

**注意：请将答案按题号和序号写在答题纸上**

**一、选择题（单选，每小题 2 分，共 30 分）**

1. `class MyClass{ }`是自定义类，则 `MyClass.class` 代表的是（ ）。  
A. 方法调用      B. `MyClass` 类的非静态域      C. 一个对象      D. `MyClass` 类的静态域
2. 接上题，“`MyClass o1=new MyClass(); MyClass o2=new MyClass();`”这两条语句执行后，表达式“`o1==o2`”的取值是（ ）。  
A. 0              B. 1              C. true              D. false
3. 接上题，表达式“`o1.equals(o2)`”的取值是（ ）。  
A. true              B. false              C. !=              D. =
4. 接上题，下边哪一项是对 `o1` 的拷贝构造（ ）。  
A. `o2`              B. `new MyClass(o1)`              C. `o1.clone()`              D. 都不是
5. 接上题，下边哪一项是对 `o1` 的克隆（ ）。  
A. `o2`              B. `new MyClass(o1)`              C. `o1.clone()`              D. 都不是
6. 

```
class Q{
    static{
        System.out.println("static init");
    }
}
public class Q1{
    { System.out.println("init");}
    public static void main(String[] args) {
        Q rQ;
        new Q1();
    }
}
```

 执行该程序，第一行的输出结果是（ ）。  
A. `Q1`              B. `static`              C. `init`              D. `static init`

7. 当 int 类型变量 i=3、j=2 后，语句 System.out.println(i+"\*"+j+"="+i\*j)的输出结果是：  
A. 3\*2=3\*2      B. 3\*2=6      C. i\*j=i\*j      D. 以上都不对
8. 关于包声明语句和包移入语句在源程序中出现的先后顺序，正确的说法是：  
A. 包声明语句在包移入语句之后      B. 包声明语句在包移入语句之前  
C. 任意顺序都可以      D. 以上说法都不对
9. 表达式(null instanceof Object)的取值为：  
A. true      B. false      C. null      D. 不能肯定
10. class Test {  
    public static void main(String[] args) {  
        int w = 0;  
        outer:  
        for(int x = 0; x < 5; x++){  
            for(int y = 0; y < 100; y++){  
                if (x == 3)  
                    continue outer;  
                w++;  
            }  
        }  
        System.out.println("w=" + w);  
    }  
}以上 Java 代码运行的结果是 ( )  
A. w=100      B. w=200      C. w=300      D. w=400
11. 首创 Android 的公司是 ( )  
A. Microsoft 公司      B. Sun 公司      C. IBM 公司      D. Google 公司
12. 若要在自定义的类中改写 Object 类中的 toString()方法，则形参的类型可以是 ( )。  
A. Object      B. 不能有形参      C. String      D. System
13. 当定义某类型 T 的一个对象引用 ref 之后，下面哪种说法不正确：  
A. ref 可指向属于类型 T 的任一个对象；      B. 类型 T 或其任一子类的对象都可赋予 ref；  
C. ref 指向一个由类型 T 创建的对象；      D. 类型 T 可以是一个接口类型。
14. class SuperClass {  
    int x;  
    void method0() {method1();}  
    private void method1() {x=11;}  
}  
public class SubClass extends SuperClass {  
    int x;  
    private void method1() {      x=2; }

```

public static void main(String[] args) {
    SubClass rSub=new SubClass();
    SuperClass rSuper=rSub;
    rSub.method0();
    System.out.println(rSuper.x+","+rSub.x); } }

```

对于以上 Java 代码，运行主类得到的输出结果是（ ）

- A. 0, 2                      B. 11, 2                      C. 11,0                      D. 2,11

```

15. public class Test {
    public static void main(String[] args) {
        static double x=0;
        System.out.println(x+2);
    }
}

```

编译并运行该主类得到的结果是（ ）

- A. 2      B. 2.0      C. 0      D. 该程序有错误，无法正确编译

## 二、是非题（正确的标记为√，错误的标记为×，每小题 2 分，共 30 分）

1. Java 中一个类只能实现一个接口。
2. 方法在被改写时，其 throws 子句中的异常类型和数目都不能改变。
3. final 变量必须在声明的同时进行初始化。
4. super(1,2); 是对超类方法的调用。
5. 如果 A 是 B 的子类，则 A[] 是 B[] 的子类型。
6. 私有方法不能被改写。
7. 静态方法不能被改写。
8. 用 Runnable 接口实现类所创建的对象是线程。
9. 静态方法可以被隐藏。
10. 构造器中可以调用本类中的其他构造器。
11. 抽象类不能派生子类。
12. 静态方法中不能出现 this 引用。
13. public 方法在被改写时，访问控制修饰符不能改变。
14. 定义方法时，参数的数目必须确定下来。
15. final 类的所有方法隐含地成为 final。

## 三、程序设计（第 1、2 题每题 15 分，第 3 题 10 分，共 40 分）

1. 设计一个 Person 类（非主类），仅有两个域 name 和 age 且是非静态和私有的（分别为 String 类型和 int 类型），仅有一个构造器，仅有一个方法且是对 Object 类中 toString 方法的改写，以一个字符串形式返回这两个非静态域的取值。再设计一个 Student 类（非主类），是 Person 类的子类，其中仅定义一个构造器，仅定义一个域 dept 且是非静态和私有的(用来表示系，且为 String 类型)，仅定义一个方法且是对 Object 类中 toString 方法的改写，以一个字符串形式返回所有非静态域的取值。最后设计一个 Test 类，从而使这三个类构成一个完整的 Application，Test 类中

仅有一个成员即 `main` 方法，其中创建一个 `name`、`age` 取值分别为 `LiPing` 和 `18` 的 `Person` 对象，并以一个字符串形式输出所有非静态域的取值，再创建一个 `name`、`age`、`dep` 取值分别为 `Mary`、`20`、`Computer` 的 `Student` 对象，并以一个字符串形式输出所有非静态域的取值。

2. 设计并实现一个类 `Image`，这是一个基类，包括两个基本属性，分别是图像的宽度 `width` 和高度 `height`（都为 `int` 类型）。`Image` 类有三个子类，分别是二值图像类 `BinaryImage`、灰度图像类 `GreyImage` 和彩色图像类 `ColorImage`，每个类都定义了方法 `int calSize()`，用来计算存储相应图像所需要的空间（单位 `bit`）。为每个类仅提供一个构造器，且带参数，利用它给各类的实例变量赋值。进一步的要求是，`BinaryImage`、`GreyImage` 和 `ColorImage` 类都实现了接口 `I`，接口 `I` 中对 `calSize()` 方法进行了声明。使 `Image` 类成为主类，在主方法中分别创建 `BinaryImage`、`GreyImage` 和 `ColorImage` 类的对象（属性值任意），调用 `calSize()` 分别得到这三个图像所需要的存储空间并输出。综合考虑后，给出完整 `application` 的实现。（提示：对于每种类型的图像，`width` 与 `height` 的乘积为该图像的像素总数，`BinaryImage` 每个像素需要 `1bit` 的存储空间，`GreyImage` 每个像素需要 `1` 个字节的存储空间，`ColorImage` 每个像素需要 `3` 个字节的存储空间）。

3. 利用 UML 类图表达第 2 题的设计。