

Concurrent Object Oriented Languages

Concurrent Programming in Java

`wiki.eecs.yorku.ca/course/6490A`

- Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes and Doug Lea. Java Concurrency in Practice. Addison-Wesley, 2006.
- Mary Campione, Kathy Walrath and Alison Huml. The Java Tutorial. Lesson: Threads: Doing Two or More Tasks At Once.
- James Gosling, Bill Joy, Guy L. Steele Jr. and Gilad Bracha. The Java Language Specification. Third edition.

Thread Creation

In Java, threads are created dynamically:

```
// create and initialize a Thread object
Thread thread = new Thread();
// start the thread
thread.start();
```

The class `Thread` is part of package `java.lang` (and, hence, does not need to be imported).

Question

Develop a Java class called `Printer` that is a `Thread` and prints its name 1000 times.

Two Concurrent Printers

Question

Develop an app that creates two `Printers` with names 1 and 2 and run them concurrently.

Java Only Supports Single Inheritance

The following is **not** allowed in Java.

```
public class Printer extends Applet, Thread
```

Thread Creation

```
// create and initialize a Runnable object
Runnable runnable = new ...();
// create and initialize a Thread object
Thread thread = new Thread(runnable);
// start the thread
thread.start();
```

The interface `Runnable` is part of package `java.lang` (and, hence, does not need to be imported).

Runnable is an Interface

In Java, you cannot create instances of an interface.

```
public class Printer implements Runnable
{
    ...
}
```

The assignment

```
Runnable printer = new Printer();
```

is valid since the class `Printer` implements the interface `Runnable`.

Question

Develop a Java class called `Printer` that implements `Runnable` and prints the thread's name 1000 times.

Semaphores

The Java class library contains the class
`java.util.concurrent.Semaphore`.

The method `acquire` represents the P-operation and the
method `release` represents the V-operation.

The Readers-Writers Problem

```
int readers = 0;  
semaphore mutex = 1;  
semaphore token = 1;
```

Reader

```
P(mutex);
readers++;
if (readers == 1)
    P(token);
V(mutex);
read
P(mutex);
readers--;
if (readers == 0)
    V(token);
V(mutex);
```

```
P(token );  
write  
V(token );
```