

Chp8 接口

参考答案

1. 参考答案

```
interface IA{
    void m1();
    int a = 100;
}

class MyClass implements IA{
    //接口中方法默认都是 public 的，因此实现类必须也是 public 的
    void m1(){}
}

public class TestInterface{
    IA ia = new MyClass();
    ia.m1();
    System.out.println(IA.a);
}
```

2. 代码填空

```
interface IA{
    void m1();
    void m2();
}

abstract class MyClassA implements IA{
    public void m1(){}
}

class MyClassB extends MyClassA{
    public void m2() {}
}
```

3. 参考答案

- 1) 要实现 ma, mb, mc, md 方法
- 2) 调用 mc 方法时不需要强转，调用 ma/mb/md 方法时需要强转
- 3) 输出为 5 个 true

4. 略

5. 输出结果为 4 个 true

如果父类实现了某些接口，则子类自动就实现了这些接口。

6. ACDE

B 选项错误，因为接口也可以有属性，但是属性必须是公开静态常量。

7. 输出结果

impl1

impl2

解释：

在调用 mc 对象的 myMethod 方法时，myMethod 方法会使用 mc 对象的 ia 属性，并调用 ia 属性的 m1 方法。

在第一次调用 myMethod 方法时，mc 对象的 ia 属性指向的是一个 IAImp11 类型的对象，因此调用 m1 方法时，根据多态，会调用 IAImp11 对象中实现的 m1 方法，输出

impl1

第二次调用 myMethod 方法时，mc 对象的 ia 属性指向的是一个 IAImp12 类型的对象，调用 m1 方法时，会调用 IAImp12 中实现的 m1 方法，输出

impl2

8. 输出结果

Red Light shine in Red

Yellow Light shine in Yellow

Green Light shine in Green

参考第 7 题的解释。

9. 输出结果

TeacherA teach Java

TeacherB teach Java

解释：

本题的重点在于利用接口，把多态用在返回值类型上。getTeacher 方法返回的对象是 JavaTeacher 接口类型，在 getTeacher 方法内部创建 JavaTeacher 实现类的对象，并作为返回值返回。

特别的，对于 getTeacher 的调用者而言，不需要关心 JavaTeacher 的实现类，而只需要操作 JavaTeacher 接口，从而实现了弱耦合。

10. 参考答案

//1 处填入的代码为

```
for (int i = 0; i<as.length; i++){  
    as[i].eat();  
}
```

//2 处填入的代码为

```
for (int i = 0; i<as.length; i++){  
    if (as[i] instanceof Pet){  
        Pet p = (Pet) as[i];  
        p.play();  
    }  
}
```

```
    }  
}
```

11. 参考 TestMedia.java

12. 参考 TestProgram.java

13. 参考 TestEmployee.java

思路：让 SalariedEmployee 和 BasePlusSalesEmployee 实现 OverTime 接口，表示他们都加班。在统计总的加班费时，使用 instanceof 进行判断。

14. ACD

强制类型转换时，由于 java 中单继承的限制，因此如果强转时没有父子类的关系，则 Java 会认为这是不可转换的类型，编译无法通过。

如果转换时是接口类型，由于接口没有单继承的限制，因此 Java 认为任何类都有可能实现任何接口，因此总是能编译通过。

15. 如果实现 ServiceImpl 接口，则必须实现其所有方法；而如果继承 AbstractService 类，则只需要覆盖其中我们感兴趣的方法即可。

在实际开发中，往往既提供接口，又提供抽象类：这样，在提供接口灵活性的基础上，也能够使用抽象类来减少代码量。

16. 输出结果为

ma in IAImpI

mb in IBImpl

本题较难

在调用 ib 对象的 method 方法时，会调用其 ia 属性的 ma 方法。由于其 ia 属性指向一个 IAImpI 的对象，因此根据多态，调用的是 IAImpI 中的 ma 方法。

调用 ma 方法时，必须要传一个 IB 类型的参数。由于 IBImpl 对象本身实现了 IB 接口，因此可以把当前对象当做 IB 对象，作为参数传递给 ma 方法。因此，调用 ma 方法时，可以使用：

```
ia.ma(this);
```

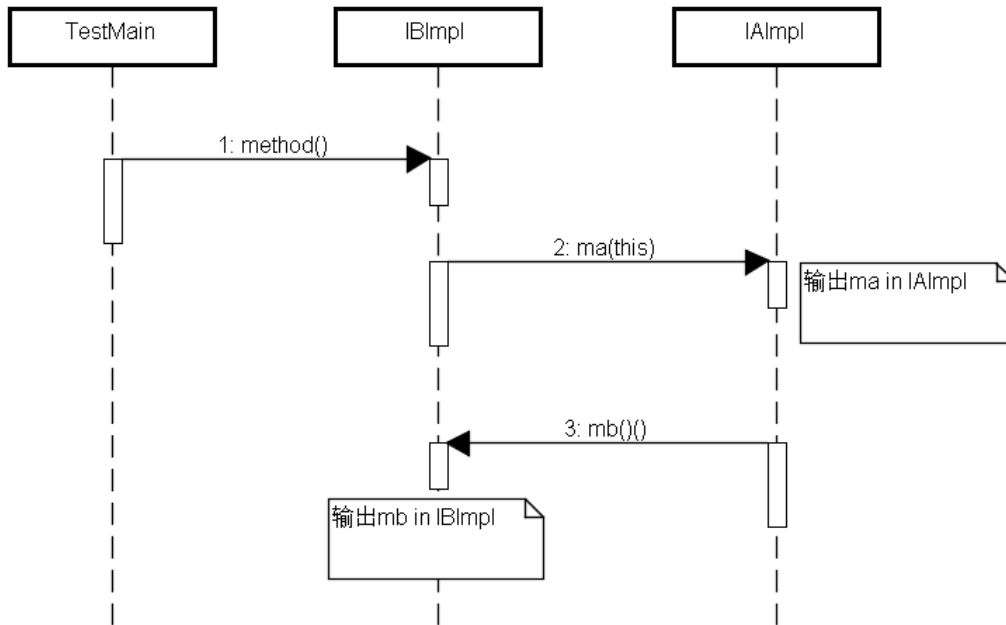
在 IAImpI 的 ma 方法中，首先输出

ma in IAImpI

之后，调用其参数 ib 的 mb 方法。ib 参数指向的是一个 IBImpl 类型的对象，因此根据多态，实际调用的是 ib 的 mb 方法，输出：

mb in IBImpl

方法调用过程示意图如下：



17. 参考 TestRole.java

注意：所有 **Weapon** 的实现类都应该有一个 **Soldier** 属性，表示使用该武器的战士是谁。在把武器分配给战士时，既要调用武器的 **setSoldier** 方法，也要调用战士的 **setWeapon** 方法，只有这样才能正确的维护这个双向关系。

18. 应当首先定义一个接口，并以此接口作为同组同学开发时的依据。

例如：

```
interface MathTool{
    public boolean isPrime(int n);
}
```

然后同组的两个开发者一个使用 **MathTool** 的方法，一个为 **MathTool** 提供实现。可以参考 **TestGoldBach.java** 程序