

An Introduction of Java

Duy Vu <vud1@mcmaster.ca>

Aliakbar Safilian <a.a.safilian@gmail.com>

Rokan Faruqui <rufaruqui@gmail.com>

Sept. 18, 2015

Table of Contents

Java

OOP

Concurrency in Java

Processes vs. Threads

Synchronization

Solve it yourself

Quiz

Java



[Credited to: martinjeeblog.wordpress.com/2012/10/24/can-oracle-keep-the-java-train-on-the-tracks/]

Java Programming Language?

Java Programming Language?

- ▶ Free and Open-source

Java Programming Language?

- ▶ Free and Open-source
 - ▶ Java SDKs can be download from
<http://www.oracle.com/technetwork/java/index.html>

Java Programming Language?

- ▶ Free and Open-source
 - ▶ Java SDKs can be download from <http://www.oracle.com/technetwork/java/index.html>
 - ▶ Source code (src.zip) can be viewed at
 - ▶ MacOS: /Library/Java/JavaVirtualMachines/<version>/Contents/Home
 - ▶ Window: C:\Program Files (x86)\Java\<version>
 - ▶ Linux: ???

Java Programming Language?

Java Programming Language?

- ▶ Object-Oriented Programming (OOP) model
 - ▶ 3 principles:

Java Programming Language?

- ▶ Object-Oriented Programming (OOP) model
 - ▶ 3 principles:
 - ▶ Encapsulation
 - ▶ Inheritance
 - ▶ Polymorphism

Table of Contents

Java

OOP

Concurrency in Java

Processes vs. Threads

Synchronization

Solve it yourself

Quiz

OOP

Encapsulation

OOP

Encapsulation

- ▶ Packing data and functions into a single component.
- ▶ Java's encapsulation key words:
 - ▶ `class`
 - ▶ `private`, `protected`, and `public`

OOP

Encapsulation

- ▶ Packing data and functions into a single component.
- ▶ Java's encapsulation key words:
 - ▶ `class`
 - ▶ `private`, `protected`, and `public`

Inheritance

OOP

Encapsulation

- ▶ Packing data and functions into a single component.
- ▶ Java's encapsulation key words:
 - ▶ `class`
 - ▶ `private`, `protected`, and `public`

Inheritance

- ▶ Derived class can reuse some code from its base class(es).
- ▶ Only `protected` and `public` attributes and methods are inheritable.

OOP

Encapsulation

- ▶ Packing data and functions into a single component.
- ▶ Java's encapsulation key words:
 - ▶ `class`
 - ▶ `private`, `protected`, and `public`

Inheritance

- ▶ Derived class can reuse some code from its base class(es).
- ▶ Only `protected` and `public` attributes and methods are inheritable.

Polymorphism

OOP

Encapsulation

- ▶ Packing data and functions into a single component.
- ▶ Java's encapsulation key words:
 - ▶ `class`
 - ▶ `private`, `protected`, and `public`

Inheritance

- ▶ Derived class can reuse some code from its base class(es).
- ▶ Only `protected` and `public` attributes and methods are inheritable.

Polymorphism

- ▶ Multiple functions have the same name but number and/or type of parameters are different.
- ▶ Inherited functions can be completely redefined.
- ▶ **Overloading vs. Overriding**

Overloading vs. Overriding

Same:

- Functions have the same name.

Differences:

Overloading	Overriding
Within a single class	In child class
The number or types of parameters are different.	The number and type of parameters are same
Resolved at compile time	Resolve at runtime

Table of Contents

Java

OOP

Concurrency in Java

Processes vs. Threads

Synchronization

Solve it yourself

Quiz

Concurrency in Java

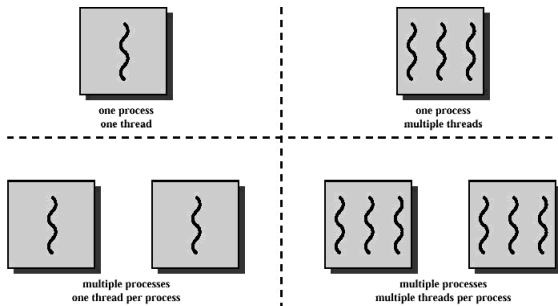
Examples:

- ▶ Simultaneously: coding Java on Eclipse, listening music on Youtube, and downloading file from email.
- ▶ Listening music on Youtube: simultaneously reading data off network, decompress it, decode it and play it.

Processes vs. Threads

Lightweight processes which share memory

- ▶ no memory protection between concurrent threads
- ▶ communication via shared variables



[Credited to: <http://www.cs.cf.ac.uk/Dave/C/node29.html>]

Java thread - key methods

- ▶ `start()`
 - ▶ causes thread to run independently
 - ▶ invokes programmer defined `run()` method
- ▶ `stop()`: explicitly terminates thread
- ▶ `yield()`: explicitly gives up processor
- ▶ `sleep()`: suspends for a fixed amount of time and automatically makes it eligible for execution

Ref:

<http://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html>

Synchronization

```
1
2 public class Counter {
3     private int c = 0;
4
5     public void increment() {
6         c++;
7     }
8
9     public void decrement() {
10        c--;
11    }
12
13    public int value() {
14        return c;
15    }
16 }
```

Synchronization

There are 2 simultaneous threads (Thread A and Thread B):

- ▶ Thread A invokes `increment()`
- ▶ Thread B invokes `decrement()`

Synchronization

There are 2 simultaneous threads (Thread A and Thread B):

- ▶ Thread A invokes `increment()`
- ▶ Thread B invokes `decrement()`

The application could do:

1. Thread A: Retrieve c.
Thread B: Retrieve c.
2. Thread A: Increment retrieved value; result is 1.
Thread B: Decrement retrieved value; result is -1.
3. Thread A: Store result in c; c is now 1.
Thread B: Store result in c; c is now -1.

Synchronization

There are 2 simultaneous threads (Thread A and Thread B):

- ▶ Thread A invokes `increment()`
- ▶ Thread B invokes `decrement()`

What we really want:

1. Thread A: Retrieve c.
Thread A: Increment retrieved value; result is 1
Thread A: Store result in c; c is now 1
2. Thread B: Retrieve c.
Thread B: Decrement retrieved value; result is 0.
Thread B: Store result in c; c is now 0.

Table of Contents

Java

OOP

Concurrency in Java

Processes vs. Threads

Synchronization

Solve it yourself

Quiz

Solve it yourself

- ▶ Class VS. Interface
- ▶ HOW TO TRANSLATE A MODEL TO JAVA IMPLEMENTATION?
→ Implement traffic light

Table of Contents

Java

OOP

Concurrency in Java

Processes vs. Threads

Synchronization

Solve it yourself

Quiz

Quiz

Answer the below question in a single word file (pdf, doc, docx, rtf)

1. What are principles of OOP model?
2. What are the differences of overloading and overriding?
3. Screenshot of the result of your HelloThread and HelloRunnable classes.
4. Screenshot of your SynchronizedCounter class.