

# Java 程序设计期末考试样卷

一、单选题（共 10 题，每题 2 分，共 20 分）

1. 下面 4 种类型的文件中（ ）可以在 Java 虚拟机中运行。

A..java 源文件 B..jre 运行环境 C..exe 可执行程序 D..class java 字节码文件的扩展名

2. 下列选项中，不属于基本数据类型的是（ ）

A.String 字符串数据类型 B.short 短整型 C.boolean 布尔型 D.char 字符型

3. 假设 `int x=2`，三元表达式 `x>0?x+1:5`（ ）

A.0 B.2 C.3 D.5 成立：不成立

4. 阅读下列程序：

```
class Person{
    void say(){
        System.out.println("hello");
    }
}

class Example{
    public static void main(String[] args){
        Person p2 = new Person();
        Person p1 = new Person();
        p2.say();
        p1.say();
        p2=null;
        p2.say();
    }
}
```

下列选项中描述正确的是（ ）

A.输出 1 个 hello B.输出 2 个 hello 后会抛出异常  
C.输出 3 个 hello 后会抛出异常 D.不会输出 hello，直接抛出异常

5. 下列关于继承的描述中，错误的是（ ）

A.Java 中一个类只能有一个直接父类  
B.多个类可以继承一个父类  
C.一个类可以派生出很多个类

D.Java 是支持多继承的

6. 在异常处理时，释放资源关闭文件一般在( )语句块内完成。

A.try B.catch C.finally D.throw

7. 以下关于 String 类的常见操作中，哪个是方法会返回指定字符 ch 在字符串中最后一次出现位置的索引（ ）

A.int indexOf(int ch) B.int lastIndexOf(int ch)  
C.int indexOf(String str) D.int lastIndexOf(String str)

8. String s="Computer"; 则 s.substring(3,4)返回的字符串是（ ）。

A.u B.ut C.p D.pu

9. 使用 Iterator 时，判断是否存在下一个元素可以使用（ ）方法。

A.hasNext() B.hash() C.hasPrevious() D.next()

10. 下列选项中，`FileWriter` 类中 `read()` 方法读取到流末尾的返回值是 ( )

A.0      B. -1      C.1      D.无返回值

二、填空题 (共 10 题, 每题 2 分, 共 20 分)

1. 类封装时会使用\_\_\_\_\_关键字将类中的属性私有化。
2. \_\_\_\_\_关键字可用于修饰类、变量和方法, 它有“无法改变的”或者“最终”的含义。
3. 异常分为两种, 分别是\_\_\_\_\_和编译时异常。
4. \_\_\_\_\_关键字修饰的变量可以直接通过类名进行访问。
5. 已知 `strb` 为 `StringBuffer` 的一个实例, 且 `strb.toString()` 的值为 "abcdefg", 则执行 `strb.reverse()` 后, `strb.toString()` 的值为\_\_\_\_\_。
6. Java 中的 `boolean` 基本数据类型只有 2 个值, 分别是\_\_\_\_\_和 `false`。
7. `Map` 集合中的元素都是成对出现的, 并且都是以 `Key` (键)、\_\_\_\_\_的映射关系存在。
8. 在图形用户界面编程中, 鼠标按下按钮所触发的事件是\_\_\_\_\_。
9. Java 中的 I/O 流, 按照传输数据的不同, 可分为字节流和\_\_\_\_\_。
10. 在 `Thread` 类中, 提供了一个\_\_\_\_\_方法用于启动新线程。

三、程序阅读题 (共 6 题, 每题 5 分, 共 30 分)

```
1. class Test{
    public static void main(String[] args) {
        int a = 21;
        int b = 5;
        System.out.print(a>=b);
    }
}
```

输出结果是:

```
2. public class Test {
    public static void main(String[] args) {
        int[] intArray={1,2,3,4,5,6};

        int s=0;
        for(int i=0;i<intArray.length;i++)
            s+=intArray[i];
        System.out.println(s);
    }
}
```

输出结果是:

```
3. public class Test {
    public static void main(String[] args) {
        String str1 = "Hello";
        String str2 = "Hel" + "lo";
        System.out.println(str1 == str2);
        System.out.println(str1.equals(str2));
    }
}
```

输出结果是:

```

4. import java.util.Scanner;
public class ExceptionTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        try {
            System.out.print(a / b);
        } catch (ArithmeticException e) {
            System.out.print(5);
        } finally {
            System.out.println(0);
        }
    }
}

```

如果从键盘上输入 16 4，则程序的输出结果是：

```

5. class Father{
    Father(){
        System.out.println("Father");
    }
}

class Son extends Father{
    Son() {
        System.out.println("Son");
    }
    public static void main(String[] args) {
        new Son();
    }
}

```

输出结果是：

```

6.class Father{
    public void fun(){
        System.out.println("Father");
    }
}

class Son extends Father{
    public void fun() {
        System.out.println("Son");
    }
}

public class Test {

```

```
public static void main(String[] args) {  
    Father f = new Son();  
    f.fun();  
}  
}
```

输出结果是：

四、程序设计题（图书馆阅读材料管理系统, 共 5 题, 每题 6 分, 共 30 分）

1. 定义一个接口 Readable, 包含方法 getTitle() 用于获取阅读材料标题。(6 分)
2. 封装一个图书类 Book, 实现 Readable 接口, 包含成员变量 title 和 author, 并实现接口中的方法, 重写 toString 方法。(6 分)
3. 封装一个杂志类 Magazine, 实现 Readable 接口, 包含成员变量 title 和 issueNumber, 并实现接口中的方法, 重写 toString 方法。(6 分)
4. 封装图书馆类 Library, 包含方法 addReadable(Readable readable) 用于添加阅读材料到图书馆, 和 displayReadables() 用于展示图书馆中所有阅读材料信息。(6 分)
5. 创建测试类 LibraryTest, 在图书馆中添加两本图书, 例如“红楼梦”(作者: 曹雪芹) 和“平凡的世界”(作者: 路遥), 然后再添加两本杂志, 例如“读者”(刊号: 62-1190/Z) 和“软件”(刊号: 12-1151/TP)。最后, 展示图书馆中所有图书和杂志的信息。(6 分)