

Chp9 Object类 包装类 内部类

Key Point

- Object 类
- finalize 方法
- toString 方法
- equals 方法
- getClass 方法
- 包装类
- 三种类型(原始类型、包装类、字符串)与6 种转换
- 内部类的基本概念
- String 类

当垃圾收集确定不再有对该对象的引用时，垃圾收集器在对象上调用该对象。

练习

返回对象的字符串表示形式

1. (Object 类) 在Object 类中，定义的finalize 方法在_____时调用，toString() 方法返回值表示_____，equals 方法的作用为_____，getClass 方法作用为 获得Object运行时的类 判断传入的对象是否相等

2. (toString 方法，equals 方法) 写出Object 类中toString 方法的签名，以及equals 方法的签名 `public String toString(); public boolean equals(Object obj);`

3. (toString 方法) 写出下列代码的运行结果

```
class Student{
    private int age;
    private String name;
    public Student() {}
    public Student(String name, int age){
        this.name = name;
        this.age = age;
    }
    public String toString(){
        return name + " " + age;
    }
}
```

```
public class TestStudent{
    public static void main(String args[]){
        Student stu1 = new Student();
        Student stu2 = new Student("Tom", 18);
        System.out.println(stu1);
        System.out.println(stu2);
    }
}
```

// 默认调用stu1.toString 方法

null o
Tom 18












4. (equals) 写出下面程序运行的结果

```
public class TestEquals{
    public static void main(String args[]){
        String str1 = new String( "Hello" );
        String str2 = new String( "Hello" );
        System.out.println(str1.equals(str2)); // 比较内容是否相等      true
        System.out.println(str1 == str2); // 比较内存地址是否相等      false
    }
}
```

5. (getClass) 写出下面程序运行的结果

```
class Animal{}
class Dog extends Animal{}
public class TestGetClass{
    public static void main(String args[]){
        Animal a1 = new Dog();
        Animal a2 = new Animal();
        System.out.println(a1 instanceof Animal); true
        System.out.println(a1.getClass() == a2.getClass()); // getClass 获得本来的类型 false
    }
}
```

6. (包装类, 类型转换) 填空

要把一个int 类型的变量转换为Integer 类型, 需要调用  类的  方法; 要把Integer 类型转为int 类型, 需要调用  类的  方法; 要把String 类型转为Integer 类型, 需要调用  类的  方法; 要把Integer 转为String 类型, 需要调用  类的  方法; 要把int 类型转化为String 类型, 需要 ; 要把String 类型转化为int 类型, 需要调用  类的  方法。

7. (内部类) Java 中的内部类包括

- | | |
|----------|------------------------|
| A. 成员内部类 | 内部类作为外部类的成员 |
| B. 静态内部类 | 内部类作为外部类的静态成员 |
| C. 局部内部类 | 内部类定义在外部类的方法中 |
| D. 匿名内部类 | 外部类的方法中 实现接口/抽象类的匿名内部类 |
| E. 公开内部类 | |
| F. 抽象内部类 | |


8. (String) 写出下面代码输出结果


```
public class TestString{
    public static void main(String args[]){
        String str1 = "Hello" ;
        String str2 = "Hello" ;
        System.out.println(str1 == str2);      true   内存地址一样
        System.out.println(str1.equals(str2)); true   字符序列一样
        str1 = new String( "Hello" );
    }
}
```

```


        str2 = new String( "Hello" );
        System.out.println(str1 == str2);           false   new 出来的内存地址不一样
        System.out.println(str1.equals(str2));       true     字符序列一样
    }
}

```

9. 为Worker 类增加equals 和toString 方法。 

10. （包装类）修改第7 章自动分配id 的Account 类，把id 写成Long 包装类类型的。 

11. 从命令行上读入一个字符串，用两种不同的方法，把该字符串转换为一个int 类型

方法一：把String 直接转换为int 

方法二：把String 转换为Integer，再把Integer 转换为int 类型

12. （toString，字符串加法）*有下面代码

```

class Student{
    private int age;
    private String name;
    public Student() {}
    public Student(String name, int age){
        this.name = name;
        this.age = age;
    }
    public String toString(){
        return name + " " + age;
    }
}

public class TestStudent{
    public static void main(String args[]){
        Student stu1 = new Student( "tom", 18);
        System.out.println(/*1*/);
    }
}

```

问：在/*1*/位置，填入什么代码能编译通过？

- A. stu1 + " " + 100
- B. 100 + " " + stu1
- C. " " + 100 + stu1

D. stu1 + 100 + " " // 只有和字符串进行运算时，才调用toString方法

13. （Object 类）*有下面代码

```

interface IA{
    void ma();
}

```

```

}
class MyClass implements IA{
    public void ma() {}
    public String toString() {
        return "MyClass toString() ";
    }
}
}
public class TestMyClass{
    public static void main(String args[]) {
        IA ia = new MyClass();
        System.out.println(ia);
    }
}
}

```

选择正确答案:

- A. 编译不通过, 因为IA 接口中没有定义toString 方法
- B. 编译通过, 输出: "IA@地址"
- C. 编译通过, 输出: "MyClass toString() "

14. (匿名内部类, 局部内部类) *写出下面代码执行的结果

```

interface IA{
    void ma();
}
}
class MyClass {
    public static void method(IA ia){
        System.out.println( "in method" );
        ia.ma();
    }
}
}
public class TestInnerClass{
    public static void main(String args[]) {
        MyClass.method(new IA() {
            public void ma() {
                System.out.println( "ma in anonymous inner class" );
            }
        });
        class MyMaClass implements IA{
            public void ma() {
                System.out.println( "ma in local inner class" );
            }
        }
        MyClass.method(new MyMaClass());
    }
}
}

```

15. (局部内部类) *有下面代码

```

class OuterClass{

```

```

private int value1 = 100;
private static int value2 = 200;
public void method(int value3) {
    final int value4 = 400;
    class InnerClass{
        public void print() {
            //1
        }
    }
}

```

问：下面哪些代码放在//1 处能够编译通过？

- A. System.out.println(value1);
- B. System.out.println(value2);
- C. System.out.println(value3);
- D. System.out.println(value4);

16. 已知接口Light 定义如下：

```

interface Light{
    void shine();
}

```

//定义Lamp 类：

```

class Lamp{
    public void on(Light light){
        light.shine();
    }
}

```

//写一个类TestLamp，部分代码如下：

```

public class TestLamp{
    public static void main(String args[]){
        /2Lamp lamp = new Lamp();
        //1
        /
    }
}

```



把TestLamp 类补充完整，要求：

- 1) 在//1 处使用局部内部类技术，调用lamp 的on 方法要求输出“shine in red”
- 2) 在//2 处使用匿名内部类技术，调用 lamp 的 on 方法要求输出 “shine in yellow”