

---

《JA V A 程序设计》期末考试试题(三)

一、单项选择题

1、如下哪个是 Java 中的标识符(D)

A、public

B、super

C、3number

D、width

2、如下哪个是 Java 中的标识符(A)

A、fieldname

B、super

✗

C、3number

✗

D、#number

3、已知如下定义:String s = "story"; 下面哪个语句不是合法的(C)

A、s += "books";

B、s = s + 100;

C、int len = s.length;

D、String t = s + "abc";

4、如下哪个是 Java 中有效的关键字(C)

A、name

---

B、hello

C、false

D、good

5、下面的代码段执行之后 count 的值是什么( D

```
int count = 1;
```

```
for (int i = 1; i <= 5; i++ {
```

```
count += i;
```

*1+2 count=count+1*

```
}
```

*2+ 2 3 4 5 6*

```
System.out.println(count;
```

A、5

B、1

C、15

D、16

6、定义一个类,必须使用的关键字是( B

A、public

B、class

C、interface

D、static

7、定义一个接口必须使用的关键字是(C

---

A、public

B、class

C、interface

D、static

8、如果容器组件 p 的布局是 BorderLayout,则在 p 的下边中添加一个按钮 b,应该使用的语句是(C

A、p.add(b;

B、p.add(b,"North";

C、p.add(b,"South";

~~D、b.add(p,"North";~~

9、声明并创建一个按钮对象 b,应该使用的语句是(A

A、Button b=new Button(;

B、button b=new button(;

C、Button b=new b(;

D、b.setLabel("确定";

10、Frame 对象默认的布局管理器是(B

A、FlowLayout

B、BorderLayout

C、CardLayout

---

D、null

11、下列哪一个 import 命令可以使我们在程序中创建输入/输出流对象(C

A、import java.sql.\*;

B、import java.util.\*;

C、import java.io.\*;

D、import java.net.\*;

12、下面哪一个 import 命令可以为我们提供编写网络应用程序的类(D

A、import java.sql.\*;

B、import java.util.\*;

C、import java.io.\*;

D、import java.net.\*;

13、如果需要从文件中读取数据,则可以在程序中创建哪一个类的对象(A

A、FileInputStream

B、FileOutputStream

C、DataOutputStream

D、FileWriter

## 二、填空题

1、如果将类 MyClass 声明为 public,它的文件名称必须是 (MyClass.java) 才能正常编译。

- 
- 2、Java 程序中的单行注释符是 `//`，多行注释符是 `/* */`。
- 3、Java 中布尔类型的常量有两种，它们是 `true` 和 `false`。
- 4、Java 中用于定义小数的关键字有两个：`float` 和 `double`，后者精度高于前者。
- 5、Java 中用于两个数相等比较的运算符是：`==`，用于不相等比较的运算符是 `<>`。或 `!=` 字符串 `s1.equals(s2)`。
- 6、在 Java 中定义一个字符串类型的变量 `str` 的语句是：`String str;`，定义一个具有 10 个元素的整型数组 `a` 的语句是：`int [] arr = new int[10];`。
- 7、导入 `mypackage` 包中的所类的命令是 `import mypackage.*;`。
- 8、当声明一个数组 `int arr[] = new int[5];` 时，这代表这个数组所保存的变量类型是 `int`，数组名是 `arr`，数组的大小为 `5`，数组元素下标的使用围是 `0` 到 `4`。
- 9、假设 `x=13,y=4`，则表达式 `x%y != 0` 的值是 `true`，其数据类型是 `boolean`。
- 10、异常处理是由 `try`、`catch` 和 `finally` 块三个关键所组成的程序块。

#### 四、简答题

1、简述 Java 中异常处理的机制？

答：首先 Java 的异常是面向对象的。一个 Java 的 `Exception` 是一个描述异常情况的对象。当出现异常情况时，一个 `Exception` 对象就产生了，并放到异常的成员函数里。

Java 的异常处理是通过 5 个关键词来实现的：`try`，`catch`，`throw`，`throws` 和 `finally`。

---

1. 异常处理:在 Java 语言的错误处理结构由 try,catch,finally 三个块组成。其中 try 块存放将可能发生异常的 Java 语言,并管理相关的异常指针;catch 块紧跟在 try 块后面,用来激发被捕获的异常;finally 块包含清除程序没有释放的资源,句柄等。不管 try 块中的代码如何退出,都将执行 finally 块。

2. 抛出异常:Java 语言可以不在方法中直接捕获,而用 throw 语句将异常抛给上层的调用者。Throw 语句就是来明确地抛出一个异常;首先你必需得到一个 Throwable 的实例句柄,通过参数传到 catch 中,或者采用 new 操作符来创建一个。

## 2、什么是继承?

答:通过必要的说明能够实现某个类无需重新定义就拥有另一个类的某些属性和方法,并把这种关系称为继承,先定义类称为父类,后定义类称为子类,并且允许多层的继承关系。

## 3、请简述重载和重写的区别?

答:方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。

重写 Overriding 是父类与子类之间多态性的一种表现,重载 Overloading 是一个类中多态性的一种表现。

如果在子类中定义某方法与其父类有相同的名称和参数,我们说该方法被重写(Overriding。子类的对象使用这个方法时,将调用子类中的定义,对它而言,父类中的定义如同被“屏蔽”了。

如果在一个类中定义了多个同名的方法,它们或有不同的参数个数或有不同的参数类型,则称为方法的重载(Overloading。Overloaded 的方法是可以改变返回值的类型。

## 五、程序设计题

### 1、按以下要求编写程序

---

(1 创建一个 Rectangle 类,添加 width 和 height 两个成员变量

(2 在 Rectangle 中添加两种方法分别计算矩形的周长和面积

(3 编程利用 Rectangle 输出一个矩形的周长和面积

解答:

```
public class Rectangle {  
  
    float width, height;  
  
    public Rectangle(float width, float height {  
  
        this.width = width;  
  
        this.height = height;  
  
    }  
  
    public float getLength({  
  
        return (this.width + this.height * 2;  
  
    }  
  
    public float getArea({  
  
        return this.width * this.height;  
  
    }  
  
    public static void main(String [] args {  
  
        Rectangle rect = new Rectangle(10, 20);  
  
        System.out.println("周长是:" + rect.getLength(;
```

---

```
System.out.println("面积是:" + rect.getArea();
```

```
}
```

```
}
```

2、按以下要求编写程序

(1 编写 Animal 接口,接口中声明 run( 方法

(2 定义 Bird 类和 Fish 类实现 Animal 接口

(3 编写 Bird 类和 Fish 类的测试程序,并调用其中的 run(方法解答:

```
public interface Animal {
```

```
void run(;
```

```
}
```

```
class Bird implements Animal {
```

```
public void run( {
```

```
System.out.println("鸟儿在飞...");
```

```
}
```

```
}
```

```
class Fish implements Animal {
```

```
public void run( {
```

```
System.out.println("鱼儿在游...");
```

```
}
```



---

```
}

public class TestAnimal {

    public static void main(String[] args) {

        Bird bird = new Bird();
        bird.run();

        Fish fish = new Fish();
        fish.run();

    }

}
```

3、编写一个Frame 框架应用程序,要求如下:

- (1 在窗口设置两个菜单“文件”、“编辑”
- (2 在“文件”菜单里添加三个菜单项“打开”、“保存”、“关闭”
- (3 在“编辑”菜单里添加两个菜单项“复制”、“粘贴”
- (4 点击关闭菜单项时,使程序关闭。

解答:

```
import java.awt.Color;

import java.awt.Frame;

import java.awt.Menu;

import java.awt.MenuBar;
```

---

```
import java.awt.MenuItem;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

/*

 * 主窗体

 */

public class MyFrame extends Frame {

    private static final long serialVersionUID = 94L;

    private Frame frmMain; // 窗体

    MenuBar mb; // 菜单栏

    Menu mnuFile; // "文件"菜单

    Menu mnuEdit; // "编辑"菜单

    MenuItem miOpen; // "打开"菜单项

    MenuItem miSave; // "保存"菜单项

    MenuItem miClose; MenuItem miCopy; MenuItem miPaste; public

    MyFrame( { // "关闭"菜单项 // "复制"菜单项 // "粘贴"菜单项 frmMain = new

    Frame("主窗体"; // 创建菜单栏 mb = new MenuBar(; // 创建菜单 mnuFile =

    new Menu("文件"; mnuEdit = new Menu("编辑"; // 创建菜单项 miOpen = new

    .. ..
```

---

```

MenuItem("打开"; miSave = new MenuItem("保存"; miClose = new MenuItem("
关闭"; miCopy = new MenuItem("复制"; miPaste = new MenuItem("粘贴"; }
public void showFrame( { frmMain.setSize(800, 600;
frmMain.setLocation(100, 100; frmMain.setBackground(Color.white;
frmMain.setVisible(true; frmMain.setLayout(null;
frmMain.addWindowListener(new WindowHandler(); // 注册窗口监听器 // 将菜
单栏放入窗体中 frmMain.setMenuBar(mb; // 将菜单放入菜单栏中
mb.add(mnuFile; mb.add(mnuEdit; // 将菜单项放入菜单中 mnuFile.add(miOpen;
mnuFile.add(miSave; mnuFile.add(miClose; mnuEdit.add(miCopy;

    mnuEdit.add(miPaste; // 注册动作事件监听器
miClose.setActionCommand("miClose_Clicked";
miClose.addActionListener(new MenuHandler(); } // 定义一个部类，在这个类
中编写窗口关闭事件处理程序。 private class WindowHandler extends
WindowAdapter { public void windowClosing(WindowEvent e { System.exit(0;
// 退出应用程序，状态码 0 表示正常终止，非 0 表示异 常终止。 } } // 定义
一个部类，在这个类中编写动作事件处理程序。 private class MenuHandler
implements ActionListener { public void actionPerformed(ActionEvent e
{ if (e.getActionCommand() == "miClose_Clicked" { System.exit(0; // 退出
应用程序 } } } } public class TestMyFrame { public static void
main(String[] args { // 启动主窗体 MyFrame guiWindow = new MyFrame(;
guiWindow.showFrame(; } }

```