

Xi'an Jiaotong-Liverpool University

西交利物浦大学

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CSE210		Computer Science & Software Engineering	

SECOND SEMESTER 2017/2018 FINAL EXAMINATIONS

BACHELOR DEGREE – Year 3

Advanced Object Oriented programming

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

- 1、 Total marks available are 100. This will count for 50% in the final assessment.**
- 2、 Answer FOUR questions only.**
- 3、 The number in the column on the right indicates the marks for each section.**
- 4、 Answer should be written in the answer booklet(s) provided.**
- 5、 The university approved calculator - Casio FS82ES/83ES can be used.**
- 6、 All the answers must be in English.**

Question 1. A Java package called 'cn.edu.xjtlu.csse' contains the following classes. The class Node is an inner class of the abstract class LinkedList. Answer the following questions.

[25 marks]

```
abstract class LinkedList{
    private Node head;

    public void add(Object obj){
        head = new Node(obj, head);
    }

    public abstract int length();

    protected void addend(Object obj){
        if (head == null){
            head = new Node(obj);
        }else{
            Node n = head;
            while (n.tail() != null){
                n = n.tail();
            }
            n .addend(obj);
        }
    }

    class Node{
        private Object value;
        private Node tail;

        public Node(Object obj){
            value = obj;
        }

        public Node(Object obj, Node t){
            //implement your code here
        }

        public Object head(){
            return value;
        }

        public Node tail(){
            return tail;
        }

        public void addend(Object obj){
```

```
        tail = new Node(obj);  
    }  
}  
}
```

- a) The class Node has two private fields, value and tail. Are they visible to the methods of the class LinkedList? [2 marks]
- b) The class LinkedList above is defined as an abstract class. Briefly describe the characteristics of an abstract class. [3 marks]
- c) LinkedList has one abstract method called length() which is intended to calculate the length of the list. Provide an implementation for this abstract method. [6 marks]
- d) In another package called 'cn.edu.xjtlu.eee', there are two classes, MyLinkedList which is a sub-class of the LinkedList, and ListTest. To which of the classes, MyLinkedList and ListTest, is the protected method addend() visible? [4 marks]
- e) Describe briefly how to use generics to re-define the class LinkedList. [4 marks]
- f) Method addend() in LinkedList makes a call to a method with the same name. Are these two methods the same? Why. [2 marks]
- g) What are the characteristics of static inner classes? [4 marks]

Question 2. Consider the Java code below which implements a Binary Search Tree and answer the following questions.

[25 marks]

```
public class BSTree<X extends Comparable<X>> {

    private BSTree<X> leftSubtree;
    private X value;
    private BSTree<X> rightSubtree;

    public BSTree(BSTree left, X val, BSTree right) {
        leftSubtree = left;
        value = val;
        rightSubtree = right;
    }

    public X getValue() {
        return value;
    }

    public boolean isInOrder() {
        if (leftSubtree == null) {
            return value.compareTo(rightSubtree.getValue()) < 0;
        } else if (rightSubtree == null) {
            return value.compareTo(leftSubtree.getValue()) > 0;
        } else {
            return leftSubtree.getValue().compareTo(value) < 0 &&
value.compareTo(rightSubtree.getValue()) < 0;
        }
    }

    public static void main(String[] args) {
        BSTree<Integer> bt = new BSTree<Integer>(null, new Integer(7),
null);
        bt = new BSTree<Integer> (bt, new Integer(10), null);
        System.out.println(bt.isInOrder());
    }
}
```

a) The code above uses generics. In this context, what is meant by 'erasure'?

[2 marks]

b) What will the constructor of BSTree be after erasure?

[4 marks]

- c) Briefly describe the function of the method-call stack in the Java interpreter.

[4 marks]

- d) The main() method has three statements. Describe the state of the method-call stack when the main() method is executed.

[9 marks]

- e) A class invariant for the BSTree can be defined as: (1) each value in the left subtree is strictly less than the internal label; and (2) each value in the right subtree is strictly greater than the internal label. Does the implementation above preserve the class invariant? Justify your answer.

[6 marks]

Question 3. Class BankQueue implements a queue for counter service at a bank. The counter can accommodate at most 20 clients at any given time. The method addClient() is called if a new client can be added into the queue. If a client has been serviced the getNextClient() or remove() method is called. Assume that the method calls can only be performed one by one. Answer the following questions.

[25 marks]

```
public class BankQueue {
    private int[] clients = new int[20];
    private int top = 0;

    public int getNextClient(){
        int first = clients[0];
        for (int i = 0; i + 1 < top; i++) {
            clients[i] = clients[i + 1];
        }
        top--;
        return first;
    }

    // int i is the service code
    public void addClient(int i) {
        clients[top++] = i;
    }

    public void remove() {
        System.out.println(getNextClient());
    }

    public static void main(String[] args) {
        BankQueue q = new BankQueue();
        q.addClient(3);
        q.addClient(7);
        q.remove();
    }
}
```

a) Is 'top >= 0' a class invariant for BankQueue? Justify your answer.

[4 marks]

b) What is the difference between a checked exception and an unchecked exception?

[2 marks]

- c) Define a checked exception class `EmptyBankQueueException` for the `BankQueue`. You need to provide a constructor and overwrite the `getMessage()` method.

[4 marks]

- d) Modify the `getNextClient()` method so it can throw an `EmptyBankQueueException` when the queue is empty.

[6 marks]

- e) With the modified `getNextClient()` method, what other changes would be necessary for the `BankQueue` class?

[4 marks]

- f) In java programming, what is the preferred place to use the 'try-catch' to handle exceptions? Why?

[5 marks]

Question 4. The RGB color model is widely used for representation and display of images in electronic systems. In this model, a color can be represented by three integers which stand for primary color values (red, green and blue). Consider the class RGBColor below and answer the following questions. It is assumed that the IllegalArgumentException class has been developed.

[25 marks]

```
public class RGBColor {
    private int red;
    private int green;
    private int blue;
    private String name;

    // Values must be between 0 and 255.
    private void check(int red, int green, int blue) {
        if (red < 0 || red > 255
            || green < 0 || green > 255
            || blue < 0 || blue > 255) {
            throw new IllegalArgumentException();
        }
    }

    public RGBColor(int red, int green, int blue, String name) {
        check(red, green, blue);
        this.red = red;
        this.green = green;
        this.blue = blue;
        this.name = name;
    }

    public void set(int red, int green, int blue, String name) {
        check(red, green, blue);
        this.red = red;
        this.green = green;
        this.blue = blue;
        this.name = name;
    }

    public String getName() {
        return name;
    }
}
```

- a) Multithreading in Java is typically implemented by time-slicing. Briefly describe how the time-slicing works.

[4 marks]

- b) If multiple threads access an instance of the RGBColor class at the same time, what kind of problem might happen? Provide an example to describe the problem.

[4 marks]

- c) What changes to the class RGBColor would be necessary to solve the problem in Question 2b) if there is any?

[6 marks]

- d) What is deadlock in Java multi-threading? How to prevent the deadlock problem in Java multi-threading programming?

[5 marks]

- e) Briefly describe what Event-Dispatching Thread is in Java GUI programming and discuss if Java Swing components suffer from the interference problem.

[6 marks]

Question 5. Answer the following questions.

[25 Marks]

- a) Define an interface called 'Function' using Java's Generic Types. The interface should have two methods. The first method is `compute()`, which takes input of type A and produces output (as return-type) of type B. The second method is `printName()`, which prints the name of its implementing class.

[4 marks]

- b) Define an abstract class called 'AbstractFunction' that implements the Function interface. It should implement `printName()` method. The method `compute()` should be defined as abstract.

[5 marks]

- c) Define a class called 'DuplicateFunction' that extends the AbstractFunction class. It should implement the `compute()` method, which takes a String as input and returns another String by duplicating the input string.

[4 marks]

- d) Define a class called 'RepeatFunction' that extends the AbstractFunction class. It should define two fields, an integer 'count' and a function 'fun' (of type Function). It should implement the `compute()` method, which repeatedly calls the fun's `compute()` method for the 'count' number of times.

[8 marks]

- e) Suppose the `main()` method in a test class is as follows. What is the output?

```
public static void main(String[] args) {  
    DuplicateFunction df = new DuplicateFunction();  
    RepeatFunction rf = new RepeatFunction(1, df);  
    System.out.println(rf.compute("ObjectOrientation"));  
}
```

[4 marks]

END OF EXAM PAPER