

1. 24设计模式-桥接模式

- 1 桥接模式概念介绍
- 2 桥接模式uml图介绍
- 3 原型模式-实例说明-模拟毛笔
- 4 原型模式-实例代码

2.DocumentBuilderFactory

- 1 DocumentBuilderFactory用于创建DOM模式的解析器对象，DocumentBuilderFactory是一个抽象工厂类，它不能直接实例化，但该类提供了一个newInstance方法，这个方法会根据本地平台默认安装的解析器，自动创建一个工厂的对象并返回。

3.DocumentBuilder

- 1 调用工厂对象的 newDocumentBuilder方法得到 DOM 解析器对象。

4.File

- 1 实例化后，使用构造方法创建一个File对象

5.Document

- 1 调用 DOM 解析器对象的 parse() 方法解析 XML 文档，得到代表整个文档的 Document 对象，进行可以利用DOM特性对整个XML文档进行操作了。

6.NodeList

- 1 NodeList 对象代表一个有顺序的节点列表。
- 2
- 3 我们可通过节点列表中的节点索引号来访问列表中的节点（索引号由0开始）。
- 4 节点列表可保持其自身的更新。如果节点列表或 XML 文档中的某个元素被删除或添加，列表也会被自动更新。

7.实例化

- 1 new Class()
- 2 类名.class
- 3 使用Class类的静态方法forName()，用类的名字获取一个Class实例（static Class forName(String className)），这种方式灵活性最高，根据类的字符串全名即可获取Class实例，可以动态加载类，框架设计经常用到；

2.桥接模式优缺点

优：

- 1 分离抽象接口及其实现部分
- 2 可以取代多层继承方案，极大地减少了子类的个数
- 3 提高了系统的可扩展性，在两个变化维度中任意扩展一个维度，不需要修改原有系统，符合开闭原则

缺：

- 1 会增加系统的理解与设计难度，由于关联关系建立在抽象层，要求开发者一开始就要针对抽象层进行设计与编程
- 2 正确识别出系统中两个独立变化的维度并不是一件容易的事情

idea自定义快捷键设置

- 1 **font-size:** 改变字体大小
- 2 **Run:** 快速运行