

笔试题，每题 2 分，最少答对 8 题。

1. `Integer n1 = new Integer(1); Integer n2 = new Integer(1); n1==n2` 结果是多少？

答案：结果是 `false`，首先，`Integer` 是引用类型，`n1` 与 `n2` 的引用地址不同，使用 `==` 判断结果是 `false`。如使用 `equals` 判断，结果为 `true`。

2. 以下代码两次打印的值是多少？

```
public class TestValidate {
    public static int staticVar = 0;

    public TestValidate() {
        this.staticVar++;
    }

    public static void main(String[] args) {
        TestValidate t1 = new TestValidate();
        System.out.println(t1.staticVar);
        TestValidate t2 = new TestValidate();
        System.out.println(t2.staticVar);
    }
}
```

答案：

先打印 1，后打印 2。

虽然创建了两个对象，但因为 `staticVar` 是静态的，在类加载后就被创建了。

3. 下面的代码有什么不妥之处？

```
if(person.getName().equals("zxx")) {}
```

答案：

如果 `person` 是一个对象，如果对象不存在，是报空指针异常。一般这样使用：

```
if(("zxx").equals(person.getName())) {}
```

4. 以下代码打印的值是什么？为什么？

```
public static void main(String[] args) {
    Person p1 = new Person();
    p1.setName("123");
    Person p2 = p1;
    p2.setName("456");
    System.out.println(p1.getName());
    System.out.println(p2.getName());
}
```

答案：两次都为 456，`Person p2 = p1`，只是把 `p1` 的引用给了 `p2`，引用指向还是一样的。

5. 以下代码打印值是什么？为什么？

```
public class TestValidate {  
    public static final int num1 = 0;  
    public static final String str1 = "a";  
    public static final Person person = new Person("zhangsan", 12);  
  
    public static void main(String[] args) {  
        num1++;  
        System.out.println(num1);  
        str1 = "b";  
        System.out.println(str1);  
        person.setName("lisi");  
        System.out.println(person.getName());  
    }  
}
```

答案：首先编译都过不了。int 和 String 被 final 修饰，表明初始化后的值是不可改的。

如果单独运行

```
person.setName("lisi");  
System.out.println(person.getName());
```

打印结果为 lisi，引用不变，但引用指向的值可以改变。

6. 以下代码打印结果是多少？

```
public static void main(String[] args) {  
    TestValidate t = new TestValidate();  
    int b = t.get();  
    System.out.println(b);  
}  
  
public int get() {  
    try {  
        return 1;  
    }  
}
```

```
        } finally {  
            return 2;  
        }  
    }  
}
```

答案：返回 2，finally 在 return 前执行。

7. 以下代码有什么问题吗？

```
public class TestCode {  
    @Test  
    public void test1() {  
        ThreadTest t1 = new ThreadTest();  
        t1.run();  
    }  
}
```

```
class ThreadTest extends Thread {  
    @Override  
    public void run() {  
    }  
}
```

答案：

直接调用 run 方法，虽然也可以执行 run 里的内容，但是不会启动线程。要启动使用，要调用 start 方法。

8. 以下代码的运行结果是多少？

```
public final int i;  
  
public static void main(String[] args) {  
    System.out.println(i);  
}
```

答案：

编译都不能通过。如果使用 final 对 int 进行修饰，那么 int 就不会存在默认值 0，必须对其进行赋值。

9. 以下代码打印出什么?

```
var val;  
  setTimeout(function () {  
    val = 11;  
  }, 100)  
console.log(val)
```

答案:

Undefined, 因为 js 默认是异步执行的。

10. 写出下面三个表的 mysql 创建语句。

学生表 student(sid,name) sid-int name-varchar cid 为主键

课程表 course(cid,name) cid-int name-varchar sid 为主键

学生课程表 student_course(sid,cid,score) sid—int,cid—int , score—int

答案:

```
CREATE TABLE student(sid INT PRIMARY KEY,`name` VARCHAR(10));
```

```
CREATE TABLE course(cid INT PRIMARY KEY,`name` VARCHAR(10));
```

```
CREATE TABLE student_course(sid INT,cid INT, score INT);
```

11. 以第 10 题的表为基础, 写出查询出总成绩大于 200 分的学生及总成绩的 mysql 语句。

答案:

```
SELECT t1.`sid`,t1.`name`,SUM(t2.`score`) FROM `student` t1 LEFT JOIN `student_course`  
t2 ON t1.`sid` = t2.`sid`
```

```
WHERE 1=1 GROUP BY t1.`sid` HAVING SUM(t2.`score`) >200
```

12. 以第 10 题的表为基础, 要得到总成绩第二的学生信息及总成绩, 请写出相应的 mysql 语句。

答案:

```
SELECT t1.`sid`,t1.`name`,SUM(t2.`score`) AS allscore FROM `student` t1 LEFT JOIN  
`student_course` t2 ON t1.`sid` = t2.`sid`
```

```
WHERE 1=1 GROUP BY t1.`sid` ORDER BY allscore DESC LIMIT 1,1
```