DIP - 依赖倒置原则

DIP - 依赖倒置原则

- 依赖倒置原则:

  高层模块不应该依赖低层模块， 两者都应该依赖**抽象。**

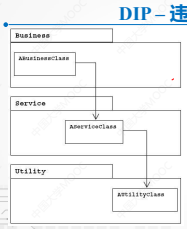
  抽象不应该依赖细节，细节应该依赖**抽象**.

DIP - 依赖倒置原则

DIP - 依赖倒置原则

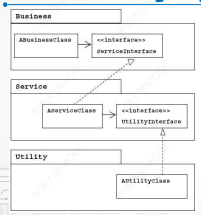
- 结构化分析与设计往往造成高层模块依赖低层模块，策略依赖细节实现。
- 一般来说，高层模块包含应用程序的商业决策和业务模型。

DIP – 违反DIP原则



- 如果 business依赖服务层中的具体services，而services依赖Utility层的具体utilities，则 business依赖utilities。
- 这意味着低层模块的变化会影响高层模块。
- 高层模块很难重用到其他系统中

DIP –与DIP原则一致



- 采用上层模块发布接口来反转依赖关系 (客户端“拥有”接口).
- 业务不再依赖具体的服务，且业务可被重用到其他系统中。

例子

```
// Dependency Inversion Principle - Bad example
class Worker {
        public void work() { // ....working }
}
class Manager {
Worker m_worker;
public void setWorker(Worker w) { m_worker=w; }
public void manage() { m_worker.work(); }
}
class SuperWorker {
public void work() { //.... working much more }
}
```

例子

```
        // Dependency Inversion Principle - Good example
interface IWorker {    public void work();    }

class Worker implements IWorker{
        public void work() {    // ....working        }
}

        class SuperWorker implements IWorker{
        public void work() {    //.... working much
more        }
}

        class Manager {
        IWorker m_worker;
        public void setWorker(IWorker w)
{    m_worker=w;    }
        public void manage() {    m_worker.work(); }
}
```

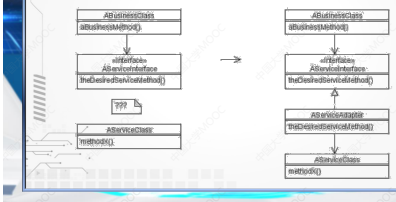
DIP – 适应现有的类

问题:

- 如果AServiceClass已经存在但是不符合所需的、ServiceInterface?

DIP – 适应现有的类

解决方案 (Adapter模式):



DIP – 取决于抽象

"幼稚"但有用的解释:

- 没有变量应该引用具体的类
- 没有类应该从具体的类中派生出来
- 没有方法应该覆盖其基类的已实现方法

DIP – 取决于抽象

DIP原则的核心是：应该只依赖抽象，即抽象类和接口

这种假设过于严格，对于不容易变化的类，并不需要使用抽象。

抽象应该用到系统的易变化部分，以及需要进行解耦的部分，譬如：层之间。

DIP – 例子: 按钮 - 灯 (1)


```
Public class Button  {
  private Lamp itsLamp;

  public  void poll()  {
    if (/* some condition */)  itsLamp.turnOn();
  }
}
```

DIP – 例子: 按钮 - 灯(2)

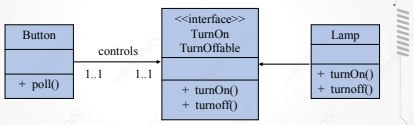
违反DIP原则

- 按钮依赖灯
- 按钮不能重用到其他系统中

解决方案

- 按钮发布一个接口。
- 使按钮依赖接口。实现该接口的不同电器均可用该按钮控制。
- 使用此接口的其他控件也可用来控制灯。

DIP – 例子: 按钮 - 灯 (1)



DIP - 总结

结构化的设计:

- 高层依赖低层
- 层与层之间不用接口分离
- 低层的改变可能会影响高层
- 低层拥有接口 - 高层适应

DIP - 总结

DIP:

- 低层和高层都依赖抽象
- 低层的变化通常不会影响高层
- 高层拥有接口 - 低层适应
- “面向对象的设计”

Exercise

- 请举一个违反DIP原则的例子并解释原因。
- 如何修改它以符合DIP原则?