



厦门大学《Java 程序设计》课程试卷

软件学院 软件工程系 07 年级 软件工程 专业

主考教师：王美红 试卷类型：(A 卷/B 卷)

1、 Multiple Choice(20 point)

Identify the letter of the choice that best completes the statement or answers the question.

- (1) The ____ statement identifies a block of statements that potentially may throw an exception.
- a. catch
 - b. finally
 - c. throw
 - d. try
- (2) Which method is used to invoke a constructor of the same class?B
- a. this()
 - b. constructor()
 - c. super()
 - d. class()
- (3) The ____ is a wildcard symbol to tell a program to import all necessary classes from a package.
- a. asterisk
 - b. comma
 - c. period
 - d. semicolon
- (4) Which of the following is not a reference type in Java?
- a. array
 - b. BufferedReader
 - c. float
 - d. String
- (5) The ____ class decodes bytes into **characters**.
- a. BufferedInputStream
 - b. BufferedReader
 - c. **InputStreamReader**
 - d. System.in
- (6) ____ is a class used to display standard dialog boxes.
- a. Box
 - b. JFrame
 - c. JOptionPane
 - d. JWindow
- (7) In java 2D, method _____ of class Graphics2D sets the characteristics of a line used to draw a shape.
- a. setBackground
 - b. setStroke
 - c. setPaint
 - d. translate
- (8) A _____ is needed to terminate recursion.
- a. recursion step
 - b. void return type
 - c. break statement
 - d. base case
- (9) What type of error will be produced first during the execution of the following code?

```
Object[] array = new Object[4];  
for(int i=0;i<= array.length;i++) System.out.println(array[i] );
```

- a. ArrayIndexOutOfBoundsException
- b. StringIndexOutOfBoundsException
- c. cannot resolve symbol
- d. NullPointerException

(10) The following is a list of classes / interfaces defined in Java's standard packages, List all interfaces in the above list.

- | | | |
|---------------|-------------|----------------|
| a. Collection | b. Math | c. List |
| d. Thread | e. Runnable | f. Collections |
| g. Integer | h. String | |

2、 True/False(10 points)

Indicate whether the sentence or statement is true or false.

- (1) Multiple inheritances are not allowed in Java. T
- (2) If I do not provide any arguments on the command line, then the String array of Main method will be null. F
- (3) java.lang package is by default loaded internally by the JVM. T
- (4) Importing a package imports the subpackages as well. e.g. Importing com.MyTest.* also import com.MyTest.UnitTests.*. T
- (5) The default value of an object reference declared as an instance variable is null. T
- (6) If I write System.exit (0); at the end of the try block, the finally block will still execute. F
- (7) When a thread is created and started, its initial state is runnable state. F
- (8) The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected. True
- (9) Static methods can be referenced with the name of the class rather than the name of a particular object of the class.
- (10) The local variables are not initialized to any default value, neither primitives nor object references.

3、 Short answer(25 points)

- (1) Which method can you use to perform binary search for elements in an ArrayList or a LinkedList? Which method can you use to sort an array of strings?(4 points)

- (2) Difference between **Vector** and **ArrayList**?(2 points)
- (3) What is the difference between method overloading and method overriding? Can static methods be overridden? Can constructor be overridden?(6 points)
- (4) What is the purpose of garbage collection in Java, and when is it used?(2 points)
- (5) Explain how the Java interface works. What is the difference between an abstract class and an interface? Produce examples of “implements” relationships.(3 points)
- (6) State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.(4 points)
- (7) What are Checked and UnChecked Exception?(2 points)
- (8) Within a method m in a class C, isn't this.getClass() always C? (2 points)

4、 Completion(45 points)

Complete each sentence or statement.

- (1) (2 points) An applet can also get references to all other applets on the same page using the getApplets() method of ____ (1) ____.
- (2) (5 points) Java applets begin execution with a series of three method calls:____ (2) ____, (3) ____ and ____ (4) ____ .The ____ (5) ____ method is invoked for an applet each time the user of a browser leaves an HTML page on which the applet resides. Every applet should extend class ____ (6) ____.
- (3) (4 points) Consider the following classes:

```
public class AnimalHouse<E> {  
    private E animal;  
    public void setAnimal(E x) {  
        animal = x;  
    }  
    public E getAnimal() {  
        return animal;  
    }  
}  
public class Animal{  
}  
  
public class Cat extends Animal {  
}  
  
public class Dog extends Animal {  
}
```

For the following code snippets(代码片段) from a to d, identify whether the code:

- fails to compile____(7)_____
- compiles with a warning____(8)_____
- generates an error at runtime____(9)_____or
- none of the above (compiles and runs without problem.)____(10)_____

- a. `AnimalHouse<Cat> house = new AnimalHouse<Cat>();`
- b. `AnimalHouse<Cat> house = new AnimalHouse<Animal>();`
- c. `AnimalHouse house = new AnimalHouse<Animal>();`
- d. `AnimalHouse house = new AnimalHouse();`
`house.setAnimal(new Dog());`

(4) (5 points) Assume the following:

`a = "abc";`

Give the values of the following expressions, or *illegal*.

____(11)____ `"Tomorrow".substring(2,4)`

____(12)____ `a.length() + a`

____(13)____ `a.length() + a.startsWith("a")`

____(14)____ `"a".compareTo("c")`

____(15)____ `"a = \"\" + a + \"\""`

(5) (5 points) Write a method to copy a file from file1 to file2:

```
public static void copy(File file1, File file2) {  
    _____(16)_____  
    _____  
}
```

(6) (6 points) Producer and Consumer are two threads which both implement interface Runnable. Complete the BlockingBufferTest class below to start the two threads.

```
public class BlockingBufferTest  
{  
    public static void main( String[] args )  
    {  
        // create new thread pool with two threads  
        ExecutorService application = _____(17)_____;  
  
        // create BlockingBuffer to store ints  
        Buffer sharedLocation = new BlockingBuffer();
```

```

try // try to start producer and consumer
{
    application.____(18)____ ( new Producer( sharedLocation ) );
    application.____(18)____ ( new Consumer( sharedLocation ) );
} // end try

catch ( Exception exception )
{
    exception.printStackTrace();
} // end catch

application.____(19)____ ;
} // end main
} // end class BlockingBufferTest

```

(7) (5 points) Establishing a simple server using stream sockets in java requires five steps.

Step 1, to create an object, such as

____(20)____ server = new ____ (20) ____ (portNumber,queueLength);

Step 2, the server listens indefinitely for an attempt by a client to connet. Such in

____(21)____ connection = server. ____ (22) ____;

Step 3, to get the OutputStream and ____ (23) ____ objects that enable that server to communicate with the client by sending and receiving bytes.

Step 4, is the processing phase.

Step 5, when the transmission is complete, the server closes the connection by invoking the ____ (24) ____ method on the streams and on the Socket.

(8) (4 points) Write method numInArray, as started below. numInArray should return the number of times an even int parameter occurs in array A. For example, assume that array A is as shown below.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
6	3	4	5	2	7	3

then numInArray(A) should return 3 since 6, 4, and 2 are even.

Complete method numInArray below.

```

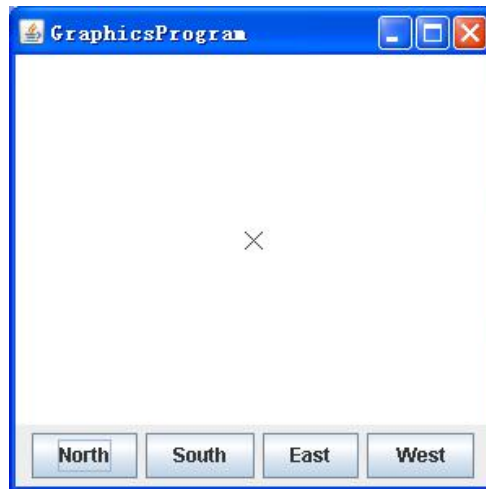
public static int numInArray(int[] A)
{
    // precondition: A will contain integers greater than zero
    // postcondition: returns the number of even integers found in A
    _____(25)_____
    _____
}

```

(9) (9 points) Complete the following program to do the following:

- a. Add buttons to the South region labeled "North", "South", "East", and "West".
- b. Create an X-shaped cross 10 pixels wide and 10 pixels high.
- c. Adds the cross so that its center is at the center of the graphics canvas.

Once you have completed these steps, the display should look like this:



- d. Implement the actions for the button so that clicking on any of these buttons moves the cross 20 pixels in the specified direction.

```
//GraphicsProgram.java
import java.awt.BorderLayout;

public class GraphicsProgram extends JFrame {

    private SouthPanel southPanel;
    private CenterPanel centerPanel;

    public GraphicsProgram(String title) {
        super(title);

        southPanel = new SouthPanel();
        southPanel.setAlignmentX(FlowLayout.CENTER);

        centerPanel = new CenterPanel();

        add(centerPanel, BorderLayout.CENTER);
        add(southPanel, BorderLayout.SOUTH);

    } //end constructor

    public static void main(String args[]) {
        GraphicsProgram frame = new GraphicsProgram("GraphicsProgram");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 400);
        frame.setVisible(true);
    }
}
```

```

}

class SouthPanel extends JPanel implements ActionListener {
    private JButton northButton, southButton, eastButton, westButton;

    public SouthPanel() {
        northButton = new JButton("North");
        southButton = new JButton("South");
        eastButton = new JButton("East");
        westButton = new JButton("West");
        northButton.addActionListener(this);
        southButton.addActionListener(this);
        eastButton.addActionListener(this);
        westButton.addActionListener(this);
        add(northButton);
        add(southButton);
        add(eastButton);
        add(westButton);
    } //end constructor

    public void actionPerformed(ActionEvent arg0) {
        _____
        _____
    } //end method actionPerformed.
} //end class SouthPanel

}

```

//CenterPanel.java

```

import java.awt.*;
import javax.swing.*;

public class CenterPanel extends JPanel {
    private int centerX; // the horizontal x point in the ¿Á center
    private int centerY; // the vertical y point in the ¿Á center

    private Boolean initState = true; // not have been moved

    public CenterPanel() {
        this.setBackground(Color.WHITE);
    } //end constructor

    public void paintComponent(final Graphics g) {
        _____
        _____
    }
}

```

```
}

//set and get method
public void setInit(Boolean init) {
    this.initState = init;
}

public void setX(int centerX) {
    if (centerX >= 0)
        this.centerX = centerX;
    paintComponent(this.getGraphics());
}

public void setY(int centerY) {
    if (centerY >= 0)
        this.centerY = centerY;
    paintComponent(this.getGraphics());
}

public int getX() {
    return centerX;
}

public int getY() {
    return centerY;
}

} //end class CenterPanel
```


07 级答案 by icebear

一、

D A A C C C B D A (aec)

二、

TFTTT FFTTT

三、

(1) Which method can you use to perform binary search for elements in an ArrayList or a LinkedList? Which method can you use to sort an array of strings?(4 points)

答：通过 `binarySearch()` 方法进行二分查找。通过 `sort()` 方法进行字符串数组排序。

(2) Difference between **Vector** and **ArrayList**?(2 points)

答：**ArrayList** 是同步的，**Vector** 不是同步的，因此 **ArrayList** 速度快，**Vector** 速度慢。

(3) What is the difference between method overloading and method overriding? Can static methods be overridden? Can constructor be overridden?(6 points)

答：

① **overloading** 和 **overriding** 的区别

(1) 方法重载是编译时多态性，使用静态绑定。方法覆盖是运行时多态性，使用动态绑定。

(2) 方法重载有助于提高程序可读性。方法覆盖对其父类或超类已提供方法进行具体实现。

(3) 进行方法重载时，参数列表应该不同，返回类型可以相同也可以不同。方法覆盖中的参数列表应该相同，返回类型应该相同或协变。

(4) 私有和最终方法可以重载，私有和最终方法不能被覆盖。

② **不能覆盖静态方法**。（因为方法覆盖基于运行时的动态绑定，而静态方法在编译时使用静态绑定进行绑定。所以不能覆盖静态方法。）

③ **不能被覆盖构造函数**。（如果尝试在子类中编写超类的构造函数，编译器会将其视为方法并期望返回类型并生成编译时错误。）

(4) What is the purpose of garbage collection in Java, and when is it used?(2 points)

答：(1) 垃圾回收的目的是识别和丢弃那些应用程序不再需要的对象，以便回收和重用资源。(2) 当程序无法访问 Java 对象时，就会进行垃圾回收。

(5) Explain how the Java interface works. What is the difference between an abstract class and an interface? Produce examples of “implements” relationships.(3 points)

答：

(1) 接口通常以 `interface` 来声明。一个类通过继承接口的方式，从而来继承接口的抽象方法。接口无法被实例化，但是可以被实现。

(2) 接口中只能有抽象方法，只能有静态常量。而**抽象类**中至少包括一个抽象方法， 但可以有若干非抽象方法， 且可以有静态和非静态的变量和常量。

1. 抽象类中的方法可以有方法体，就是能实现方法的具体功能，但是接口中的方法不行。
2. 抽象类中的成员变量可以是各种类型的，而接口中的成员变量只能是 **public static final** 类型的。
3. 接口中不能含有静态代码块以及静态方法(用 **static** 修饰的方法)，而抽象类是可以有静态代码块和静态方法。
4. 一个类只能继承一个抽象类，而一个类却可以实现多个接口。

(3) 类使用 **implements** 关键字实现接口。在类声明中，**Implements** 关键字放在 **class** 声明后面。比如 **public class Cat implements Animal** 就是 **Cat** 类对 **Animal** 接口进行实现。**Cat** 类中将会实现 **Animal** 接口中的所有方法。

(6) State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.(4 points)

答：

public: **public** 类在其他各种包中可见，成员变量同样在其他任何地方都可见。

protected : 可用于同一包中所有类，用于所有子类（可以是不同包的子类）。

default: 只要在同一包内便可使用。

private : **private** 变量或方法只能由声明 **private** 变量或方法的类实例使用。其他任何外部类都无法导入它。

	public	protected	default	private
同类	ok	ok	ok	ok
同包	ok	ok	ok	
子类(不同包)	ok	ok		
不同包中无继承关系的类	ok			

(7) What are Checked and UnChecked Exception?(2 points)

答：

(1) **Checked Exception**: **Exception** 的子类，但并非 **RuntimeException**。由不在程序控制范围内的条件引起。编译器检查每个方法调用和方法声明，以确定该方法是否抛出 **Checked Exception**。如果是，编译器将验证选中的异常是否被 **catch** 或者 **throws** 子句中声明。如果未满足 **catch** 或 **declare** 要求，编译器将发出一条错误消息，指示必须捕获或声明异常。

(2) **UnChecked Exception**: 类 **RuntimeException** 的直接或间接子类。通常是由程序代码中的缺陷引起的。编译器不检查代码以确定 **UnChecked Exception** 是否被捕获或声明。这些通常可以通过适当的编码来防止，不需要在方法的抛出子句中列出 **UnChecked Exception**。

(8) Within a method **m** in a class **C**, isn't this **getClass()** always **C**? (2 points)

答：不，如果实例化一个 **C** 的子类 **C1** 对象 **c1**。则使用 **c1** 去调用方法 **m** 去 **getClass** 得到的是子类 **C1**。

四、

- (1) java.applet.AppletContext
- (2) init (3) start (4) paint (5) stop (6) JApplet
- (7) b (8) cd (9) (10) a

注：上机调试发现没有 error at runtime 的情况。

- (11) mo (12) 3abc (13) illegal (14) -2 (15) a="abc"
- (16) Files.copy(source.toPath(),dest.toPath());
- (17) Executors.newCachedThreadPool() (18) execute (19) shutdown
- (20) ServerSocket (21) Socket (22) accept() (23) InputStream() (24) Socket.close();
- (25)

```
public class test {  
    public static void main(String args[]) {  
        int[] A= {6,3,4,5,2,7,3};  
        System.out.println(numInArray(A));  
    }  
    public static int numInArray(int[] A) {  
        int count=0;  
        for(int n:A) {  
            if(n%2==0) count++;  
        }  
        return count;  
    }  
}
```

- (26) 略