Week 3 Lab Session

CS2030S AY21/22 Semester 2

Lab 14B

Yan Xiaozhi (David)
@david_eom
yan_xiaozhi@u.nus.edu

27 Jan 2022

NUS School of Computing
13 COMPUTING DRIVE

Who is this dude?

- Yan Xiaozhi (David)
- Year 2 Computer Science
- Ex-avenger for CS1101S
- Tembusu College, tShuttlers
- Guitar, piano, badminton
- Fun fact: I had 3 nationalities at one point
- Summer internship at IMDA, working on TTS with Singaporean accent

What about you guys!

- Name (any preferred name?)
- Any prior programming experience with Java?
- What do you hope to gain out of CS2030S?

What is CS2030S?

- Gateway to future modules
  - CS1101S: foundation building
  - CS2030S: solving problems elegantly
  - CS2040S: solving problems correctly and efficiently
- Medium workload, definitely less than CS1101S
- NOT a module to learn Java as a programming language
- Other miscellaneous but useful skills e.g. unix commands, vim, ssh

Importance of CS2030S

- Useful in future modules (CS2103T especially)
- Indispensable skill in the industry
  - Well-designed code
  - Good programming habits
  - Online assessment / coding interview
- OOP paradigm & Java

Got questions?

- Our Lab Telegram chat
- Piazza forum
- Other CS friends
- PM me (only as a last resort ((:)!
  - I’ll reply in the Telegram chat as well!
  - Might be thought-provoking and important for all!
  - No such thing as a stupid question

Admin

- Thurs, 2pm-4pm, release ~3.45pm (remind me if i run overtime)
- Free and easy, you do not need to stay until 4pm
- No attendance marks, but good opportunity to clear any doubts
- General flow:
    - Recap of content / previous week's lab
    - Briefing for current week's lab
    - Coding & QnA
    - Anonymous feedback: bit.ly/lab14bfeedback

Admin

- Labs consists of 60% of overall grade in CS2030S
  - Weekly lab assignments: 30%
  - PE1: 10% (Week 7 Saturday)
  - PE2: 20% (Week 12 Saturday)
- Before PE1/PE2: possibly a mock PE

Admin

- Please do your weekly FET before coming to class
- Contact tracing
- Any lesson before / after?
- Plagiarism

5 cents from me

- Find friends to discuss and learn with
- Ask questions, Piazza can be quite useful!
- Learn Vim well, customise and get used to your own `.vimrc`
- Do not overlook practical assessment! (super tight timeframe)
    - Try to time yourself and finish lab within 2-3 hours

Setup

SSH (Secure Shell)

- Why do we bother using this?
- Why must we use Vim? (ew)
- What should I use to SSH?
  - Mac: Terminal
  - Windows: Windows Subsystem for Linux (WSL), Ubuntu
  - Linux: you shouldn’t be asking this question LOL

CS2030S Programming Methodology II
Home
Notes
Guide
Programming Environments
GitHub Setup
Using Unix CLI
Using Vim
Java Style Guide
JavaDoc Guide
Lab Guide

The CS2030S Programming Environment

Java version

Java is a language that continues to evolve. A new version is released every six months. For CS2030S, we will *only* use Java 11, the second most recent version with long-term support[1]. Specifically, we use `openlogic-openjdk-11.0.8+10` on Ubuntu 20.04.3 LTS.

Programming Servers

The school has provided a list of computing servers for you to use, with all the required software for CS2030S installed. You can access them remotely via `ssh`, or secure shell. The hosts are named `pe111`, `pe112`, ... , `pe120`. (`pe` stands for "programming environment"). We will refer to these servers generally as the *PE hosts*.

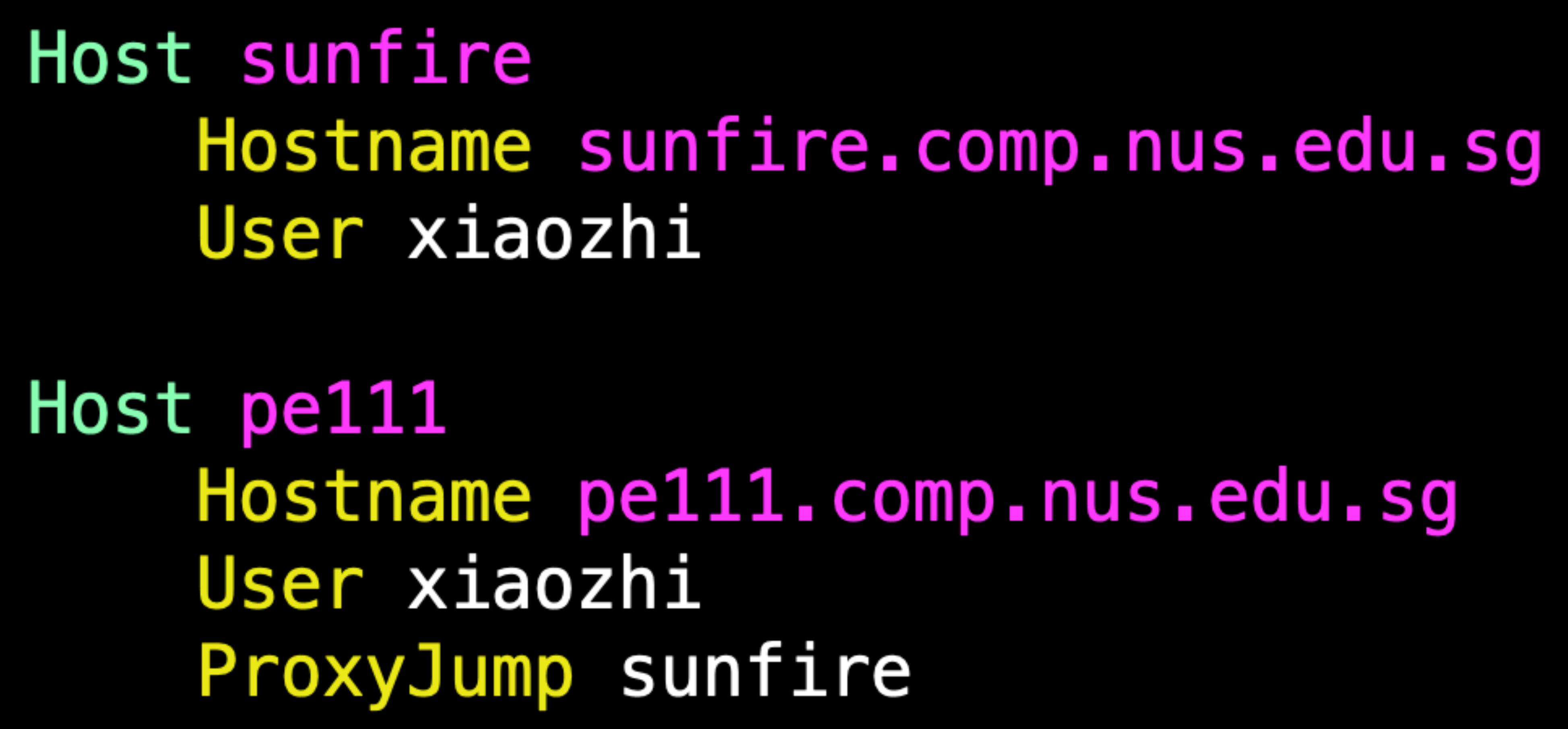
For this semester, the two servers `pe115` and `pe116` are not available.

SSH Shortcut

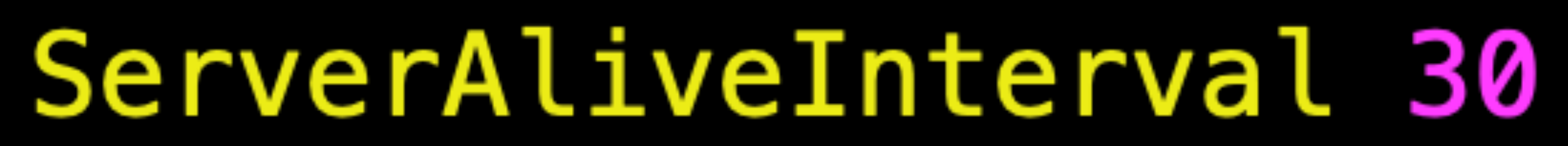
- Saves tons of time
- ~/.ssh/config
- Key in the following in the config file:

  -
    ```
    Host sunfire
         Hostname sunfire.comp.nus.edu.sg
         User <your soc account username>
    ```

  -
    ```
    Host pe11x
         Hostname sunfire.comp.nus.edu.sg
         User <your soc account username>
         ProxyJump sunfire
    ```
- `ssh pe11x`/`ssh sunfire` in terminal should prompt you for password



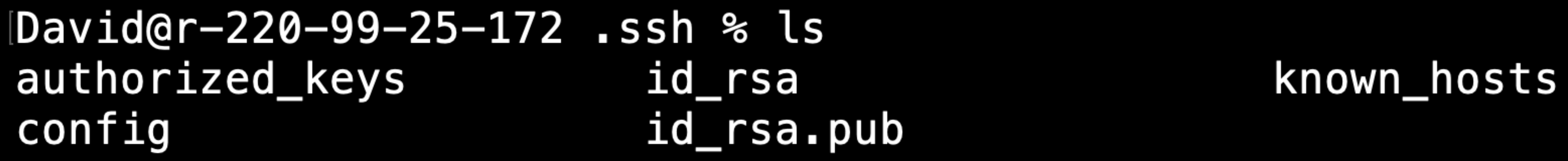
```
Host sunfire
     Hostname sunfire.comp.nus.edu.sg
     User xiaozhi

Host pe111
     Hostname pe111.comp.nus.edu.sg
     User xiaozhi
     ProxyJump sunfire
```



```
ServerAliveInterval 30
```

SSH Password-less Login

- `cd` to `~/.ssh`
- `ssh-keygen`



```
David@r-220-99-25-172 .ssh % ls
authorized_keys         id_rsa                 known_hosts
config                  id_rsa.pub
```

- Follow instructions on the terminal, you can choose to skip passphrase
  - Need to type in passphrase every time if you do specify
- There should be two new files in the directory: `id_rsa` & `id_rsa.pub`
  - `id_rsa`: private key, please never leak!
  - `id_rsa.pub`: public key, to be used for all sorts of SSH needs e.g. GitHub
- `scp id_rsa.pub sunfire:~/.ssh/authorized_keys`
- `scp id_rsa.pub pe11x:~/.ssh/authorized_keys`

Lab 0 Recap

Important Concepts

- Information hiding
  - `Point` having two `private` fields, `x` and `y`
- Tell, don’t ask
  - There should be no getter/setter methods for `Point`
  - Method to compute distance between two points
  - Getting the `Point` to print itself in `Circle`
- Inheritance
  - `RandomPoint` IS-A `Point`
- RNG being a class field, `setSeed` should be a class method
  - All random points share the same RNG

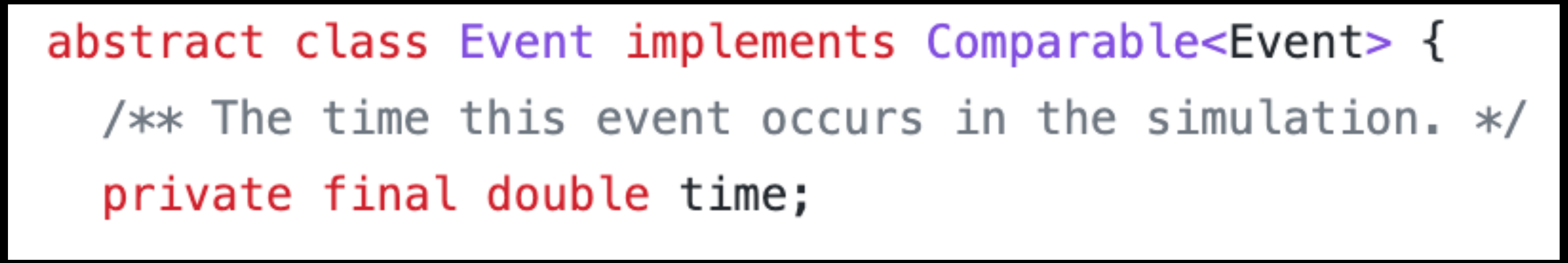
Lab 1 Overview

Lab 1: Discrete Event Simulator (Part 1)

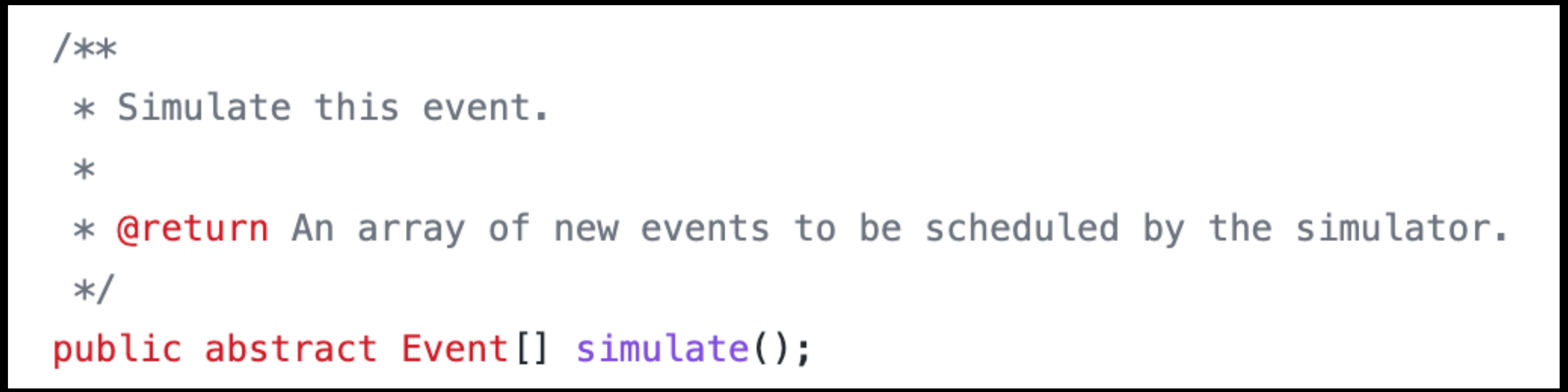
- Used to be a semester-long project, toned-down version
- You will be working on the same problem for the next 3 weeks
  - Plan well!
- Provided with poorly designed implementation
- Apply OOP principles, familiarise with PE
- Two main classes to edit:
  - `ShopEvent`: represents the various events that can happen in the simulation
  - `ShopSimulation`: main driver for simulating the events

Lab 1: Discrete Event Simulator (Part 1)

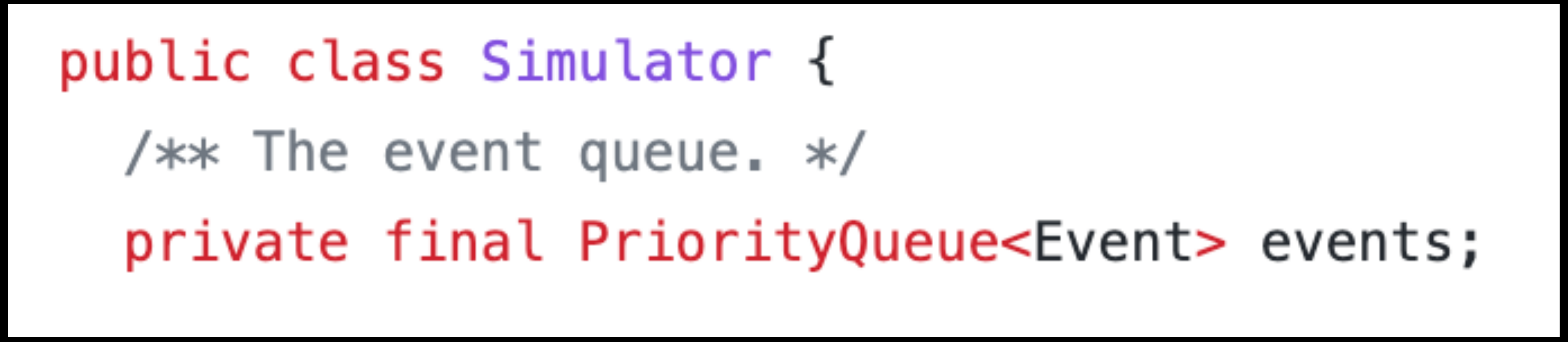
- `Event`: abstract class
  - We won't know how to simulate the event without knowing what type of event it is, thus `simulate` method is also abstract
- `Simulation`: abstract class
  - We won't know what to simulate
- WTH is Comparable<Event> & PriorityQueue<Event>?
  - Will be touched on later in the mod



```
abstract class Event implements Comparable<Event> {
  /** The time this event occurs in the simulation. */
  private final double time;
```



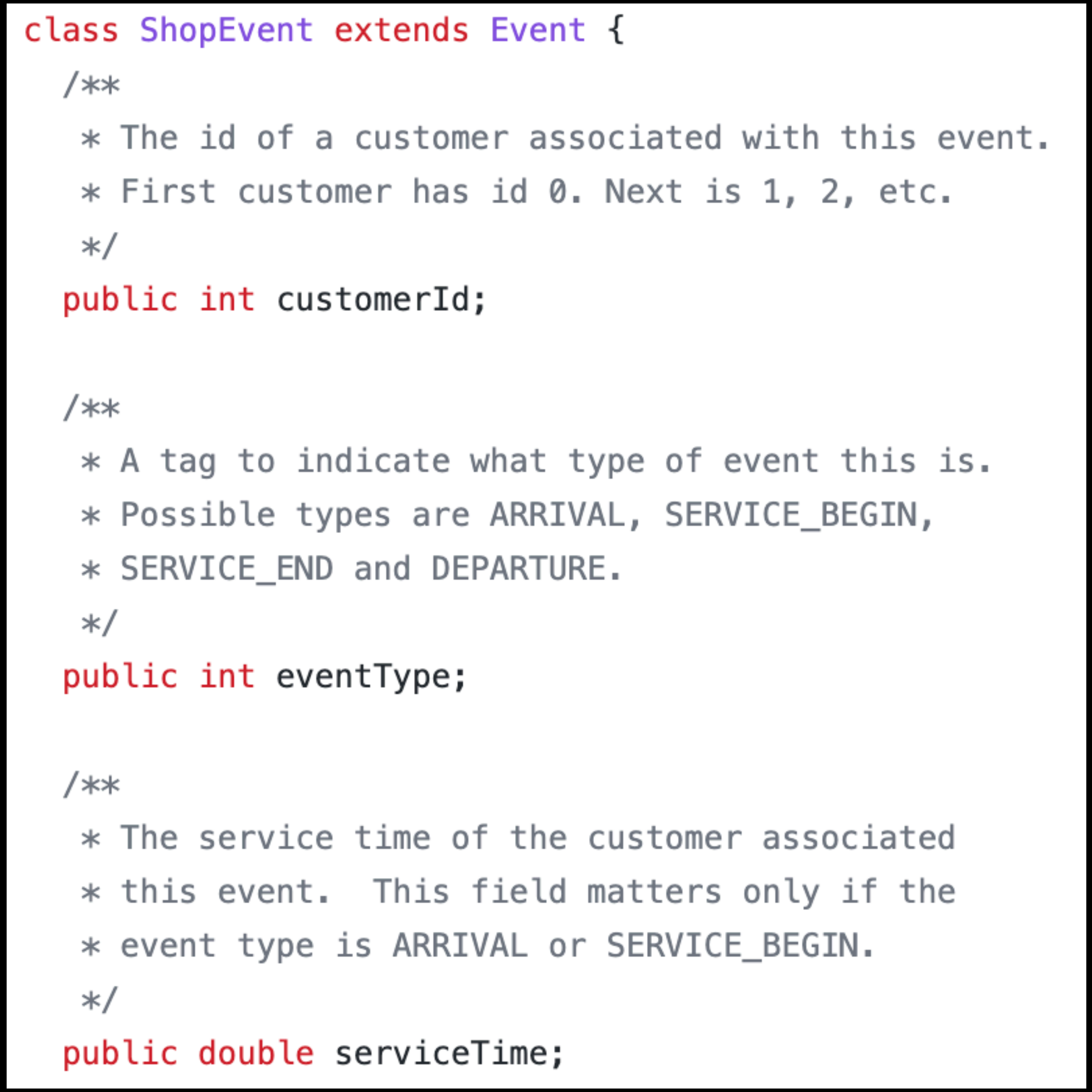
```
/**
 * Simulate this event.
 *
 * @return An array of new events to be scheduled by the simulator.
 */
public abstract Event[] simulate();
```



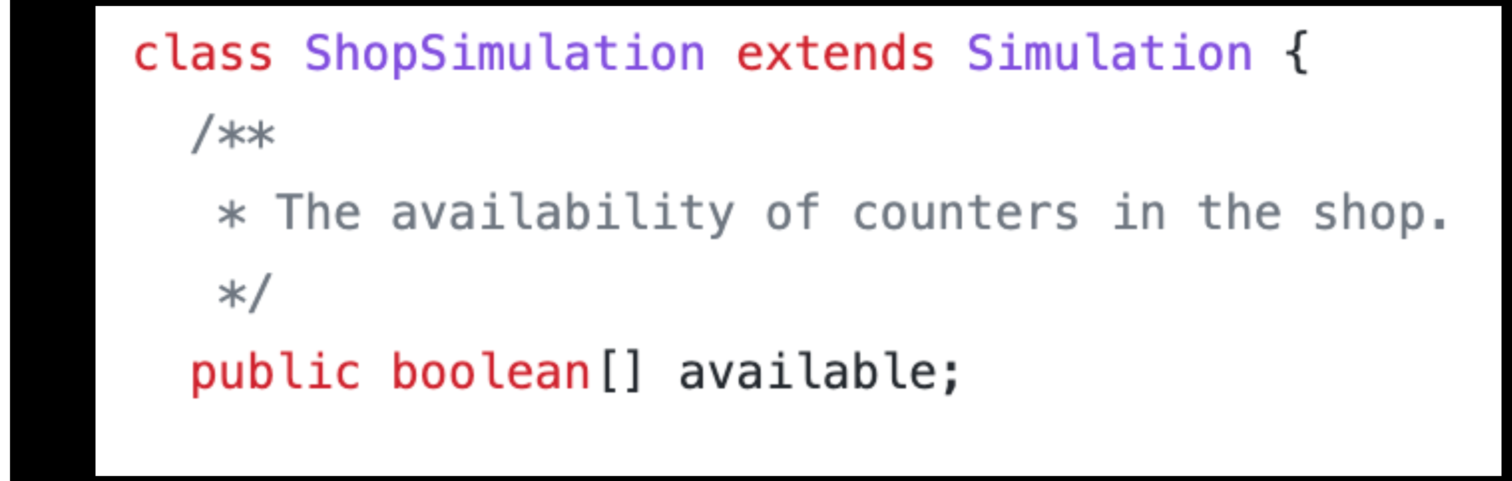
```
public class Simulator {
  /** The event queue. */
  private final PriorityQueue<Event> events;
```

Lab 1: Discrete Event Simulator (Part 1)

- Some of the violations:
  - Information hiding
    - ShopEvent
  - Encapsulation
    - ShopEvent
    - ShopSimulation



```java
class ShopEvent extends Event {
  /**
   * The id of a customer associated with this event.
   * First customer has id 0. Next is 1, 2, etc.
   */
  public int customerId;

  /**
   * A tag to indicate what type of event this is.
   * Possible types are ARRIVAL, SERVICE_BEGIN,
   * SERVICE_END and DEPARTURE.
   */
  public int eventType;

  /**
   * The service time of the customer associated
   * this event.  This field matters only if the
   * event type is ARRIVAL or SERVICE_BEGIN.
   */
  public double serviceTime;
```



```java
class ShopSimulation extends Simulation {
  /**
   * The availability of counters in the shop.
   */
  public boolean[] available;
```

Happy coding! 👨‍💻