Part B

Part A

A1

1Describe defensive copying in terms of a reference parameter to a constructor of a class. [2 marks]

If you just pass a parameter of a type that is a mutable, then if you just copy that straight into the class then you can change it. i.e Date example.

2 What is an immutable class? Give two advantages of immutable classes.

Class who’s instance cannot be changed after it’s been initialized.

Thread safe;Simplicity;make good building blocking.

3 1: Explain the meaning of the final qualifier in the Book class declaration and in the declaration of its member fields.

Final class- one that you can’t inherit from; Final field – once initialized can’t be changed.

2: Explain why instances of both Author and Book are mutable.

Author has set method; Books does not defensively copy; final for class and field.

3. Re-write the Book class to make it immutable (without modifying the Author class).

4. Briefly describe the changes necessary to make Author immutable. If these changes were made, would it still be necessary to re-write the Book class? Briefly explain your answer.

Make the class and member fields final and remove the set method; Yes because date…. Is still mutable.

A2

1 Object factories are an alternative to constructors for the controlled instantiation of objects. Give an example of the application of the object factory design pattern in the standard Java libraries.

Object factories are used in Collections framework to provide wrapper implementations such as unmodifiable and thread-safe synchronized views of an underlying collection

2 List three advantages of an object factory compared with the use of constructors for object instantiation. [3 marks]

1. Hiding implementation detail 2.user does not need to know the concrete implementation of the type. 3.factory determines which implementation to use 4.the choice of implementation can be deferred to run-time 5.more meaningful name than constructor. 6.static factory method need not to create a new object each time they are invoked.

3

1) Explain why it is not necessary to define a separate interface for the ModuleCode class?

No need to define an interface because modulecode is final ->there will only ever be one implementation.

2) There are no constraints on the values of prefix and number passed to the constructor of the ModuleCode class. Suggest appropriate invariants with respect to the member fields and the exception(s) to throw if the parameters do not comply with the invariants. [3 marks]

a. prefix non-null ,not empty length>0,can specify length==3 :throw NullPointerExepction if null or IllegalArgumentException if empty or does not comply with specified length.

b. number should be >0 (maybe 1000-9999) IllegalArgumentException if not in range.

3) change to factory mode

4) Do you need to override the Object equals method in your modified class? Briefly explain why. [3 marks]

Question B1

1. Describe and illustrate with a diagram the life cycle of a thread from being created to termination. List the ways by which a thread can be blocked and unblocked. [10 marks]

P40

Blocked: 1.wait 2.sleep 3.others join 4 wait for resource like lock

Unblocked: 1.signal 2.join 3.notify

b) Write a program that creates two threads, one to print out “Hip, hip!”, and the other to print out “Hooray!” (both in finite counting loops). After creating the two threads, the main

program should terminate. [10 marks]

1. Modify your program so that the thread printing out “Hip, hip!” would run first, and the thread printing out “Hooray!” be allowed to run only after the “Hip, hip!” thread has finished. [5 marks]

都看代码啦

Question B2

1. What are the three key features of a monitor and how are they used to support concurrent programming? [6 marks]

一样

b) A bridge over a river is only wide enough to permit a single lane of traffic. A collision can occur if two cars moving in different directions enter at the same time. We can call the cars moving in one direction “red cars” and the cars moving in the opposite direction “blue cars”. Write a monitor SafeBridge (with redEnter, redExit, blueEnter, blueExit operations) to control access to the bridge, so that the cars moving in different directions cannot concurrently use the bridge. [12 marks]

c) Provide an implementation of RedCar class (or BlueCar class) describing the behaviour of red (or blue) car threads which access the safe bridge from part b). You may assume that a journey across the bridge takes 5 seconds. [7 marks]

剩下都是看代码了