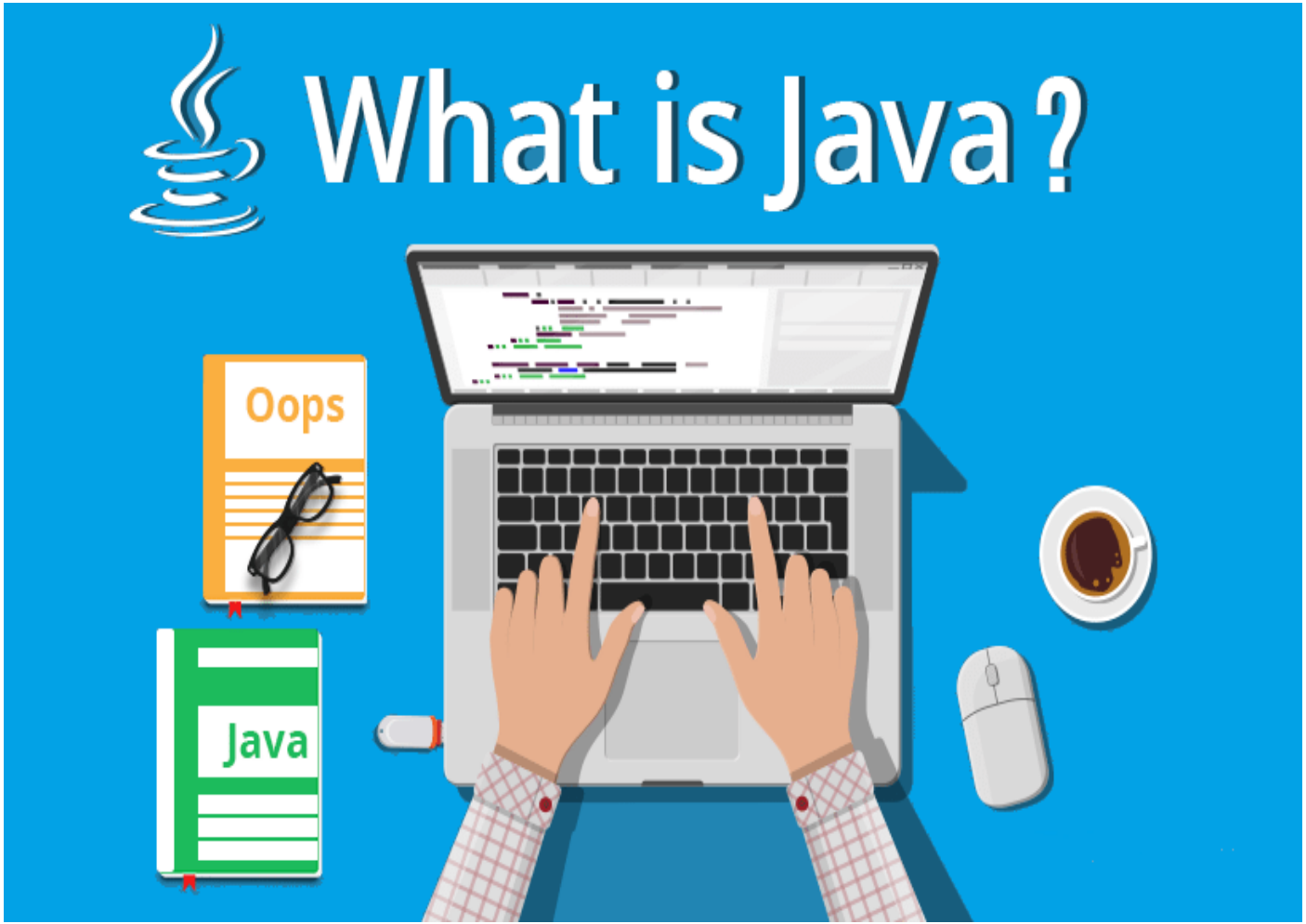
Tell me about yourself

Course agenda

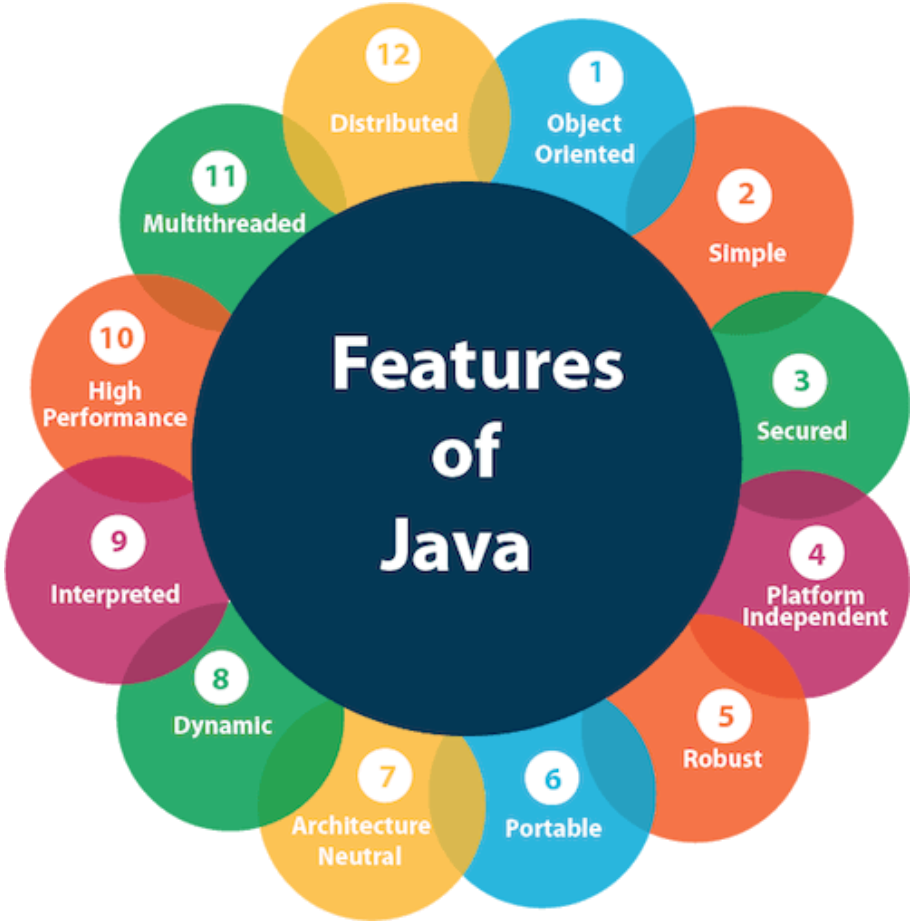
- Intro Java
- OOP & OOD
- Java Core (Strings, Collections)
- XML
- Design Patterns
- Multithreading
- RDBMS, SQL, JDBC
- HTML & CSS
- Web Application (Final project)

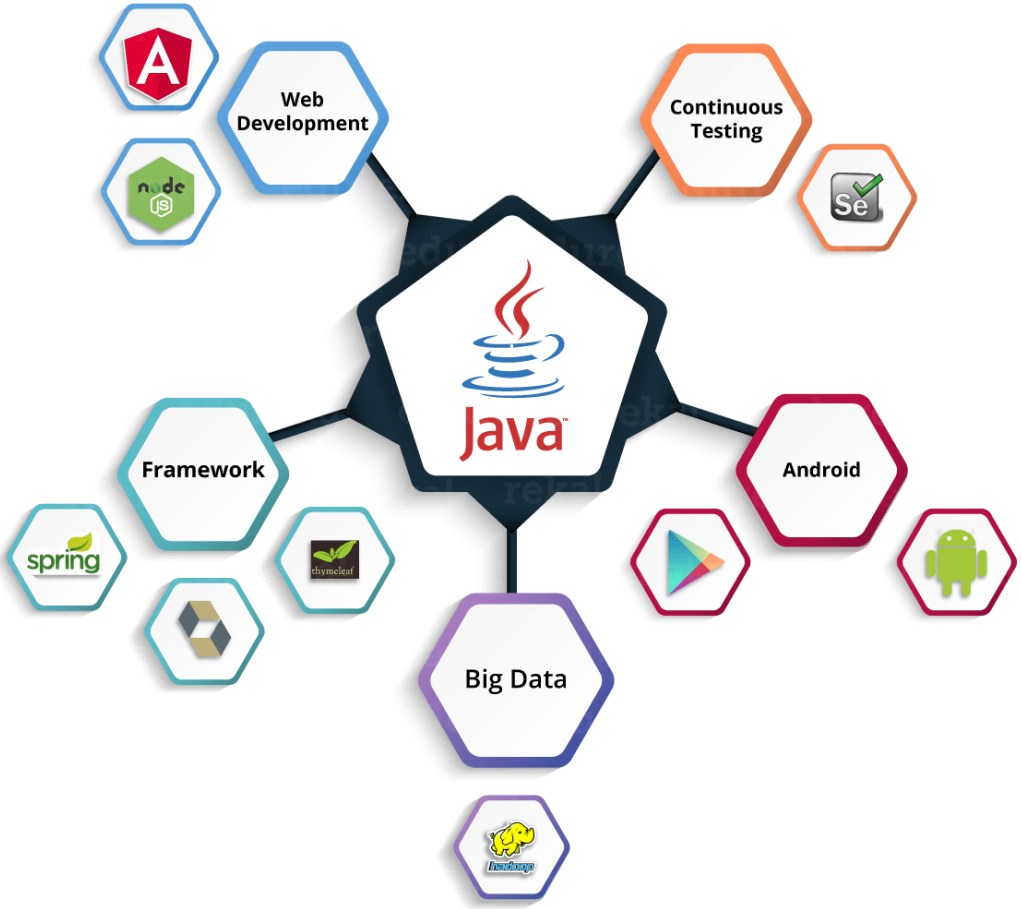
Today’s agenda

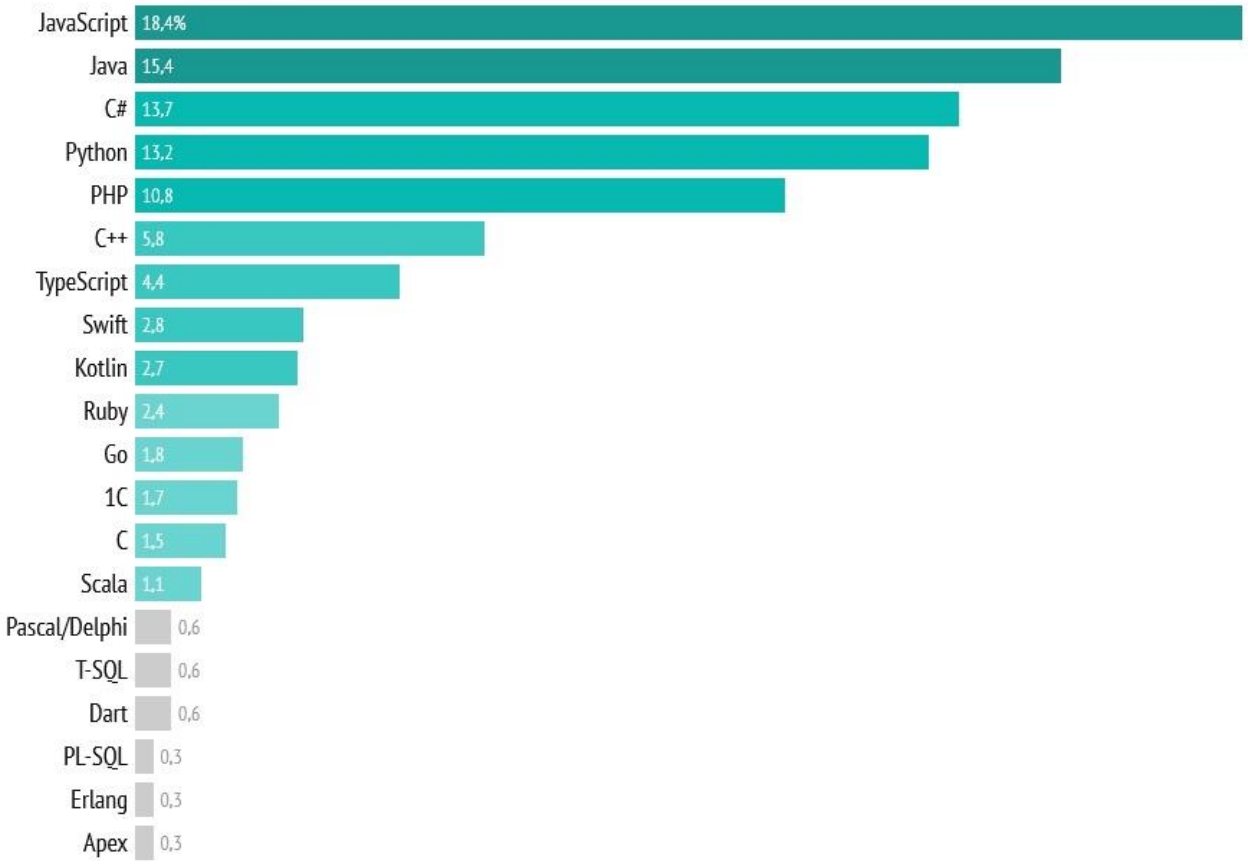
- What is Java
- What is the main goal of the developer?
- Structural programming
- Object-oriented programming
- Design principles and patterns
- Best Practices


What is Java?
Oops
Java

- Java is a Cross-platform Language.
- Java is an Object-Oriented Programming Language.
- It was developed by James Gosling and first released by Sun Microsystems in 1995.
- More than 3 billion devices run Java programs.


Features of Java
1 Object Oriented
2 Simple
3 Secured
4 Platform Independent
5 Robust
6 Portable
7 Architecture Neutral
8 Dynamic
9 Interpreted
10 High Performance
11 Multithreaded
12 Distributed


Web Development
Continuous Testing
Framework
Android
Big Data
Java
spring
thymeleaf


JavaScript 18,4%
Java 15,4
C# 13,7
Python 13,2
PHP 10,8
C++ 5,8
TypeScript 4,4
Swift 2,8
Kotlin 2,7
Ruby 2,4
Go 1,8
1C 1,7
C 1,5
Scala 1,1
Pascal/Delphi 0,6
T-SQL 0,6
Dart 0,6
PL-SQL 0,3
Erlang 0,3
Apex 0,3

What is the main goal of the professional developer?

- Can I repair my car?

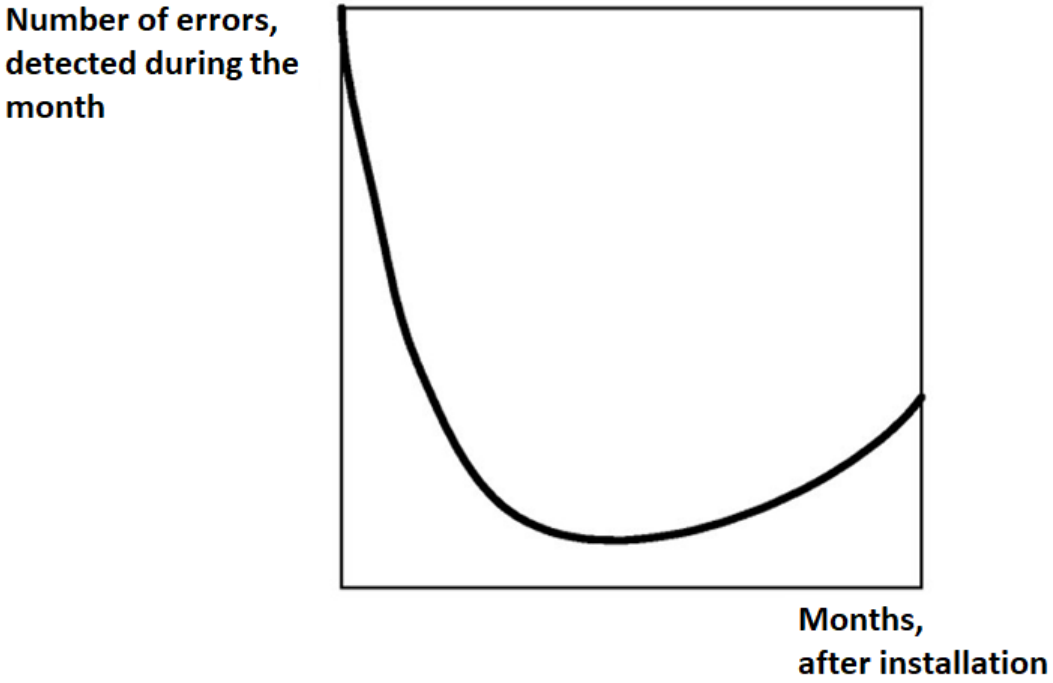
- The parts for your car are no longer produced, it's cheaper to buy a new one...

The answer is:

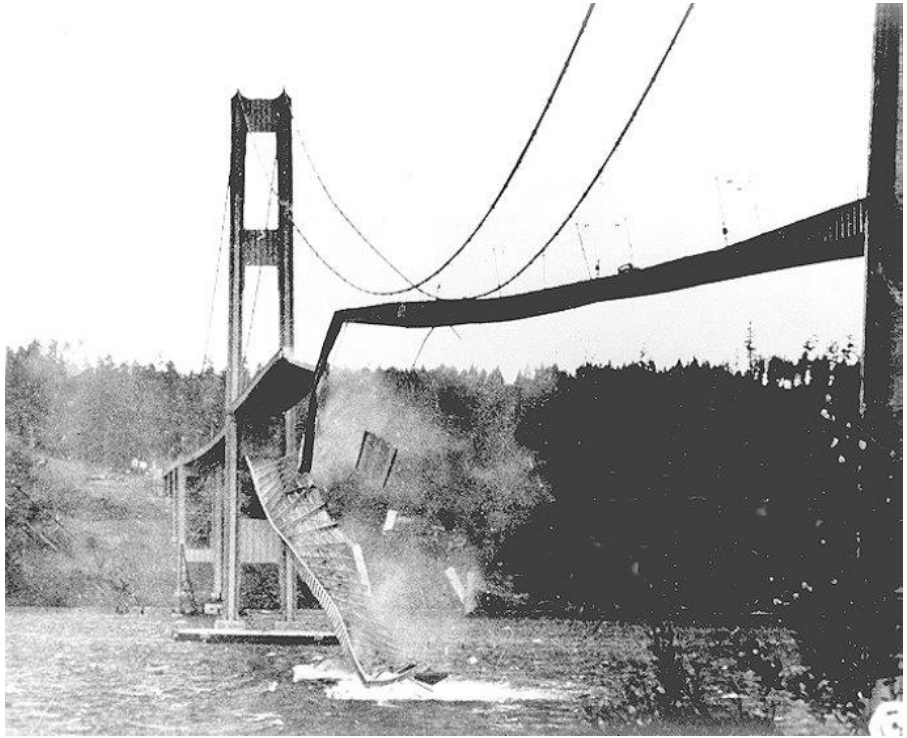
Maintainability

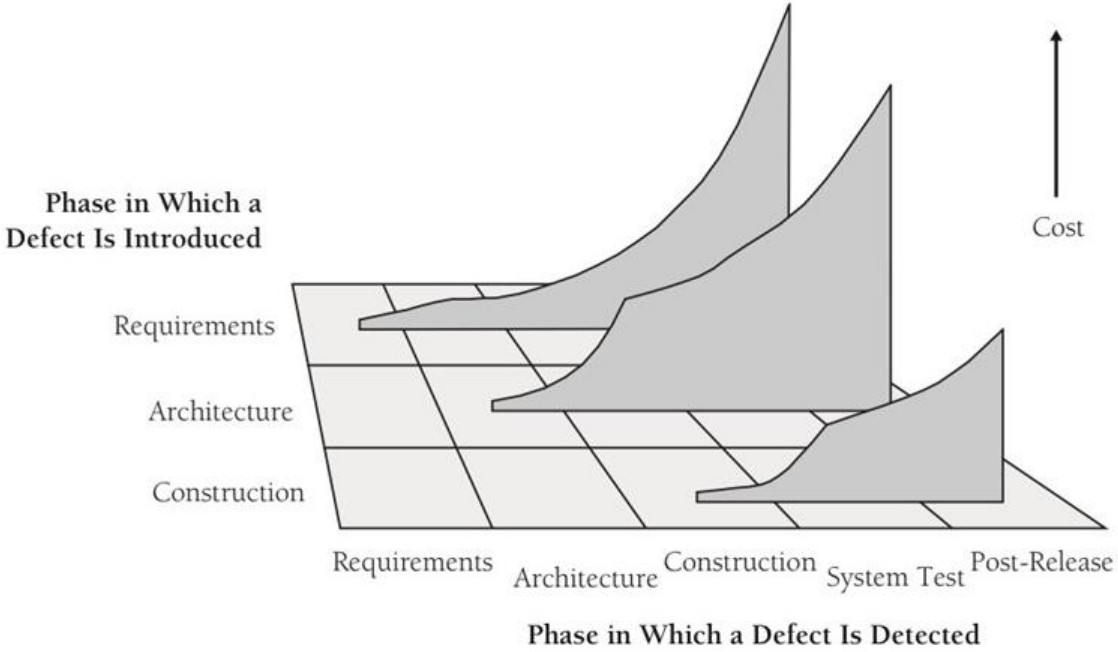
(Readability)

Error detection frequency as a function of program age



Before the destruction of the bridge, engineers did not know that aerodynamics played such a big role

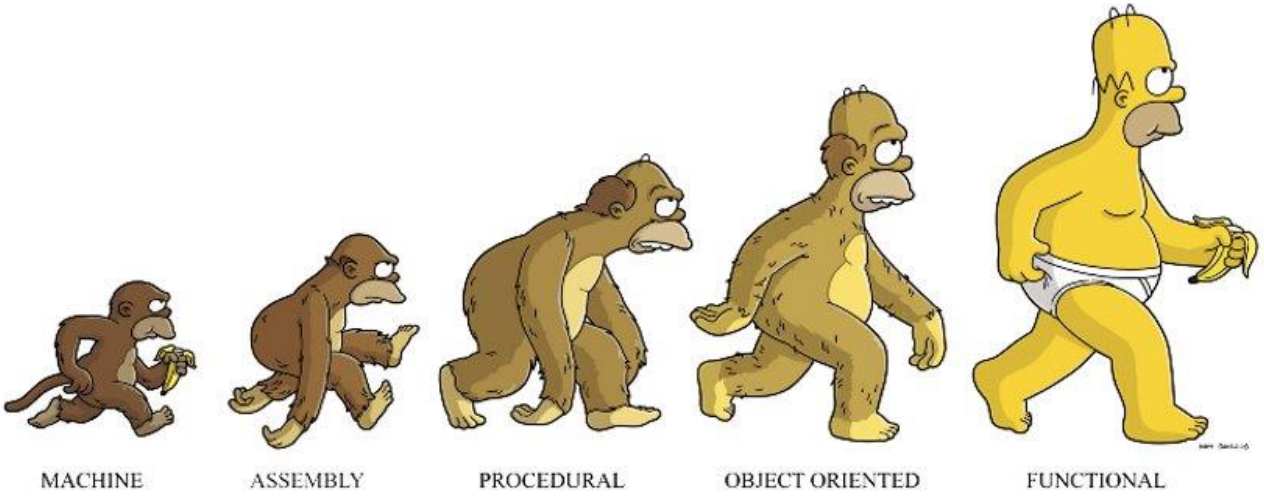


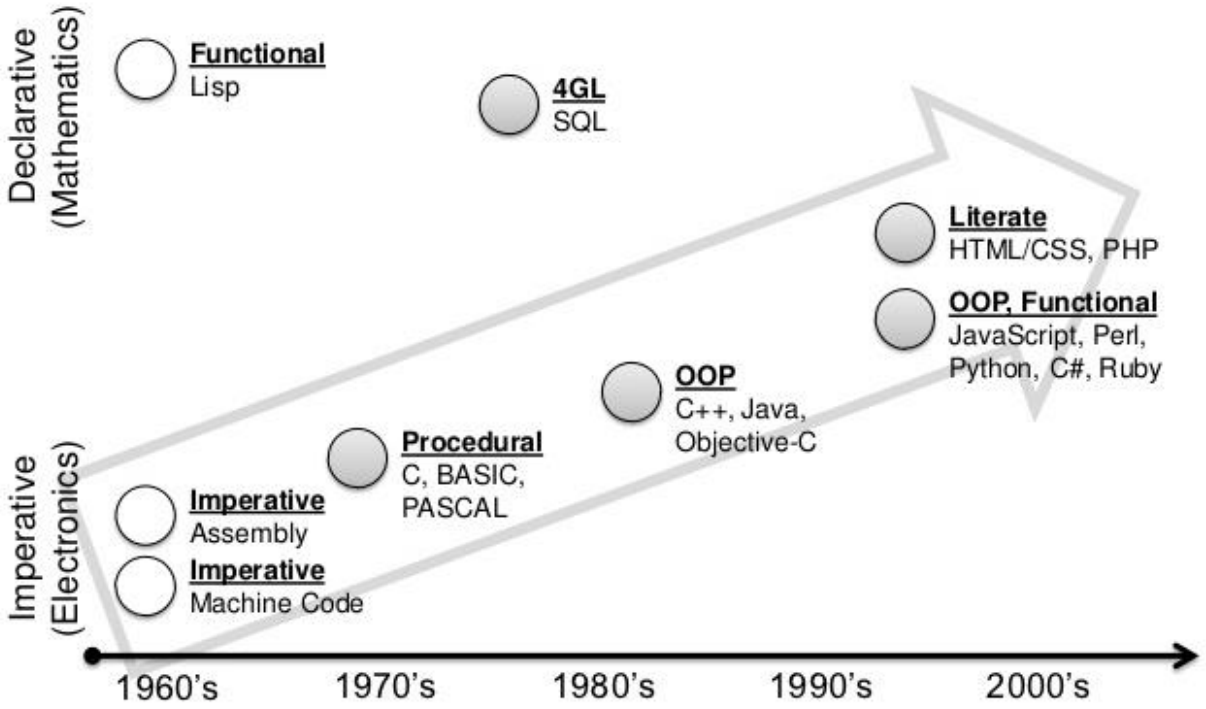


The cost to fix a defect rises dramatically as the time from when it's introduced to when it's detected increases. This remains true whether the project is highly sequential (doing 100 percent of requirements and design up front) or highly iterative (doing 5 percent of requirements and design up front).

Programming paradigms evolution

Programming paradigms evolution

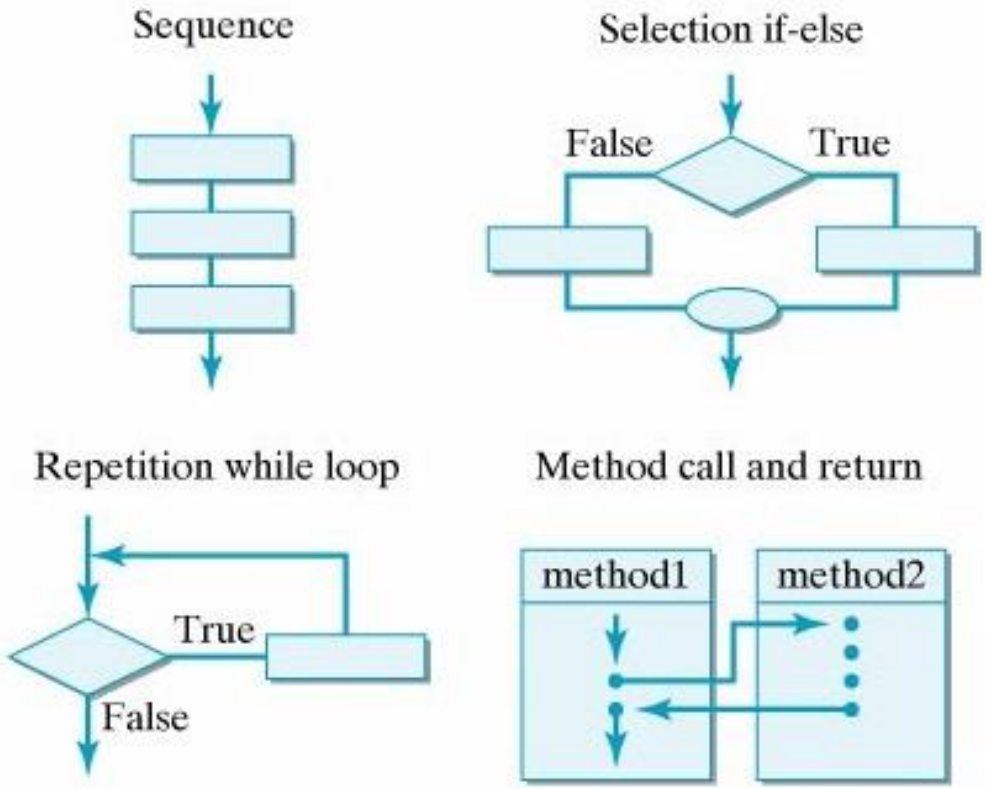



Declarative
(Mathematics)
Functional
Lisp
4GL
SQL
Literate
HTML/CSS, PHP
OOP, Functional
JavaScript, Perl,
Python, C#, Ruby
OOP
C++, Java,
Objective-C
Procedural
C, BASIC,
PASCAL
Imperative
(Electronics)
Imperative
Assembly
Imperative
Machine Code
1960's
1970's
1980's
1990's
2000's

Structural Programming

Structural Programming Principles

- Program design using top-down or bottom-up approach
- Decomposition of program into components
- Structuring of flow control


Sequence
Selection if-else
False
True
Repetition while loop
True
False
Method call and return
method1
method2

Top-down Approach

Factorial calculation

```
class Main {

    public static void main(String[] args) {
        // read argument
        // do calculation
        // show result
}
```

```
class Main {

    public static void main(String[] args) {
        int number = readArgument();
        int result = calculateFactorial(number);
        printResult(result);
    }
}
```

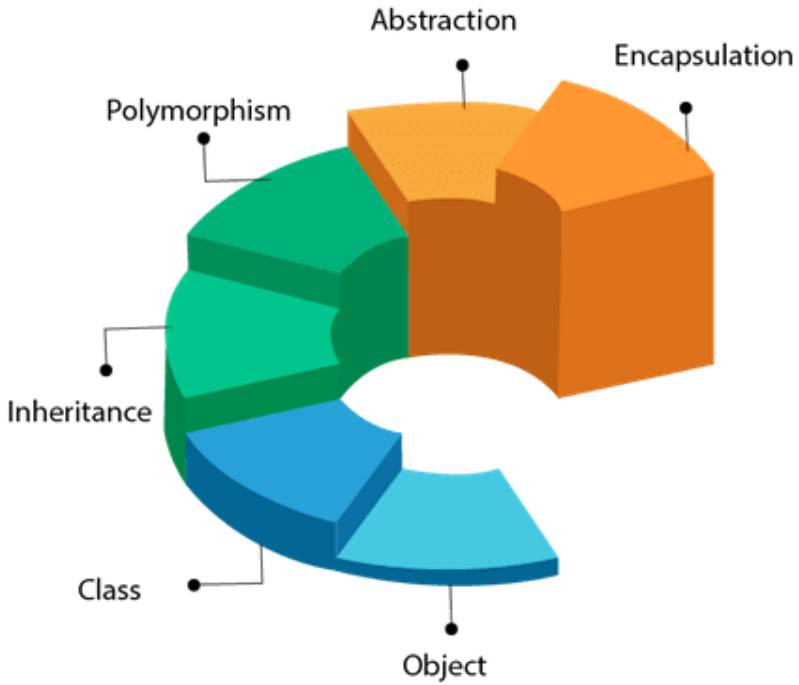
```
int readArgument() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter number: " );
    String line = scanner.nextLine();
    return Integer.parseInt(line);
}
```

```
int calculateFactorial(int number) {
    int reuslt = 1;
    for (int i = 1; I <= number; i++) {
        reuslt = reuslt * i;
    }
    return reuslt;
}
```

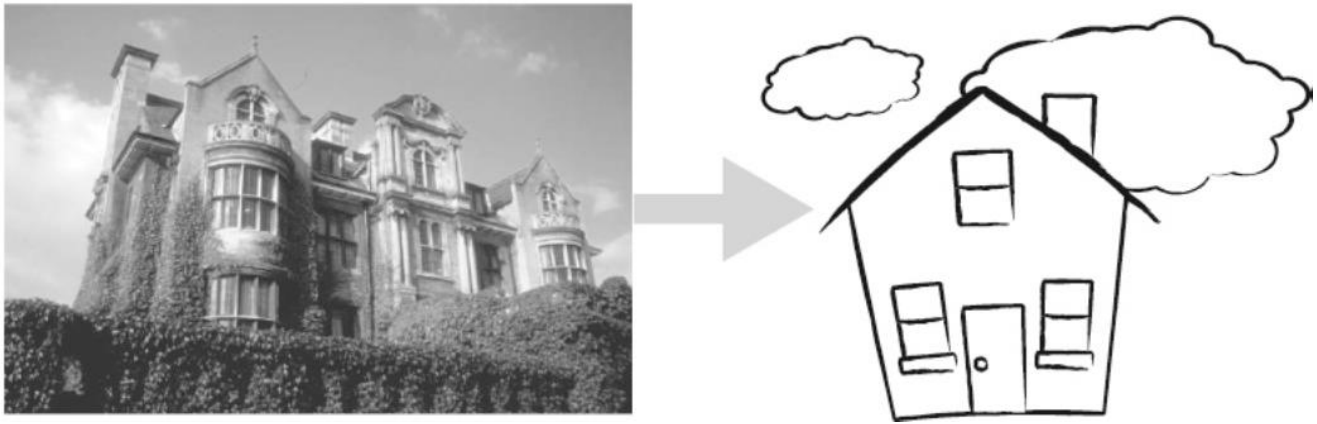
```
void printResult(int number) {
  System.out.print("Factorial : " + number)
}
```

Object Oriented Programming

Object-oriented programming

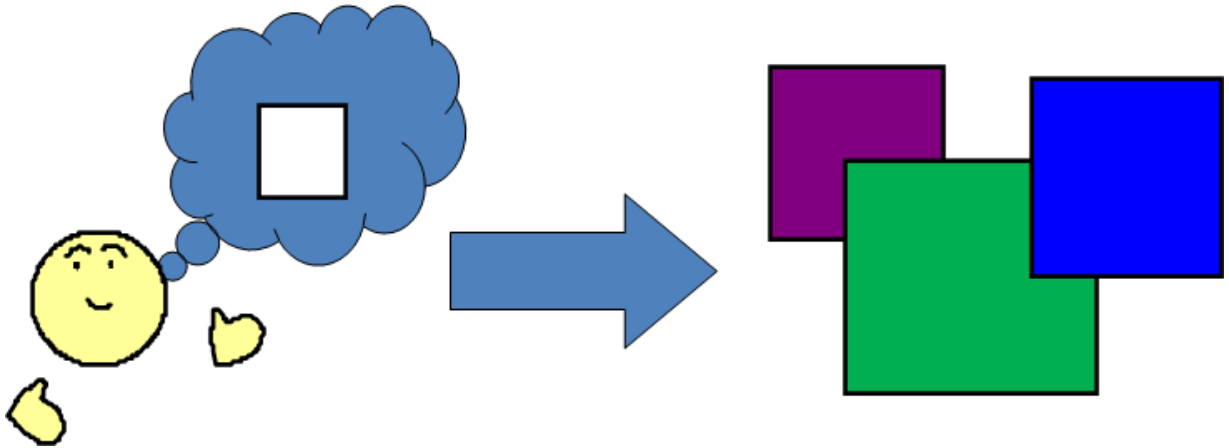


Abstraction

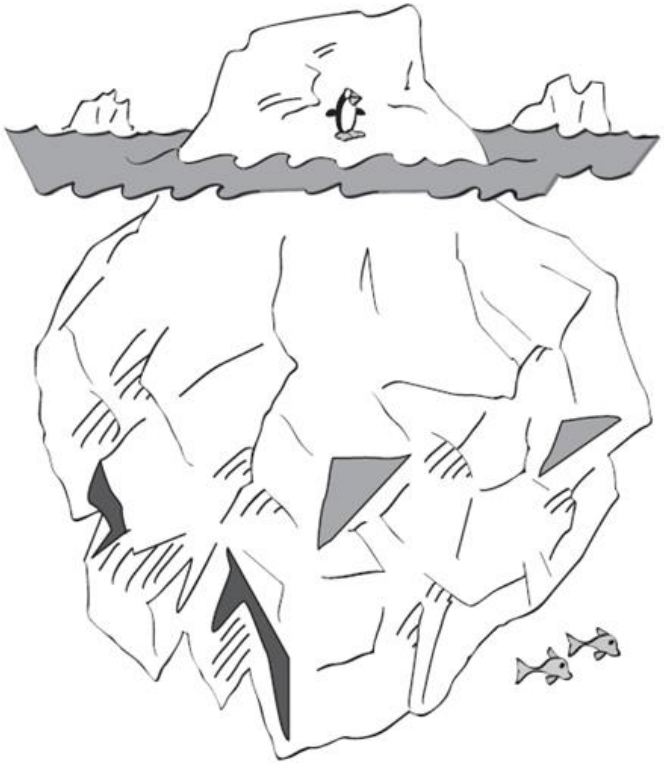


Class is an object template (or an abstraction)

Object is an instance of a class.



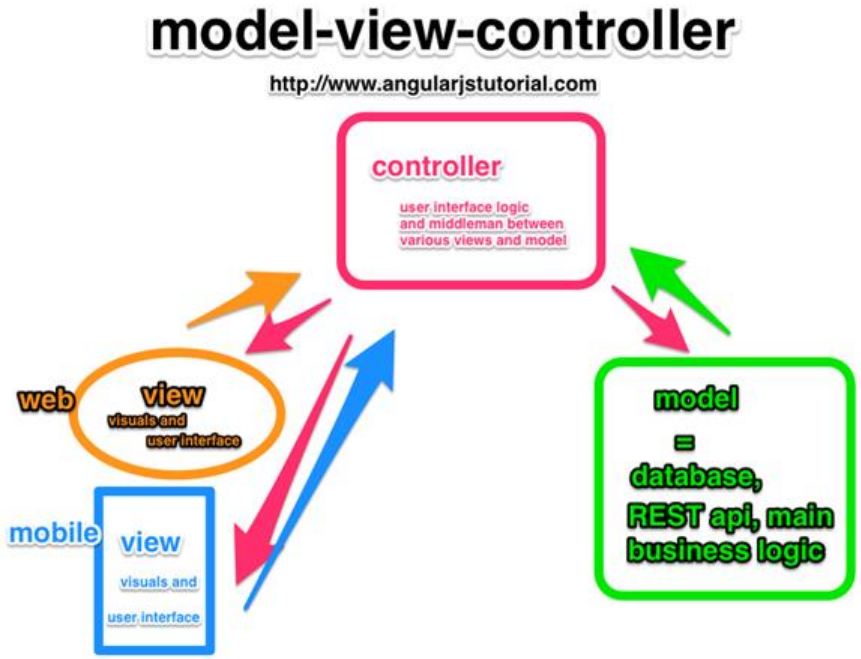
Encapsulation

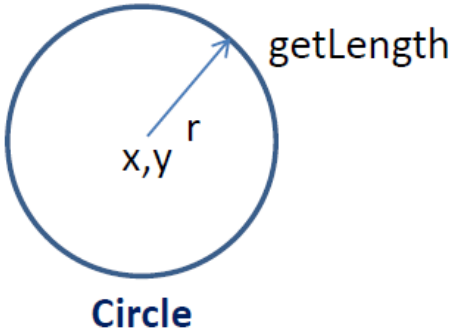


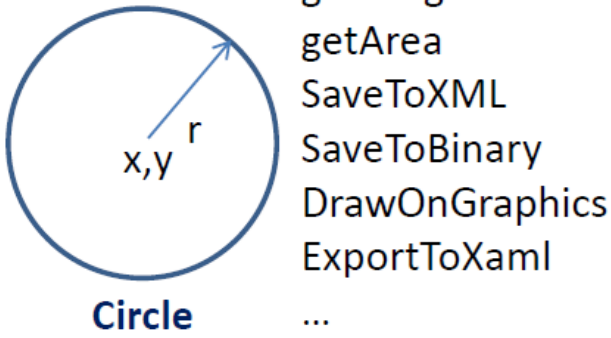
Design Patterns/Principles

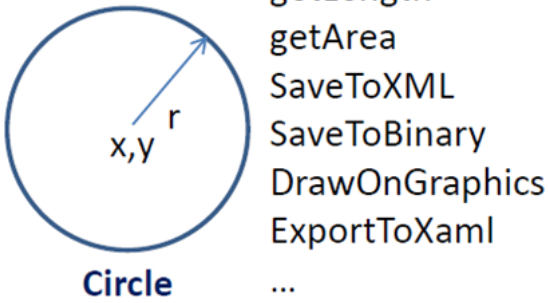
Design Principles

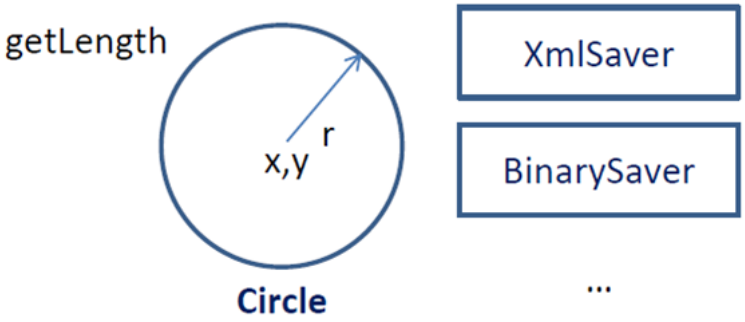
- Separation of Concerns
- Low coupling
- High cohesion


model-view-controller
http://www.angularjstutorial.com
controller
user interface logic
and middleman between
various views and model
web
view
visuals and
user interface
mobile
view
visuals and
user interface
model
=
database,
REST api, main
business logic


getLength
x,y
r
Circle


x,y
r
Circle
getLength
getArea
SaveToXML
SaveToBinary
DrawOnGraphics
ExportToXaml
...


x,y
r
Circle
getLength
getArea
SaveToXML
SaveToBinary
DrawOnGraphics
ExportToXaml
...


getLength
x,y
r
Circle
XmlSaver
BinarySaver
...

Best Practices

- Test Driven Development
- Continuous Integration
- Code review

Recommended materials

- Code Complete, McConnell
- Effective Java, Bloch
- Building Maintainable Software (java edition), Visser
- Thinking in Java, Eckel

Next class topics

- Git
- Maven

Homework

- Install IDE (e.g. IntelliJ IDEA Community Edition)
- Install JDK

Thanks