我将带大家一起去看一看在面向对象编程中出现的23大设计模式和9大设计原则。设计原则是我们开发研究设计模式的目的，而设计模式则是达到设计原则的手段和方法。在记录笔记过程中，会在不同的设计模式中穿插讲解设计原则。
本笔记知识点主要来自于《Head First 设计模式》

行为型模式之四：模板方法模式

定义：
模板方法定义一个方法，在这个方法中定义了一些算法的骨架，但是对于一些算法的具体实现则推迟到具体的子类中。这使得子类继承并保持了模板中的算法结构，但是又可以重新定义算法中的具体操作。

设计原则：
别调用我们，我们会调用你

组成：

- 抽象类：抽象类含有模板方法，定义了算法的基本骨架。如果存在变化的部分，可以在子类中进行具体实现；此外，抽象类中还可以存在hook(钩子)方法，它的意义是给子类覆盖，用来完成子类需要对既有算法骨架的改变；
- 具体类：实现抽象类，对于那些抽象方法给出自己的实现；如果对既有算法骨架需要修改，可以重写钩子方法；

看代码：

```
在这个例子中，老师和学生都要去学校，两人行为类型一致，但是具体行动不同。可以利用模板方法模式。
//抽象类
public abstract class Person{
    public void goToSchool() {
        dressUp();
        eatBreakfast();
        takeThings();
    }
    public abstract void dressUp();
    public abstract void eatBreakfast();
    public abstract void takeThings();
}

//具体类
public class Student extends person{
    public void dressUp() {
        System.out.println("Put on school uniform");
    }
    public void eatBreakfast() {
        System.out.println("Eat bread and milk");
    }
    public void takeThings() {
        System.out.println("Take books");
    }
}
public class Teacher extends person{
    public void dressUp() {
        System.out.println("Put on work uniform");
    }
    public void eatBreakfast() {
        System.out.println("Eat juice and noddles");
    }
    public void takeThings() {
        System.out.println("Take exams");
    }
}
定义一个抽象类Person，并且定义算法骨架去学校的三个步骤。然后再两个具体的类中对于自己的行动给出具体的实现
```