Day12 - 三个修饰符

百知教育 Baizhi Education

应用知识点：

- **abstract**
- **static**
- **final**
- **修饰的综合应用**

习题：

1. 关于三个修饰符描述不正确的是()

   A. static 可以修饰所有类型的变量

   B. static 可以修饰方法

   C. final 可以修饰类

   D. abstract 可以修饰类、方法、属性

2. 下面关于 static 描述不正确的是()

   A. static 修饰的属性称为静态变量，被所有的对象共享

   B. static 修饰的方法，可以直接用 类名.方法名(参数)调用

   C. static 修饰的方法中可以直接访问本类的其他方法

   D. 父类中定义了 static 修饰的方法，子类中不能再定义同名的 static 方法

3. 关于 final 描述错误的是()

   A. final 修饰的类是作用范围内常量，只允许一次赋值

   B. final 修饰的方法不能被继承

   C. final 修饰的方法不允许被覆盖

   D. final 修饰的类没有子类

4. 关于 abstract，以下选项正确的是：

A. abstract 类中可以没有 abstract 方法

B. abstract 类的子类也可以是 abstract 类

C. abstract 类不能创建对象，但可以声明引用

D. abstract 方法不能有方法体

5. 仔细阅读以下程序，写出程序输出的结果。

```
class MyClass {
    static int a;
    int b;
}
public class TestMain {
    public static void main(String args[]) {
        MyClass mc1 = new MyClass();
        MyClass mc2 = new MyClass();
        mc1.a = 100;
        mc1.b = 200;
        mc2.a = 300;
        mc2.b = 400;
        System.out.println(mc1.a);
        System.out.println(mc1.b);
        System.out.println(mc2.a);
        System.out.println(mc2.b);
    }
}
```

6. 仔细阅读以下程序，选出哪些代码会导致程序出错()

```
class MyClass {
    int a;
    static int b;
    void fa() {}
    static void fb() {}
    public void m1() {
        System.out.println(a); // 1
        System.out.println(b); // 2
        fa(); // 3
        fb(); // 4
    }
    public static void m2() {
        System.out.println(a); // 5
        System.out.println(b); // 6
        fa(); // 7
        fb(); // 8
    }
}
```

A. //1　　B. //2　　C. //3　　D. //4

E. //5　　F. //6　　G. //7　　H. //8

7. 仔细阅读以下程序，写出程序运行输出的结果。

```
class MyClass {
    static int i = 10;
    static {
        i = 20;
        System.out.println("In Static");
    }
    public MyClass() {
        System.out.println("MyClass()");
    }
    public MyClass(int i) {
        System.out.println("MyClass(int)");
        this.i = i;
    }
}
public class TestMain {
    public static void main(String args[]) {
        MyClass mc1 = new MyClass();
        System.out.println(mc1.i);
        MyClass mc2 = new MyClass(10);
        System.out.println(mc2.i);
    }
}
```

8. 仔细阅读以下代码，关于代码描述正确的是()

```
1)      class MyClass{
2)          final int value;
3)          public MyClass(){}
4)          public MyClass(int value){
5)              this.value = value;
6)          }
7)      }
8)      public class TestMain{
9)          public static void main(String args[]){
10)             MyClass mc = new MyClass(10);
11)             System.out.println(mc.value);
12)         }
13)     }
```

A. 编译通过，输出 10

B. 编译不通过，把第 2 行改为 final int value = 10;

C. 编译不通过，把第 3 行改为 public MyClass(){ value = 10; }

D. 以上描述都不正确

9. 仔细阅读以下代码，关于以下程序描述正确的是()

```
class MyValue {
    int value;
}
public class TestFinal {
    public static void main(String args[]) {
        final MyValue mv = new MyValue();
        mv.value = 100;
        // 1
        System.out.println(mv.value);
    }
}
```

A. 编译不通过

B. 编译通过；在//1 处加上：mv.value = 200; 则编译不通过

C. 编译通过。如果在//1 处加上：mv = new MyValue(); 则编译不通过

D. 以上描述都不正确

10. 仔细阅读以下代码，选出正确选项()

```
class MyClass {
    public void printValue(final int value) {
        System.out.println(value);
    }
    public void changeValue(int value) {
        value = value * 2;
        System.out.println(value);
    }
}

public class TestMain {
    public static void main(String args[]) {
        MyClass mc = new MyClass();
        int value = 5;
        final int fvalue = 10;
        mc.printValue(value); // 1
        mc.printValue(fvalue); // 2
        mc.changeValue(value); // 3
        mc.changeValue(fvalue);// 4
    }
}
```

A. 编译通过　　B. //1 出错　　C. //2 出错

D. //3 出错　　E. //4 出错

11. 仔细阅读以下代码，代码中哪些内容是错误的()

```
abstract class MyAbstractClass{
    public abstract void m1(); //1
    abstract protected void m2(){} //2
}
class MySubClass extends MyAbstractClass{
    void m1(){} //3
    protected void m2(){} //4
}
```

A. //1　　B. //2　　C. //3　　D. //4

12. 仔细阅读以下代码，程序是否能编译通过？如果可以，输出运行结果；如果不可以，应该怎样修改？

```
class Super {
    public final void m1() {
        System.out.println("m1() in Super");
    }
    public void m1(int i) {
        System.out.println("m1(int) in Super");
    }
}
class Sub extends Super {
    public void m1(int i) {
        System.out.println("m1(int) in Sub");
    }
    public void m1(double d) {
        System.out.println("m1(double) in Sub");
    }
}
public class TestMain {
    public static void main(String args[]) {
        Sub s = new Sub();
        s.m1();
        s.m1(10);
        s.m1(1.5);
    }
}
```

13. 仔细阅读以下程序，写出程序运行输出的结果。

```
class Super {
    public static void m1() {
        System.out.println("m1 in Super");
    }
    public void m2() {
        System.out.println("m2 in Super");
    }
}
class Sub extends Super {
    public static void m1() {
        System.out.println("m1 in Sub");
    }
    public void m2() {
        System.out.println("m2 in Sub");
    }
}
public class TestMain {
    public static void main(String args[]) {
        Super sup = new Sub();
        sup.m1();
        sup.m2();
        Sub sub = (Sub) sup;
        sub.m1();
        sub.m2();
    }
}
```

14. 下面关于方法声明正确的是()

A. abstract final void m()

B. public void final m()

C. static abstract void m()

D. private abstract void m()

E. public static final void m()

15. 判断以下描述是否正确，正确填 T；否则填 F。

(1) abstract 可以修饰类、方法、属性（ ）

(2) 抽象类中不一定有构造方法（ ）

(3) 抽象类只能声明引用，不允许单独 new 对象（ ）

(4) 所有类中都可以定义抽象方法（ ）

(5) 子类继承抽象类必须实现抽象类中所有抽象方法（ ）

(6) static 可以修饰属性、方法、代码块（ ）

(7) 静态属性只能用 类名.静态属性名 访问 （ ）

(8) 静态方法可以被本类中的其他方法直接访问（ ）

(9) 静态代码在创建对象时完成静态属性的初始化工作（ ）

(10)final 只能修饰属性、方法、类（ ）

(11)final 修饰的方法不能被继承（ ）

(12)final 修饰的类不能被继承，即没有子类（ ）

16. 把三大特性中的 Shape 类改为抽象类，并把其中的求周长和求面积的方法改为抽象方法。

17. 设计一个类 MyClass，为 MyClass 增加一个 count 属性，用来统计总共创建了多少个对象。