Chapter 05 面向对象基础

Key Point：

- ●类和对象的概念
- ●实例变量
- ●方法重载
- ●构造方法
- ●引用的概念
- ●this关键字

问题：

1. （重载，实例变量）有以下代码：

```
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

2. （方法重载，函数返回值）有以下代码

```
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。

3. （构造方法）关于构造方法，下列说法正确的是：

A. 每个类中都有至少一个构造方法

B. 一个类中可以有多个构造方法

C. 构造方法可以有返回值

D. 构造方法可以有多个参数

4. （引用）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。

5. （构造函数）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？

6. （面向对象基础）根据注释，把下面代码补充完整

```
//定义一个 Dog 类
class Dog{
    //定义一个 name 属性，该属性为 String 类型
```

```
    _____;
    //定义一个 age 属性，该属性为 int 类型
    _____;
    //定义一个 sexual 属性，该属性为 boolean 类型
    //true 表示为公，false 表示为母
    _____;
    public Dog(){
    }
    public Dog(String name, int age, boolean sexual){
        //分别根据参数，设置 Dog 类的属性
    }
    public void play(){
        System.out.println(name + “ play”);
    }
    public void play(int n){
        System.out.println(name + “ play ” + n + “ minutes”);
    }
}
public class TestDog{
    public static void main(String args[]){
        Dog d;
        //创建一个 Dog 对象，调用有参构造函数
        //名字为 joy，年龄为 2 岁，性别为母
        _____;
```

```
        //调用 Dog 对象无参的 play 方法。
        _____;
        //调用 Dog 对象有参的 play 方法，参数为 30
        _____;
    }
}
```

7. （对象创建过程）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()

8. （引用，方法参数传递）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 10

C. 输出 10 11

D. 输出 11 11

9. （构造函数，this 关键字）程序改错

```
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;
}
```

10. （面向对象基础）写一个 Worker 类，并创建多个 Worker 对象。

I. 为 Worker 类添加三个属性：

1). String 类型的 name，表示工人的姓名。

2). int 类型的 age，表示工人的年龄。

3). double 类型的 salary，表示工人的工资。

II. 为 Worker 类添加两个构造方法：

1). 公开无参构造方法。

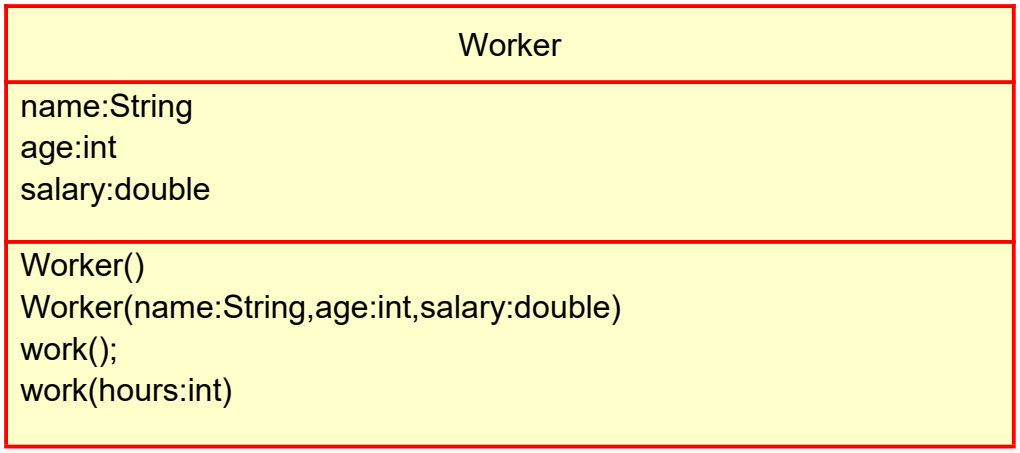
2). 接受三个参数的构造方法，三个参数分别为字符串、int 和 double 类型。

III. 为 Worker 类添加两个 work 方法：

1). 无参构造方法

2). 带整数参数构造方法，表示工人工作的时间（为多少小时）。

类图如下：



11. （面向对象基础）创建一个 Address 类，描述如下：

I. 该类有两个属性：

1). String 类型的 address，表示地址。

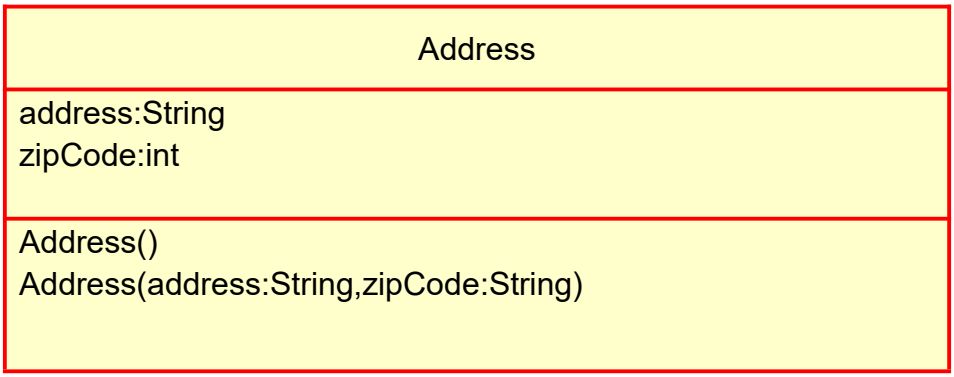
2). String 类型的 zipCode，表示邮编。

II. 该类有两个构造方法：

1). 无参构造方法。

2). 带三个参数的方法。

类图如下：



12. （面向对象基础）

为第 1 题中的 Worker 类添加一个属性 addr，该属性为 Address 类型。

要求：创建一个 Worker 对象，其姓名为"zhangsan"，年龄为 25，工资为 2500，家庭住址为“北京市

海淀区清华园 1 号”，邮政编码为 100084。