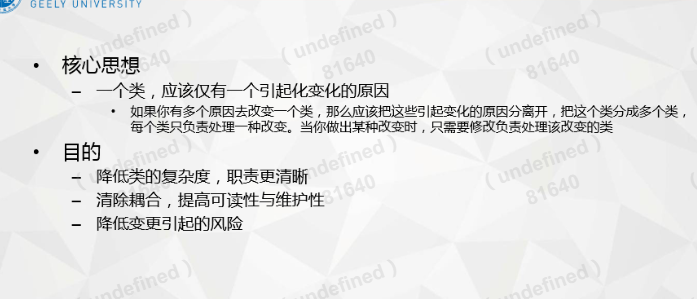
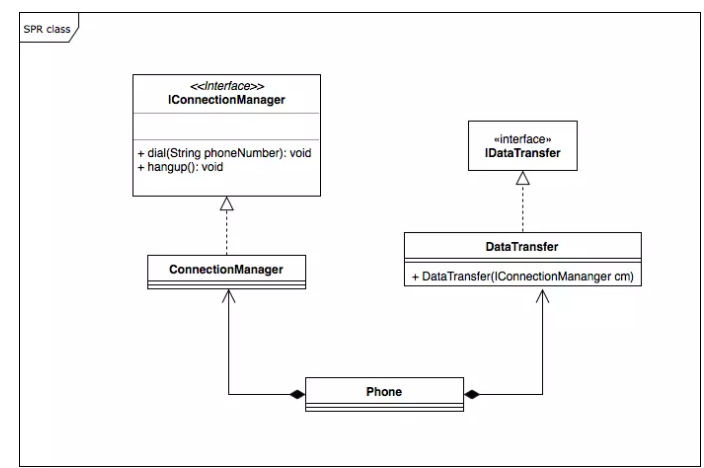
**一、单一职责原则**

**1、定义**

**应该有且仅有一个原因引起类的变更**



**2、实现**

**（1）定义通话连接服务**

public interface IConnectionManager {

void dial(String phoneNumber); // 拨打电话

void hangup(); // 挂断电话

}

**（2）实现通话连接服务**

public class ConnectionManager implements IConnectionManager {

@Override

public void dial(String phoneNumber) {

Log.d("JimmyZhang", "拨打电话");

}

@Override

public void hangup() {

Log.d("JimmyZhang", "挂断电话");

}

}

**（3）定义通话数据传输服务**

public interface IDataTransfer {

public void chat(Object o);

}

**（4）实现通话数据传输服务**

public class DataTransfer implements IDataTransfer {

IConnectionManager cm;

public DataTransfer(IConnectionManager cm) {

this.cm = cm;

}

@Override

public void chat(Object o) {

Log.d("JimmyZhang", "通话");

}

}

**（5）实现通话功能**

DataTransfer dataTransfer;

public void setDataTransfer(DataTransfer dataTransfer) {

this.dataTransfer = dataTransfer;

}

public void dial(String phoneNumber) {

dataTransfer.cm.dial(phoneNumber);

}

public void hangup() {

dataTransfer.cm.hangup();

}

public void chat(Object o) {

dataTransfer.chat(o);

}

**3、优点**

**（1）类的复杂性降低、实现什么职责都有清晰明确的定义；**

**（2）可读性提高，复杂性降低；**

**（3）可维护性提高；**

**（4）提高系统可维护性。**