

## 应用知识点：

- 内部类
- Object类
- 包装类
- String类

## 习题：

1. (内部类)Java 中内部类包括()
  - A. 成员内部类
  - B. 静态内部类
  - C. 局部内部类
  - D. 匿名内部类
  - E. 公开内部类
  - F. 抽象内部类
2. (内部类)关于 Java 中的内部说法正确的是()
  - A. 成员内部类中可以访问外部类的所有属性或是方法
  - B. 静态内部类中可以访问外部类的所有的属性或是方法
  - C. 局部内部类只能在当前类中创建对象
  - D. 局部内部类可以访问所在方法中的常亮
3. (内部类)仔细阅读以下代码，下面哪些选项放在 //1 处可以编译通过()

```
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  
            }  
        }  
    }  
}
```

A. System.out.println(value1);

B. System.out.println(value2);

C. System.out.println(value3);

D. System.out.println(value4);

4. (内部类)编程：已知 Light 接口如下，根据下面要求完成代码。

```
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[]) {
        Lamp lamp = new Lamp();
        // 1
        // 2
    }
}
```

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”

5. (Object 类)仔细阅读以下代码，写出程序运行的结果。

```
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);
    }
}
```

6. (Object 类)仔细阅读以下代码，写出程序运行的结果；并简述 == 和 equals 的区别。

```
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));  
        System.out.println(str1 == str2);  
    }  
}
```

7. (Object 类)仔细阅读以下代码，写出程序运行的结果。

```
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);  
        System.out.println(a1.getClass() == a2.getClass());  
    }  
}
```

8. (Object 类)仔细阅读以下代码，//1 处填入哪些代码可以编译通过()

```
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*/);  
    }  
}
```

A. stu1 + " " + 100

B. 100 + " " + stu1

C. " " + 100 + stu1

D. stu1 + 100 + " "

9. (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\_\_\_\_\_。

10. (Object 类)编程:定义一个用户类(User 类),属性有用户名、用户密码 password、电话(tel),  
要求如下:

(1) 对类进行封装,提供 get/set 方法;同时提供无参数、有参数的构造方法

(2) 覆盖 toString 方法,要求格式为:

username:一如既往,password:123456,tel:13051800688

(3) 覆盖 equals 方法,要求:只要用户名相同则为相同对象

(4) 写一个测试类,利用键盘分别输入两个用户信息并存储 Use 对象中,判断两个对象是否相同。

11. (包装类)以下包装类的类名正确的是()

- |          |           |         |            |
|----------|-----------|---------|------------|
| A. Byte  | B. Short  | C. Int  | D. Long    |
| E. Float | F. Double | G. Char | H. Boolean |

12. (包装类)下面关于数据类型之间的转换正确的是()

- A. Integer i = 5;
- B. int a = new Integer(12);
- C. Integer i2 = 20 + "";
- D. String s = true + " ";

13. (包装类)仔细阅读以下代码,写出程序运行打印输出的结果;并解释每个结果的原因。

```
public class TestMain{
    public static void main(String[] args) {

        int a1 = new Integer(128);
        int a2 = new Integer(128);

        System.out.println(a1==a2);

        Integer i1 = new Integer(120);
        Integer i2 = new Integer(120);
        System.out.println(i1==i2);

        Integer i3 = new Integer(128);
        Integer i4 = new Integer(128);
        System.out.println(i3==i4);
    }
}
```

14. (String 类)仔细阅读以下代码段：

```
String s = "hello";
```

```
String t="hello";
```

```
char[] c = {'h','e','l','l','o'};
```

下列选项输出结果为 false 的语句是()

- A. System.out.println( s.euqals( t ) );
- B. System.out.println( t.equals( c ) );
- C. System.out.println( s==t );
- D. System.out.println( t.equals( new String(c) ) );
- E. System.out.println( t.equals( new String("hello")) );

15. (String 类)关于 java.lang.String 类，以下描述正确的一项是()

- A. String 类是 final 类故不可以继承。
- B. String 类是 final 类故可以继承。
- C. String 类不是 final 类故不可以继承。
- D. String 类不是 final 类故可以继承。

16. (String 类)应用程序的 main 方法中有以下语句，则输出的结果是

```
public class TestMain{  
    public static void main(String[] args) {  
        String s1 = "Abc",s2 = "abc";  
        boolean b1 = s1.equals( s2 );  
        System.out.println( b1 );  
    }  
}
```

- A. false
- B. true
- C. 1
- D. 0

17. (String 类)已知 String 对象 s=" abcdefg" ，则 s.substring(2,5)的返回值为 ( )

- A . " bcde"
- B . " cde"
- C . " cdef"
- D . " def"

18. (String 类)已知 s 为一个 String 对象，s=" abcdefg" ，则 s.charAt(1)的返回值为 ( )。

- A . a
- B . b
- C . f
- D . g

19. (String 类)仔细阅读以下代码，关于程序描述正确的是()

```
public class TestMain{
    public static void main(String[] args) {
        String str = new String("abcde");
        str.replace('c','x');
        System.out.println( str );
    }
}
```

- A、编译错误，String 类没有 replace 方法。
- B、运行后输出的结果是 abcde
- C、运行后输出的结果是 abxde
- D、编译成功，运行时抛出异常,提示 String 类的内容是不可变的。

20. (String 类)仔细阅读以下代码，执行后的结果是()

```
public class TestMain{
    public static void main(String[] args) {
        String s = new String("abcdefg");
        for(int i=1; i<s.length(); i+=2){
            System.out.println(s.charAt(i));
        }
    }
}
```

- A. aceg
- B. abc
- C. bdf
- D. abcdefg

21. (String 类)请解释 Sting 和 StingBuilder 操作字符串时的区别。

22. (String 类)编程：任意输入一个字符串，统计字符串中字母的个数。

23. (String 类)编程：给定一个邮箱“zhangmj@zparkhr.com”，根据下面要求完成程序代码：

(1) 获取邮箱中的用户名：zhangmj

(2) 验证邮箱“zhangmj@zparkhr.com”是否是一个合法的邮箱格式。

提示：① 邮箱必须包含“@”和“.”

② 最后一个“.”的位置必须大于“@”的位置。

24. (String 类)编程：编写一个方法，功能是实现判断某字符串数组中是否有重复元素。

请自行拟定合适的方法名称，实现方法自行选择，可实现要求即可；

例如：public boolean 方法名 ( String[] strArray ) { .... }

25. (String 类)编程：在给定的字符串 "ABCDEFGHijklmn1234567" 中获取随机的 4 个字符，并使用 StringBuilder 拼接成字符串。（随机获取到的 4 个字符中可以出现重复字符）

提示：创建随机数对象 java.util.Random。

```
java.util.Random random = new java.util.Random();  
  
random.nextInt(100); //可以获取到 0~99 中任意一个随机数
```

26. (String 类)编程：将给定的字符串进行反转，并打印输出反转之后的字符串。

例如：给定的字符串为 "hello"

反转的字符串为 "olleh"

27. (String 类)编程：给定一个由数字组成的字符串，如："1239586838923173478943890234092"；统计出每个数字出现的次数。