



福昕PDF编辑器

• 永久 • 轻巧 • 自由

升级会员

批量购买



永久使用

无限制使用次数



极速轻巧

超低资源占用，告别卡顿慢



自由编辑

享受Word一样的编辑自由



扫一扫，关注公众号



福昕PDF编辑器

• 永久 • 轻巧 • 自由

升级会员

批量购买



永久使用

无限制使用次数



极速轻巧

超低资源占用，告别卡顿慢



自由编辑

享受Word一样的编辑自由



扫一扫，关注公众号

Day06

Key Point

- 面向对象与面向过程的比较
- 类与对象的关系
- 成员变量与局部变量的区别
- 构造函数
- this关键字
- 代码块
 - 局部代码块
 - 构造代码块
- 对象在内存中的存储
- 匿名对象

一、 选择题

1. 下列各项中关于面向对象的说法正确的是：

- A. 面向对象实际上是基于面向过程的
- ☒ B. 面向对象适用于一些相对复杂的事务
- C. 面向对象需要关心流程中的每一步
- D. 面向对象的特征主要有三个：高可用、高并发、跨平台
- E. 面向对象和面向过程是相对于不同的场景而言的，但是面向对象一定是由于面向过程的

2. 对于代码：

```
class ClassA{
    public void method(int value){
        System.out.println(value);
    }
    public void method(){
        System.out.println(value);
    }
    int value;
}
class TestClassA{
    public static void main(String args[]){
        ClassA classA = new ClassA();
        classA.value = 10;
        classA.method();
        classA.method(20);
    }
}
```

请选择正确结果：

- A. 编译不通过
- B. 输出10 10
- C. 输出 10 20
- D. 输出0 20

3. 下列说法正确的有：

- A. 类中的构造函数不可省略
- B. 构造函数必须与类同名，但是普通方法不能与类同名
- C. 构造函数在一个对象被new时调用
- D. 一个类中只能定义一个构造函数
- E. 构造函数可以通过实例来调用

4. 有以下代码:

```
class ClassA{
    int value = 10;
}
public class TestReturnRef{
    public static void main(String args[]){
        ClassA ca = new ClassA();
        ca = getObject();
        ca = getObject();
        ca = getObject();
        System.out.println(ca.value);
    }
    public static ClassA getObject(){
        ClassA newObject = new ClassA();
        newObject.value += 10;return newObject;
    }
}
```

编译运行TestReturnRef 程序，结果为：

- A. 编译出错
- B. 输出10
- C. 输出20
- D. 输出40

5. 有以下代码

```
class ClassA{
    public ClassA(){
        System.out.println("ClassA()");
    }
}
class ClassB{
    public ClassB(){
        System.out.println("ClassB()");
    }
}
```

```

}
class ClassC{
ClassA a = new ClassA();
ClassB b;
public ClassC(){
System.out.println("ClassC()");
    b = new ClassB();
}
}
public class TestConstructor{
public static void main(String args[]){
    ClassC cc = new ClassC();
}
}

```

请选择正确答案：

- A. 编译不通过
- B. 输出ClassA() ClassB() ClassC()
- C. 输出 ClassA() ClassC() ClassB()**
- D. 输出 ClassC() ClassB() ClassA()

6. 有以下代码

```

class ClassA{
    int value;
}
public class TestClassA{
public static void main(String args[]){
    int value = 10;
    changeInt(value);
    System.out.println(value);
    ClassA ca = new ClassA();
    ca.value = 10;
    changeObject(ca);
    System.out.println(ca.value);
}
public static void changeInt(int value){
    value++;
}
public static void changeObject(ClassA ca){
    ca.value++;
}
}

```

编译运行TestClassA 时，结果是

- A. 编译出错**
- B. 输出 10 11**
- C. 输出 10 10
- D. 输出 11 11

7. 有以下代码

```
class ClassA{
    int value;
}
public class ChangeRef{
    public static void main(String args[]){
        ClassA ca = new ClassA();
        changeValue(ca);
        System.out.println(ca.value);
        changeRef(ca);
        System.out.println(ca.value);
    }
    public static void changeValue(ClassA ca){
        ca.value = 100;
    }
    public static void changeRef(ClassA ca){
        ca = new ClassA();
        ca.value = 200;
    }
}
```

编译运行ChangeRef，结果为

- A. 编译不通过
- B. 输出100 200
- C. 输出100 100
- D. 输出0 200

二、简答题

1. 对于代码：

```
class ClassA{
    void method(){
        System.out.println("method()");
    }
    int method(int i){
        System.out.println("method(int)");
    }
    public static void main(String args[]){
        ClassA a = new ClassA();a.method();
        a.method(10);
    }
}
```

该程序是否能编译通过？如果可以，写出该程序运行结果。如果不能，请说明理由，以及如何修改。

可以

2. 有以下代码

```
class MyClass{
    int value;
}
public class TestRef{
    public static void main(String args[]){
        int a = 10;
        int b = a;
        b ++ ;
        System.out.println(a);
        MyClass mc1 = new MyClass();
        mc1.value = 10;
        MyClass mc2 = mc1;
        mc2.value ++;
        System.out.println(mc1.value);
    }
}
```

请写出编译运行后的结果. 10, 11

3. 有以下代码

```
class MyClass{
    int value;
}
public class TestMyClass{
    public static void main(String args[]){
        MyClass mc1 = new MyClass();
        MyClass mc2 = new MyClass(10);
        System.out.println(mc1.value);
        System.out.println(mc2.value);
    }
}
```

问：这个程序能否编译通过？如果可以，输出结果是什么？如果不可以，则应该如何修改？

编译不过，可以在方法中添加

4. 程序改错

public MyClass(int vlaue){
 this.value = value;

```
}
public class Student{
    public void Student(){
        void init(){
            age = 10;
            name = "limy";
        }
    }
    public Student(String name){
        this.init();
    }
}
```

```

        this.name = name;
    }
    public Student(String name, int age){
        this.init();
        this(name);
        this.age = age;
    }
    int age;
    String name;
}

```

三、编程题

1. 复数概念如下：

每个复数都有实部和虚部。例如， $3 + 5i$ ， 3 为实部， $5i$ 为虚部。其中， i 称为虚数单位，有 $i*i = -1$ 。两个复数进行加法运算，运算时实部与实部相加，虚部与虚部相加。例如：

$$(1.5 - 3i) + (2.3 + 2.4i) = (1.5+2.3) + (-3 + 2.4)i = 3.8 - 0.6i$$

两个复数进行减法运算，与加法运算类似。两个复数进行乘法运算，其过程如下：

$$(a+bi) * (c + di) = ac + adi + bci + bd(i*i) = (ac-bd) + (ad+bc)i$$

例如：

$$(3+5i) * (4+6i) = (3*4-5*6) + (3*6+4*5)i = -18 + 38i$$

写一个类 `Complex`，用来表示复数。这个复数类具有两个属性：`double real`，表示实部；`double im`，表示虚部。并为 `Complex` 类增加 `add`、`sub`、`mul` 方法，分别表示复数的加法、减法和乘法运算。其中，`add` 方法的声明如下：`public Complex add(Complex c)` //表示当前 `Complex` 对象与参数 `c` 对象相加. `public Complex add(double real)` //表示当前 `Complex` 对象与实数 `real` 相加