

Chapter 06 面向对象三大特性

Key Point :

- 封装/数据隐藏
- 继承的基本语法
- 访问修饰符
- 对象创建过程
- super 关键字
- 方法覆盖
- 多态的基本语法和使用
- instanceof
- 多态用在参数和返回值上

答案：

1. 输出结果为：

Super()、Sub() //第一个 Sub 对象 Super()

Sub()、Sub(int) //第二个 Sub 对象 Super(String)

Sub(String) //第三个 Sub 对象

I. 创建第一个 Sub 对象时，由于第一行既不是 this，又不是 super，因此会调用父类的无参构造函数。

从而输出：Super() Sub()

II. 创建第二个 Sub 对象时，首先调用的是 this()，也就是说，先调用本类的无参构造方法。

调用 Sub()时，根据刚刚的解释，会输出 Super()和 Sub()在调用完 this()之后，会输出 Sub(int)。

因此，输出的结果为：Super() Sub() Sub(int)

III. 创建第三个 Sub 对象时，调用的是 super(str)，因此会调用父类的带字符串参数的构造方法。

因此输出：Super(String) Sub(String)

2. 运行结果为：

m1() in Sub

m1() in Super

m2() in Super

注意：因为在子类的 m1 方法中调用了父类的 m1 方法。所以调用子类的 m1 方法时，会输出两行：

m1() in Sub

m1() in Super

3. 编译出错：

Super 类中没有定义 method(String)方法，因此无法调用。

4. 输出结果为：

m() in Super

m() in Sub

5. ABCD

注意：return null 同样可以编译通过。

6. TestMyClass1 和 TestMyClass2 都有错

```
//TestMyClass1.java
```

```
package corejava.chp6;
```

```
public class TestMyClass1{
```

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

```
        MyClass mc1 = new MyClass();
```

```
        MyClass mc2 = new MyClass(10);
```

```
        System.out.println(mc1.value); //应改 mc1.getValue()
```

```
        System.out.println(mc2.value); //应改 mc2.getValue()
```

```
    }
```

```
}
```

```
//TestMyClass2.java
```

```
public class TestMyClass2{
```

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

```
        MyClass mc1 = new MyClass();
```

```
        MyClassmc2=newMyClass(10); //此处错误，带参数的构造方法在非同包的类中不能访问
```

```
        System.out.println(mc1.value); //应改为 mc1.getValue()
```

```
        System.out.println(mc2.value); //应改 mc2.getValue()
```

```
    }
```

```
}
```

7. E

子类 and 父类不在同一个包中，无法访问父类 default 修饰的属性。

注意：D 选项也是正确的，只是会形成属性遮盖。

8. 输出结果为：

Meal()

Lunch() // 递归构造父类对象 Vegetable()

Potato() //Potato 属性

Meat() //Meat 属性 Vegetable()

Tomato() //Tomato 属性

Sandwich()//本类构造方法

9.为 Super 增加两个构造函数 Super()和 Super(String)

10.AC

A 正确，这是方法覆盖的正确形式。

B 错误，方法覆盖要求返回值类型相同

C 正确，选择 C 之后 Sub 类中就有两个 method 方法，一个是从 Super 中继承的，另一个是 Sub 类中定义的。这两个方法方法名相同，参数表不同，构成重载。

11.ABCD

父类的方法是 private 方法，无法被子类继承，因此不用考虑重载和覆盖的问题。

12.参考代码如下：

I. 省略//1 //2 //3

II. 参考如下：

```
public static Dog[] getAllDog(Animal[] as){
```

```

//统计 Dog 的个数

int sumDog = 0;

for(int i = 0 ; i<as.length ; i++){

    if(as[i] instanceof Dog){

        sumDog ++;

    }

}

//创建 Dog 数组

Dog[] dogs = new Dog[sumDog];

int dogIndex = 0;

for(int i = 0 ; i<as.length ; i++){

    if(as[i] instanceof Dog){

        dogs[dogIndex] = (Dog)as[i];

        dogIndex++;

    }

}

return dogs;

}

```

13.参考：Worker.java 、 Address.java

14.参考：Student.java

注意：

- I. 由于 getPostAddress 方法要求返回两个值：address 和 zipCode。

II. 因此应该封装一个包含两值的 PostAddress 对象作为 getPostAddress 方法的返回值。

15.参考：Shape.java。

16.参考：TestShape.java。

17、18：参考 TestEmployee.java。

重点在于 super 关键字的使用几个要点：

- I. name 和 birthMonth 是每个员工都具有的属性，因此应放在 Employee 类中（共性放在父类）。
- II. 子类无法直接访问这两个属性，但是可以利用 super()在构造方法中设置父类的属性。
- III. 判断员工是否过生日的逻辑也是所有员工都有的逻辑，应当写在 Employee 类（共性放在父类）。
- IV. 子类的 getSalary 方法应用 super.getSalary 来调用父类的 getSalary 方法。