author：黄继升

**一、软件设计模式的关键基本要素**

（1）模式名称

（2）问题

（3）解决方案

（4）效果

**二、GoF的23种设计模式**

设计模式有两种分类方法：

（1）根据模式的目的来分

创建型模式：用于描述“怎样创建对象”，特点是“将对象的创建与使用分离”。有5种创建型模式：单例、原型、工厂方法、抽象工厂、建造者。

结构型模式：用于描述“如何将类或对象按某种布局组成更大的结构”。有7种结构型模式：代理、适配器、桥接、装饰、外观、享元、组合

行为型模式：用于描述“类或对象之间怎样相互协作共同完成单个对象都无法单独完成的任务，以及怎样分配职责”。有11种行为型模式：模板方法、策略、命令、职责链、状态、观察者、中介者、迭代器、访问者、备忘录、解释器。

（2）根据模式的作用范围来分

类模式：用于处理类与子类之间的关系，这些关系通过继承来建立，是静态的，在编译时刻便确定下来了。有4种类模式：工厂方法、（类）适配器、模板方法、解释器

对象模式：用于处理对象之间的关系，这些关系可以通过组合或聚合来实现，在运行时刻是可以变化的，更具动态性。

目的

|  |  |  |  |
| --- | --- | --- | --- |
| 范围 | 创建型模式 | 结构型模式 | 行为型模式 |
| 类模式 | 工厂方法 | （类）适配器 | 模板方法、解释器 |
| 对象模式 | 单例  原型  抽象工厂  建造者 | 代理  （对象）适配器  桥接  装饰  外观  享元  组合 | 策略  命令  职责链  状态  观察者  中介者  迭代器  访问者  备忘录 |

**三、统一建模语言UML**

（1）类、接口和类图

类的UML表示：

类名 +

属性集合 {[可见性]属性名:类型[=默认值]，…} +

操作集合（成员方法集合）{[可见性]名称(参数列表)[:返回类型]}

可见性：public + 、 private - 、 protected # 、 friendly ~

eg：学生类的UML表示：

|  |
| --- |
| 学生  Student |
| -no:long  -name:String  -school:String  -totalScore:float |
| +display():void |

接口的UML表示：

eg：Graph接口的UML表示：

Graph

+getArea():double

+getPerimeter():double

类图是用来表示系统中的类、接口、协作以及它们之间的静态结构和关系的一种静态模型，用于描述软件系统的结构化设计。

|  |
| --- |
| 圆形  Circular |
| -radius:double |
| +getArea():double  +getPerimeter():double |

|  |
| --- |
| 圆形  Client |
|  |
| +calculate(Graph tx):void |

Graph

+getArea():double

+getPerimeter():double

|  |
| --- |
| 长方形  Rectangle |
| -length:double  -width:double |
| +getArea():double  +getPerimeter():double |

（2）类之间的关系 （按照类与类之间的耦合度从弱到强排列）

1、依赖关系

一种使用关系，耦合度最弱。在代码中，某个类的方法通过局部变量、方法参数或者对静态方法的调用来访问被依赖类中的某些方法来完成功能。

|  |
| --- |
| 人  Person |
| -name: String |
| +call (MobilePhone mp):void |

|  |
| --- |
| 手机  MobilePhone |
|  |
| +transfer() : void |

打电话

2、关联关系

表示对象之间的一种引用关系，用于表示一类对象与另一类对象之间的联系，如老师和学生，丈夫和妻子等。分为一般关联关系、聚合关联关系、组合关联关系。

3、聚合关系

4、组合关系

5、泛化关系

6、实现关系