**《JAVA语言程序设计》期末考试试题及答案3（应考必备题库）**

**一、单项选择题**

1、如下哪个是Java中的标识符（）

A、public B、super C、3number **D、**width

2、如下哪个是Java中的标识符( )

**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( )

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()

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( )

**int** count = 1;

**for** (**int** i = 1; i <= 5; i++) {

count += i;

}

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( )

A、public **B、**class C、interface D、static

7、定义一个接口必须使用的关键字是（ ）

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（ ）

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、**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（ ）

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（）

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（）

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、**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（**<>**）。

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范围是（**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。

11、以下程序段的输出结果是（**三角形**）

**int** x = 5, y = 6, z = 4;

**if** (x + y > z && x + z > y && z + y > x)

System.*out*.println("三角形");

**else**

System.*out*.println("不是三角形");

12、下面程序段的执行结果是（**6 5 4 3 2**）

**int** a[] = { 2, 3, 4, 5, 6 };

**for** (**int** i = a.length - 1; i >= 0; i--)

System.*out*.print(a[i] + "");

**三、程序阅读题**

**1、**以下程序的输出结果为\_ **Peter is 17 years old!**\_\_\_。

**public** **class** Person {

String name;

**int** age;

**public** Person(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** **static** **void** main(String[] args) {

Person c = **new** Person("Peter", 17);

System.*out*.println(c.name + " is " + c.age + " years old!");

}

}

**2、**以下程序的输出结果为\_\_**课程号:101 课程名:ASP 学分:3**\_\_\_。

**public** **class** Course {

**private** String cNumber;

**private** String cName;

**private** **int** cUnit;

**public** Course(String number, String name, **int** unit) {

cNumber = number;

cName = name;

cUnit = unit;

}

**public** **void** printCourseInfo() {

System.*out*.println("课程号:" + cNumber + " 课程名:" + cName + " 学分:" + cUnit);

}

}

**class** CourseTest {

**public** **static** **void** main(String[] args) {

Course c;

c = **new** Course("101", "ASP", 3);

c.printCourseInfo();

}

}

**3、**以下程序的输出结果为\_\_**汤姆猫体重：20.0斤**\_\_\_。

**public** **class** Tom {

**private** **float** weight;

**private** **static** String *name*;

**public** **void** setWeight(**float** weight) {

**this**.weight = weight;

}

**private** **void** out() {

System.*out*.println(*name* + "体重：" + weight + "斤");

}

**public** **static** **void** main(String[] args) {

Tom.*name* = "汤姆猫";

Tom cat = **new** Tom();

cat.setWeight(20);

cat.out();

}

}

**4、**以下程序的输出结果\_\_。

**public** **class** Father {

String name, address, tel;

**int** age;

**public** Father(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**void** out() {

System.*out*.print("姓名:" + name);

System.*out*.print(" 年龄:" + age);

}

**void** outOther() {

System.*out*.print(" 家庭住址:" + address);

System.*out*.print(" 电话:" + tel);

}

}

**class** Son **extends** Father {

String school;

**public** Son(String name, **int** age) {

**super**(name, age);

}

**void** out() {

**super**.out();

**super**.outOther();

System.*out*.println(" 学校：" + school);

}

**public** **static** **void** main(String args[]) {

Son son = **new** Son("Tom", 15);

son.address = "金水区";

son.school = "九中";

son.tel = ;

son.out();

}

}

**5、**下列程序的运行结果是\_\_**12345**\_\_\_\_。

**public** **class** MyClass {

**int** a[] = { 1, 2, 3, 4, 5 };

**void** out() {

**for** (**int** j = 0; j < a.length; j++)

System.*out*.print(a[j] + "");

}

**public** **static** **void** main(String[] args) {

MyClass my = **new** MyClass();

my.out();

}

}

**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，先定义的类称为父类，后定义的类称为子类，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**

**import**

**import**

**import**

**import**

**import**

**import**

**import**

**import**

/\*

\* 主窗体

\*/

**public** **class** MyFrame **extends** Frame {

**private** **static** **final** **long** *serialVersionUID*

**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; // "关闭"菜单项

MenuItem miCopy; // "复制"菜单项

MenuItem miPaste; // "粘贴"菜单项

**public** MyFrame() {

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());

}

// 定义一个内部类，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。

}

}

// 定义一个内部类，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();

}

}