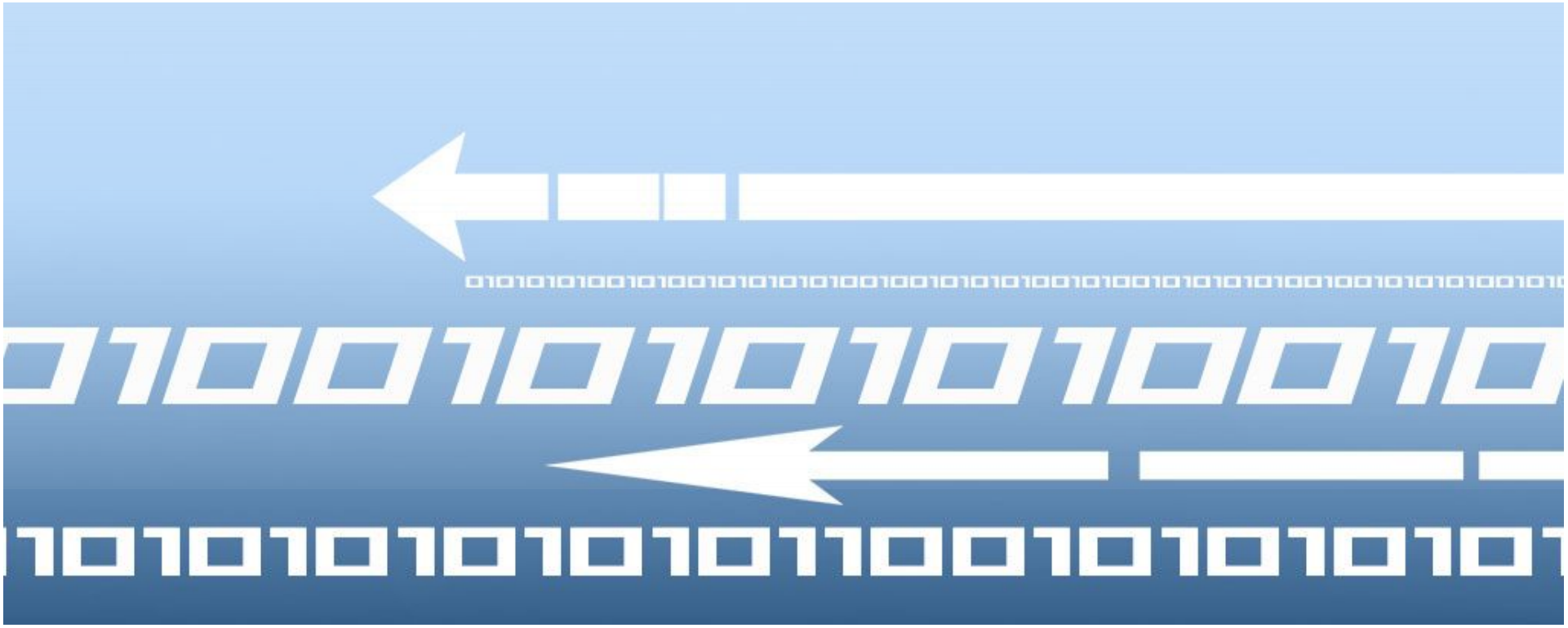


抽象工厂模式

包子

2014-07

1 抽象工厂模式概述

2 抽象工厂模式的结构与实现

3 抽象工厂模式的应用实例

4 抽象工厂模式的模式扩展

亨哄纳置聋桃专亨哄智

- ✓ 工厂方法模式
  - 每个具体工厂只有一个或者一组重载的工厂方法，只能生产一种产品，可能会导致系统中存在大量的工厂类，势必会增加系统的开销
- ✓ 抽象工厂模式
  - 一个工厂可以生产一系列产品（一族产品），极大减少了工厂类的数量

✓ 模式动机

- 为了更清晰地理解工厂方法模式，需要先引入两个概念：

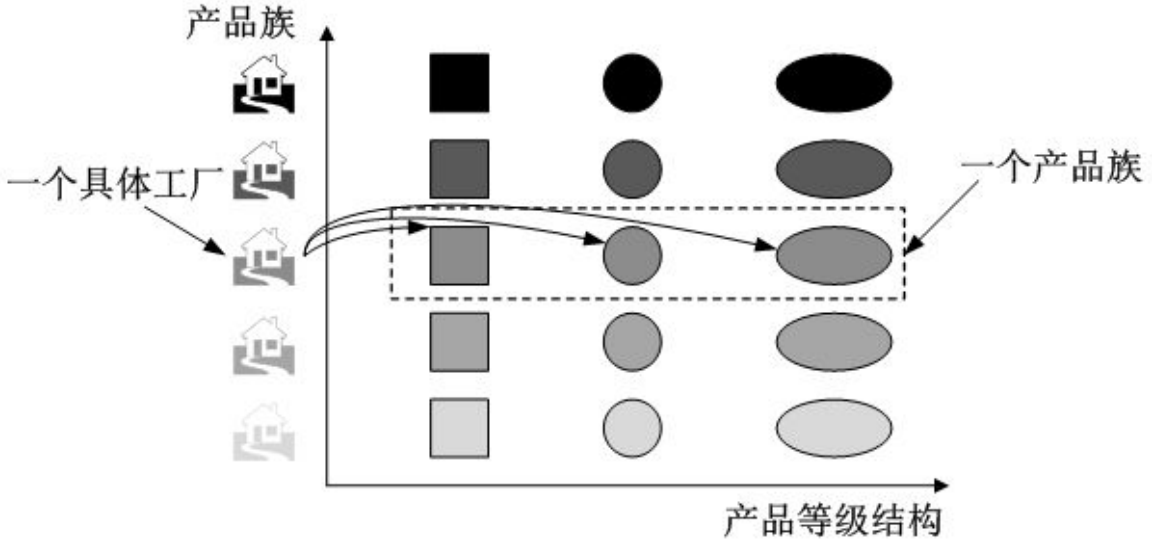
产品等级结构：产品等级结构即产品的继承结构，如一个抽象类是电视机，其子类有海尔电视机、海信电视机、TCL电视机，则抽象电视机与具体品牌的电视机之间构成了一个产品等级结构，抽象电视机是父类，而具体品牌的电视机是其子类。

产品族：在抽象工厂模式中，产品族是指由同一个工厂生产的，位于不同产品等级结构中的一组产品，如海尔电器工厂生产的海尔电视机、海尔电冰箱，海尔电视机位于电视机产品等级结构中，海尔电冰箱位于电冰箱产品等级结构中。

挝踩左厄毫弓比释

✓ 模式动机

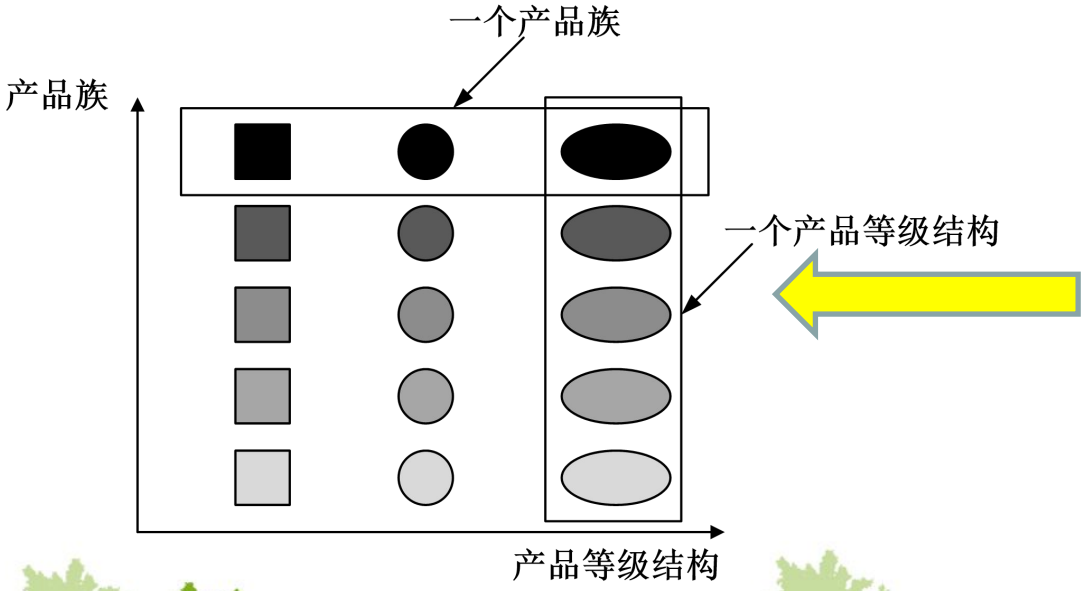
- 当系统所提供的工厂生产的具体产品并不是一个简单的对象，而是多个位于不同产品等级结构、属于不同类型的具体产品时就可以使用抽象工厂模式
- 抽象工厂模式是所有形式的工厂模式中最为抽象和最具一般性的一种形式



亨哄纳置聋桃专亨哄智

✓概念

- 产品等级结构：产品等级结构即产品的继承结构
- 产品族：产品族是指由同一个工厂生产的，位于不同产品等级结构中的一组产品



五个产品族，分属于三个不同的产品等级结构

挝踩左厄毫弓比释

✓抽象工厂模式的定义

抽象工厂模式：提供一个**创建一系列相关或相互依赖对象的接口**，而无须指定它们具体的类

Abstract Factory Pattern: Provide an interface for **creating families of related or dependent objects** without specifying their concrete classes.

挝踩左厄毫弓比释

✓抽象工厂模式的定义

- 又称为工具(Kit)模式
- 抽象工厂模式中的具体工厂不只是创建一种产品，它负责创建一族产品
- 当一个工厂等级结构可以创建出分属于不同产品等级结构的一个产品族中的所有对象时，抽象工厂模式比工厂方法模式更为简单、更有效率

1 抽象工厂模式概述

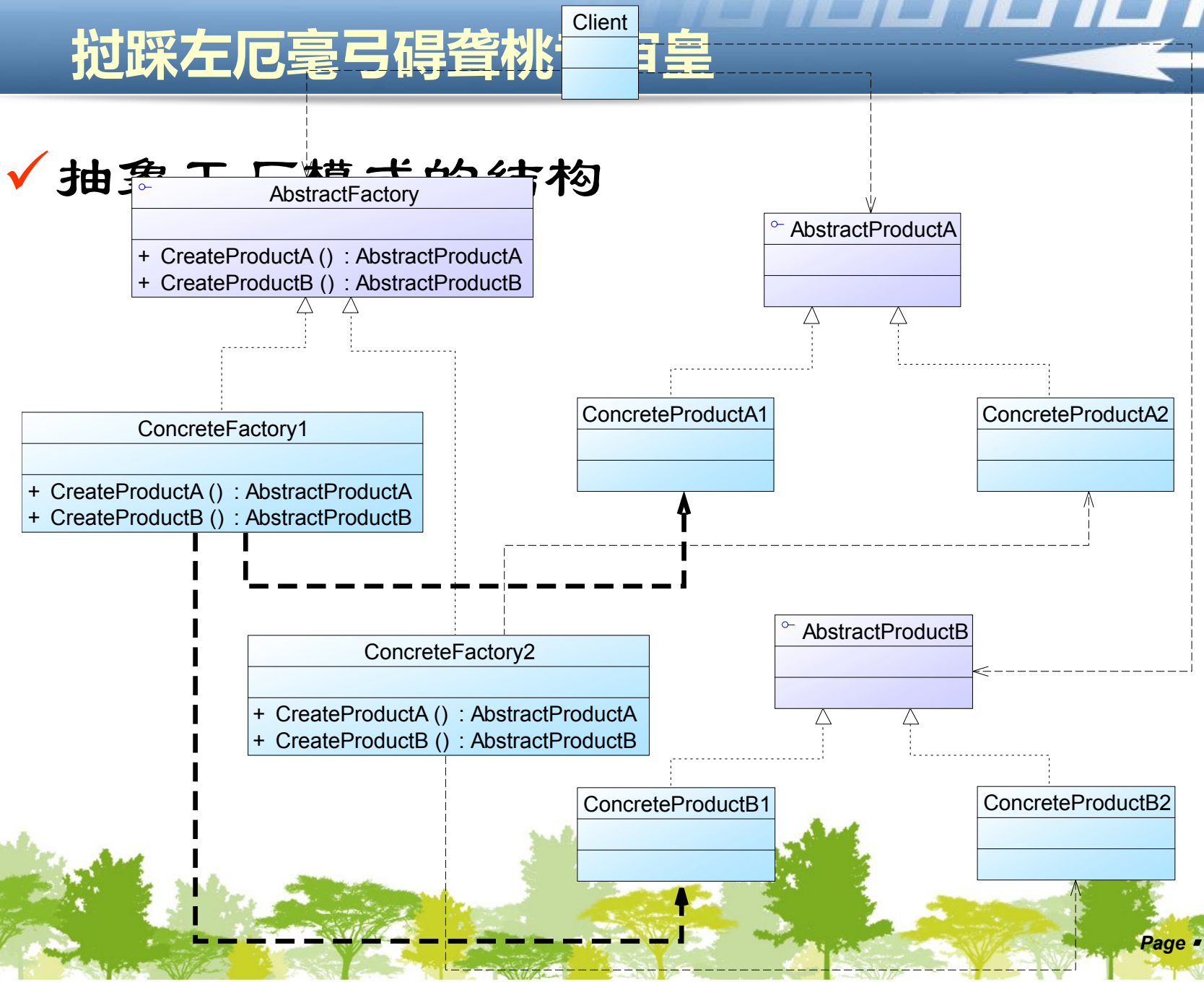
2 抽象工厂模式的结构与实现

3 抽象工厂模式的应用实例

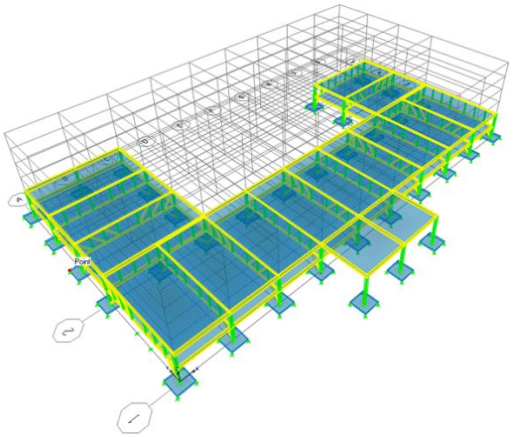
4 抽象工厂模式的模式扩展

挝踩左厄毫弓碍聋桃

✓ 抽


Client
AbstractFactory
+ CreateProductA () : AbstractProductA
+ CreateProductB () : AbstractProductB
AbstractProductA
ConcreteProductA1
ConcreteProductA2
ConcreteFactory1
+ CreateProductA () : AbstractProductA
+ CreateProductB () : AbstractProductB
AbstractProductB
ConcreteFactory2
+ CreateProductA () : AbstractProductA
+ CreateProductB () : AbstractProductB
ConcreteProductB1
ConcreteProductB2

挝踩左厄毫弓碍聋桃专审皇



✓抽象工厂模式的结构

- 抽象工厂模式包含以下4个角色：

  AbstractFactory（抽象工厂）

  ConcreteFactory（具体工厂）

  AbstractProduct（抽象产品）

  ConcreteProduct（具体产品）

✓抽象工厂模式的实现

- 典型的抽象工厂类代码：

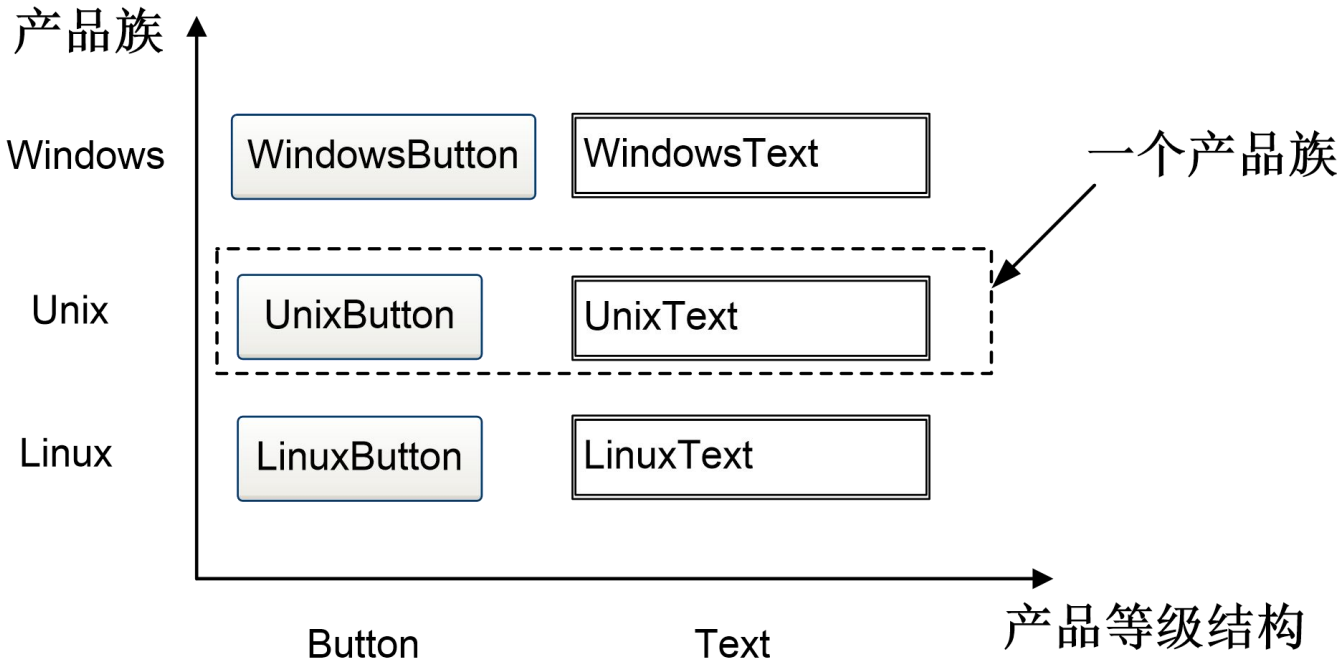
```
abstract class AbstractFactory
{
public abstract AbstractProductA CreateProductA(); //工厂方法一
public abstract AbstractProductB CreateProductB(); //工厂方法二
……
}
```

✓抽象工厂模式的实现

- 典型的具体工厂类代码：

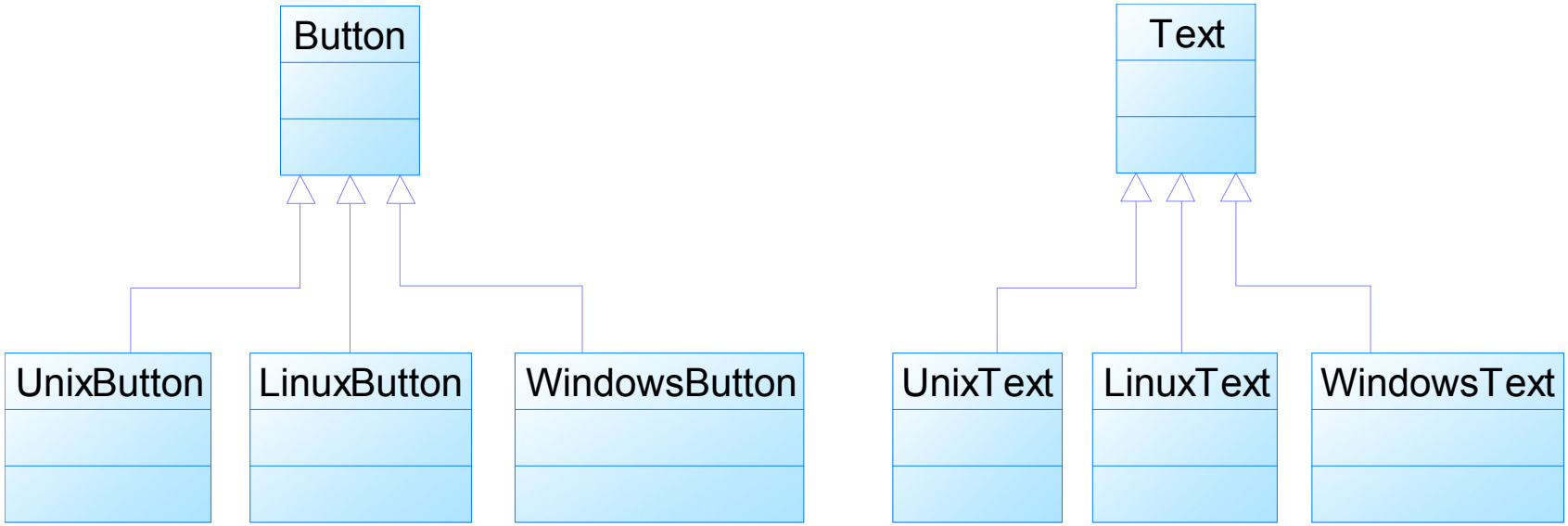
```
class ConcreteFactory1 : AbstractFactory
{
    //工厂方法一
public override AbstractProductA CreateProductA()
{
    return new ConcreteProductA1();
}
//工厂方法二
public override AbstractProductB CreateProductB()
{
    return new ConcreteProductB1();
}
……
}
```

✓模式分析


产品族
Windows
WindowsButton
WindowsText
一个产品族
Unix
UnixButton
UnixText
Linux
LinuxButton
LinuxText
Button
Text
产品等级结构

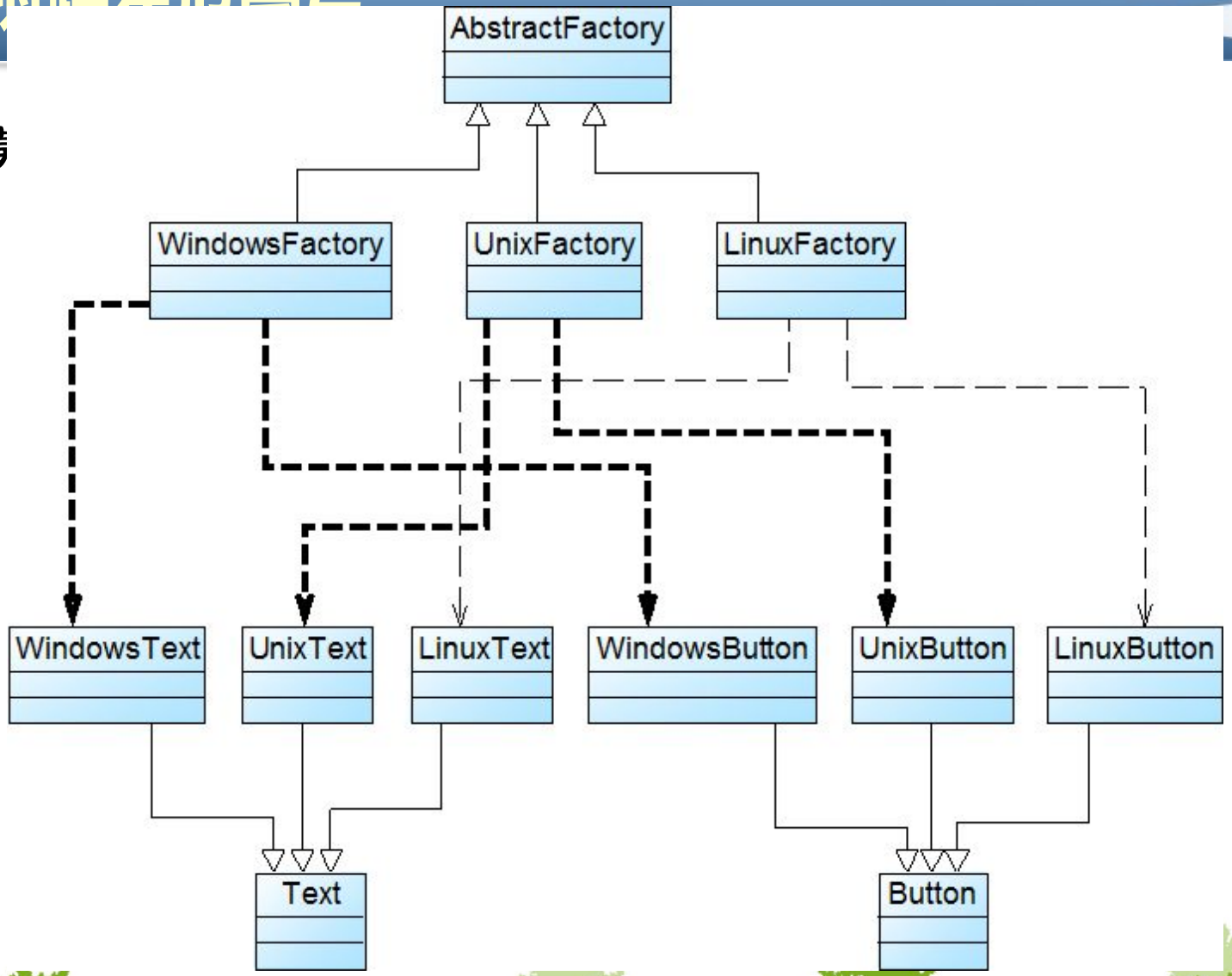
挝踩左厄毫弓

✓模式分析


Button
Text
UnixButton
LinuxButton
WindowsButton
UnixText
LinuxText
WindowsText

抽象工厂模式

- ✓ 模

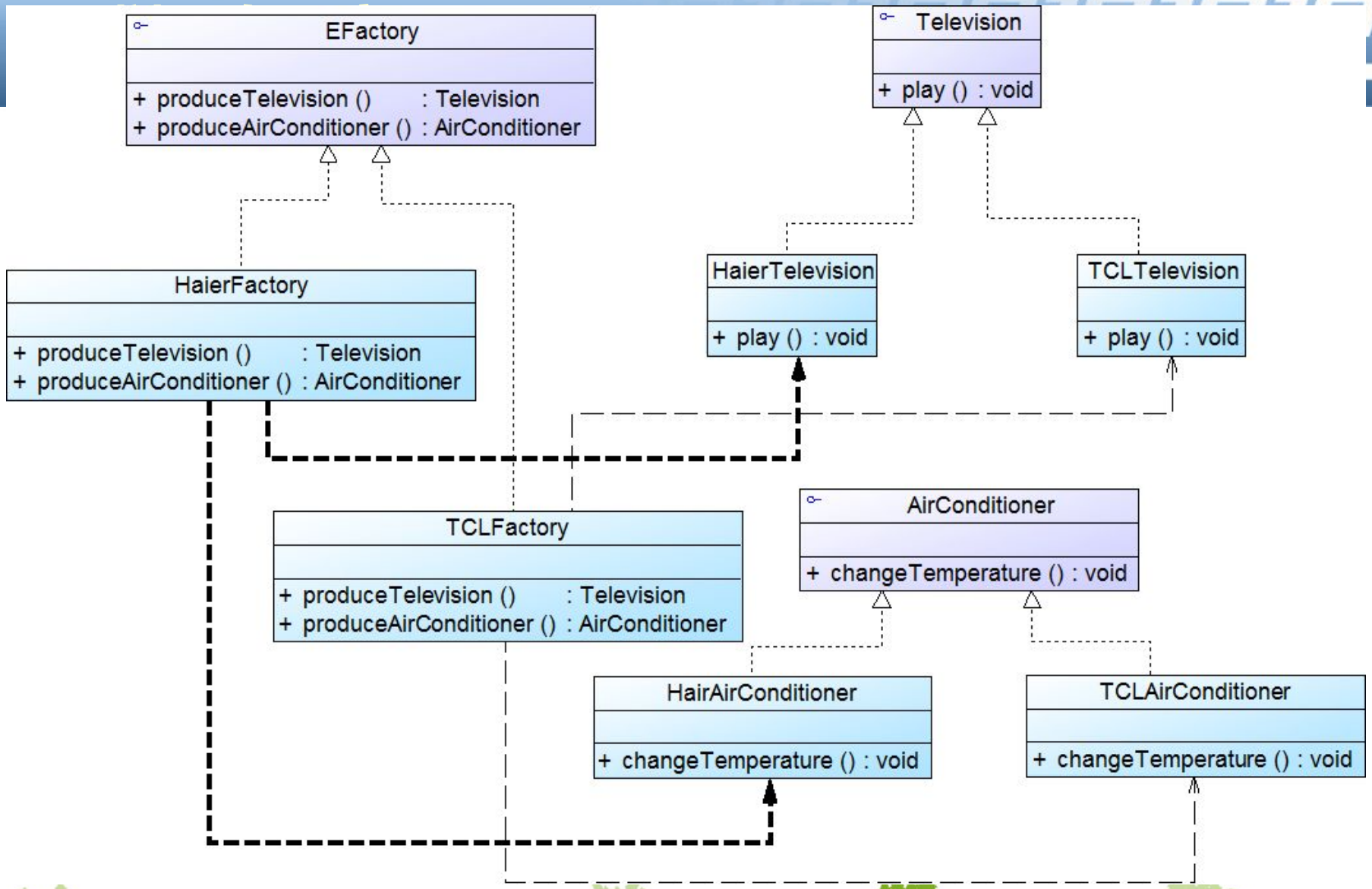

AbstractFactory
WindowsFactory
UnixFactory
LinuxFactory
WindowsText
UnixText
LinuxText
WindowsButton
UnixButton
LinuxButton
Text
Button

1. 抽象工厂模式概述
2. 抽象工厂模式的结构与实现
3. 抽象工厂模式的应用实例
4. 抽象工厂模式的模式扩展

✓模式实例与解析

实例一：电器工厂

一个电器工厂可以产生多种类型的电器，如海尔工厂可以生产海尔电视机、海尔空调等，TCL工厂可以生产TCL电视机、TCL空调等，相同品牌的电器构成一个产品族，而相同类型的电器构成了一个产品等级结构，现使用抽象工厂模式模拟该场景。


EFactory
+ produceTelevision () : Television
+ produceAirConditioner () : AirConditioner
Television
+ play () : void
HaierFactory
+ produceTelevision () : Television
+ produceAirConditioner () : AirConditioner
HaierTelevision
+ play () : void
TCLTelevision
+ play () : void
TCLFactory
+ produceTelevision () : Television
+ produceAirConditioner () : AirConditioner
AirConditioner
+ changeTemperature () : void
HairAirConditioner
+ changeTemperature () : void
TCLAirConditioner
+ changeTemperature () : void

✓模式实例与解析

- 实例一：电器工厂

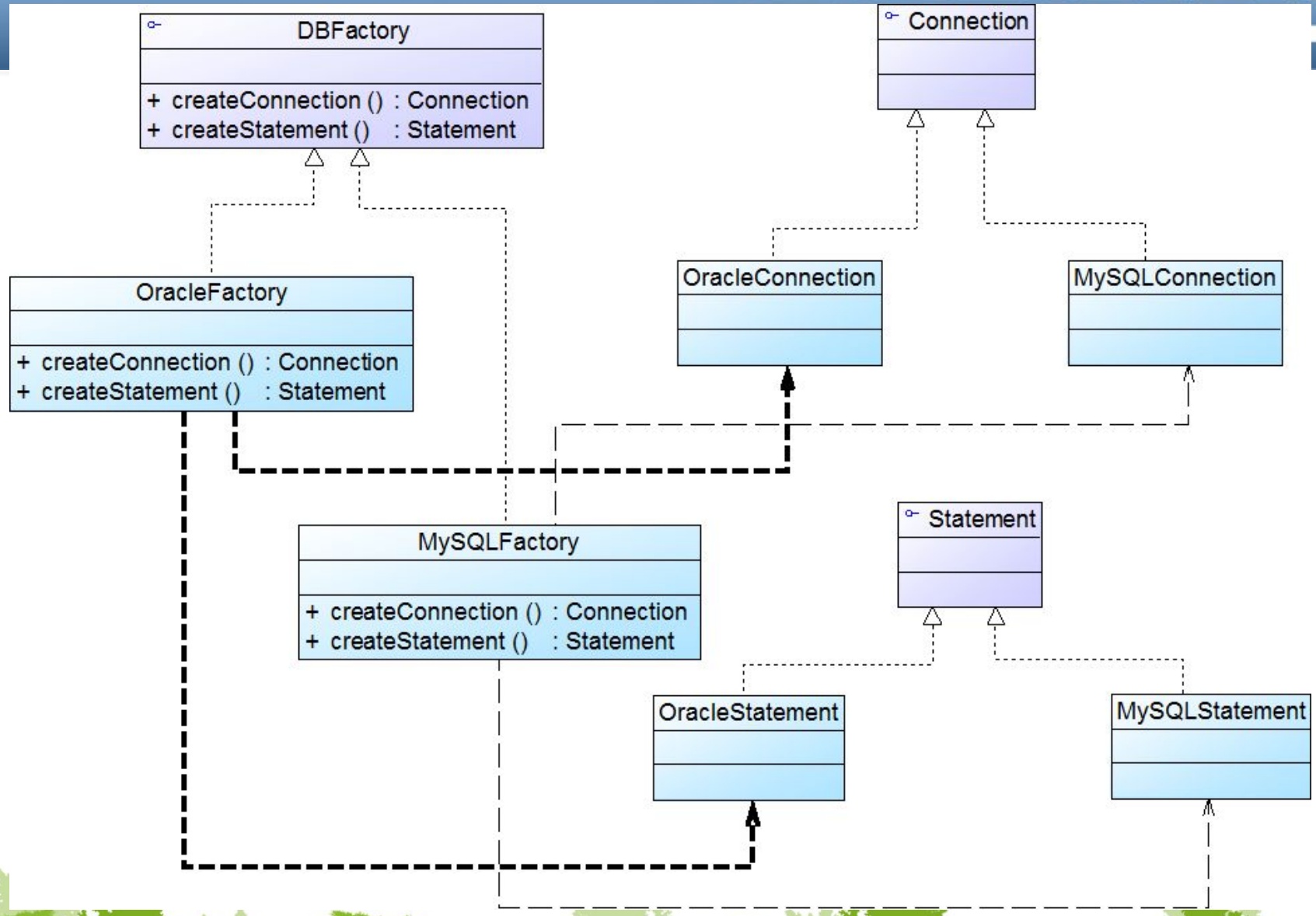
参考代码 (Chapter 06 Abstract Factory\sample01)

演示.....

✓模式实例与解析

实例二：数据库操作工厂

某系统为了改进数据库操作的性能，自定义数据库连接对象Connection和语句对象Statement，可针对不同类型的数据库提供不同的连接对象和语句对象，如提供Oracle或SQL Server专用连接类和语句类，而且用户可以通过配置文件等方式根据实际需要动态更换系统数据库。使用抽象工厂模式设计该系统。


DBFactory
+ createConnection () : Connection
+ createStatement () : Statement
Connection
OracleFactory
+ createConnection () : Connection
+ createStatement () : Statement
OracleConnection
MySQLConnection
MySQLFactory
+ createConnection () : Connection
+ createStatement () : Statement
Statement
OracleStatement
MySQLStatement

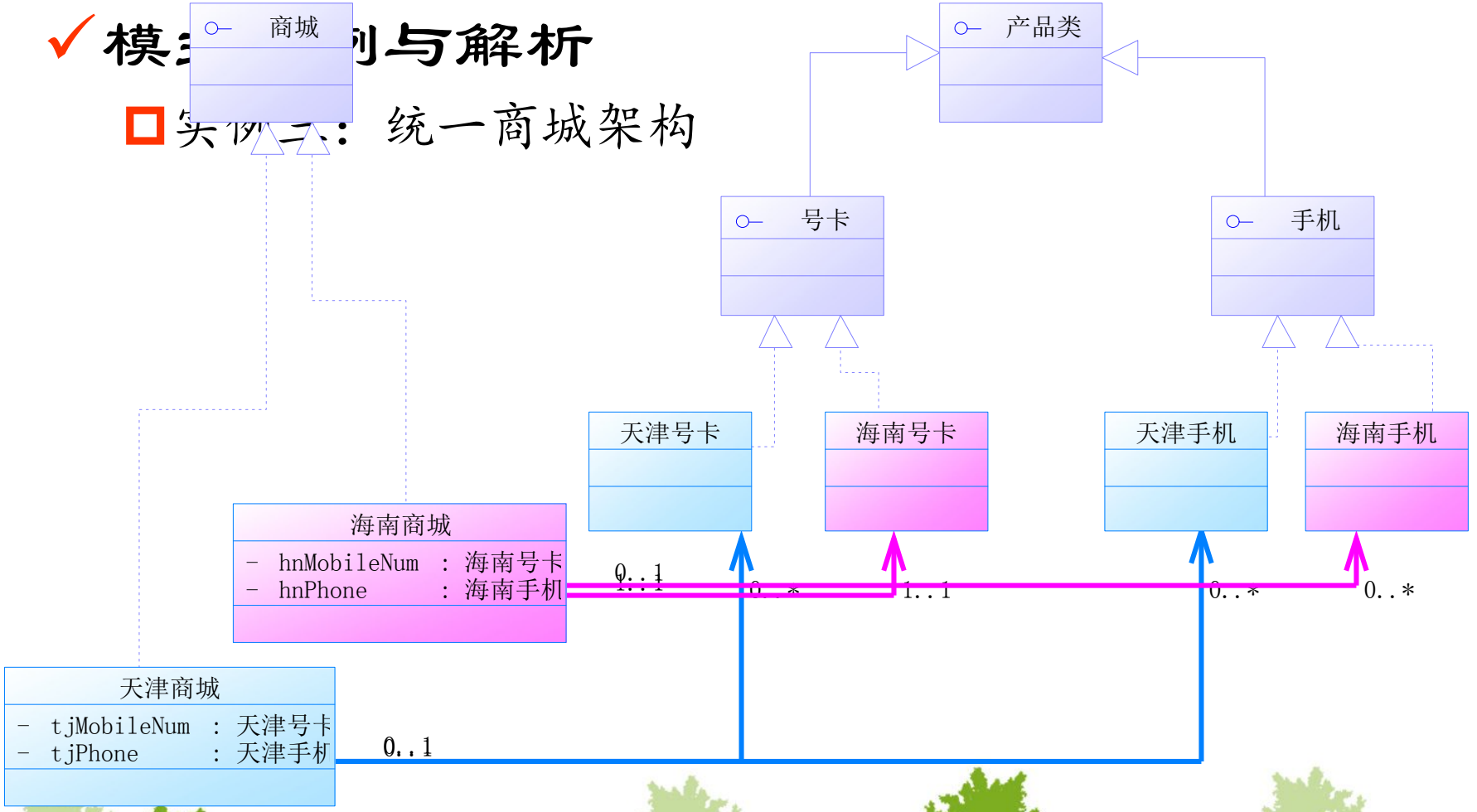
✓ 模式实例与解析

□ 实例三：统一商城架构

某公司承建了多省移动电子渠道商城，虽然承载的产品内容基本一致，但业务逻辑处理方面各省都有区别，如号卡，有些存放在CRM，有些由电渠自己维护；手机的定义也增加了各省的个性化营销，考虑项目实施成本，并考虑以后的市场扩展趋势，公司决定使用同一版本进行管理，可以考虑使用抽象工厂模式进行设计。

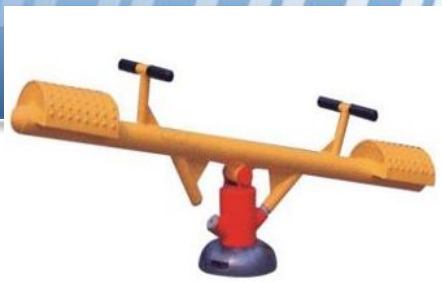
挝踩左厄毫弓

- ✓ 模[illegible]例与解析
  - 实例二：统一商城架构



1. 抽象工厂模式概述
2. 抽象工厂模式的结构与实现
3. 抽象工厂模式的应用实例
4. 抽象工厂模式的模式扩展

异需去刚碍假是懂



✓增加产品族

- 对于增加新的产品族，抽象工厂模式很好地支持了开闭原则，只需要增加具体产品并对应增加一个新的具体工厂，对已有代码无须做任何修改

✓增加新的产品等级结构

- 对于增加新的产品等级结构，需要修改所有的工厂角色，包括抽象工厂类，在所有的工厂类中都需要增加生产新产品的方法，违背了开闭原则

挝踩左厄毫弓碍伙腊玉专釜看皆墙

✓模式优点

- 隔离了具体类的生成，使得客户端并不需要知道什么被创建
- 当一个产品族中的多个对象被设计成一起工作时，它能够保证客户端始终只使用同一个产品族中的对象
- 增加新的产品族很方便，无须修改已有系统，符合开闭原则

挝踩左厄毫弓碍伙腊玉专釜看皆墙

✓模式缺点

- 增加新的产品等级结构麻烦，需要对原有系统进行较大的修改，甚至需要修改抽象层代码，这显然会带来较大的不便，违背了开闭原则

挝踩左厄毫弓碍伙腊玉专釜看皆墙

✓模式适用环境

- 一个系统不应当依赖于产品类实例如何被创建、组合和表达的细节
- 系统中有多于一个的产品族，但每次只使用其中某一产品族
- 属于同一个产品族的产品将在一起使用，这一约束必须在系统的设计中体现出来
- 产品等级结构稳定，设计完成之后，不会向系统中增加新的产品等级结构或者删除已有的产品等级结构

✓模式应用

□Java SE AWT（抽象窗口工具包）

在Java语言的AWT（抽象窗口工具包）中就使用了抽象工厂模式，它使用抽象工厂模式来实现在不同的操作系统中应用程序呈现与所在操作系统一致的外观界面。

✓模式应用

- 在很多软件系统中需要更换界面主题，要求界面中的按钮、文本框、背景色等一起发生改变时，可以使用抽象工厂模式进行设计。

✓ 模式扩展

□ 工厂模式的退化

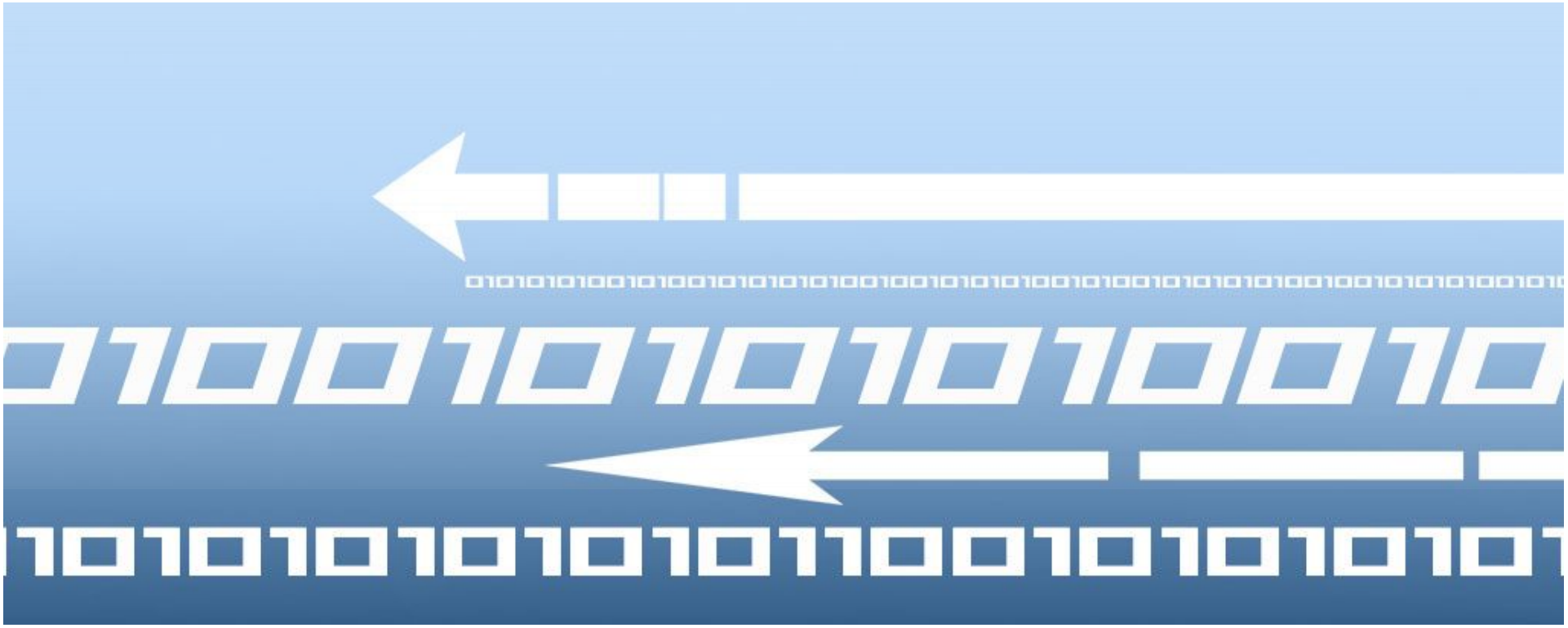
当抽象工厂模式中每一个具体工厂类只创建一个产品对象，也就是只存在一个产品等级结构时，抽象工厂模式退化成工厂方法模式；当工厂方法模式中抽象工厂与具体工厂合并，提供一个统一的工厂来创建产品对象，并将创建对象的工厂方法设计为静态方法时，工厂方法模式退化成简单工厂模式。

柏繁少聋

- ✓ 抽象工厂模式提供一个创建一系列相关或相互依赖对象的接口，而无须指定它们具体的类。抽象工厂模式又称为Kit模式，属于对象创建型模式。
- ✓ 抽象工厂模式包含四个角色：抽象工厂用于声明生成抽象产品的方法；具体工厂实现了抽象工厂声明的生成抽象产品的方法，生成一组具体产品，这些产品构成了一个产品族，每一个产品都位于某个产品等级结构中；抽象产品为每种产品声明接口，在抽象产品中定义了产品的抽象业务方法；具体产品定义具体工厂生产的具体产品对象，实现抽象产品接口中定义的业务方法。
- ✓ 抽象工厂模式是所有形式的工厂模式中最为抽象和最具一般性的一种形态。抽象工厂模式与工厂方法模式最大的区别在于，工厂方法模式针对的是一个产品等级结构，而抽象工厂模式则需要面对多个产品等级结构。

柏繁少聋

- ✓ 抽象工厂模式的主要优点是隔离了具体类的生成，使得客户并不需要知道什么被创建，而且每次可以通过具体工厂类创建一个产品族中的多个对象，增加或者替换产品族比较方便，增加新的具体工厂和产品族很方便；主要缺点在于增加新的产品等级结构很复杂，需要修改抽象工厂和所有的具体工厂类，对“开闭原则”的支持呈现倾斜性。
- ✓ 抽象工厂模式适用情况包括：一个系统不应当依赖于产品类实例如何被创建、组合和表达的细节；系统中有多于一个的产品族，而每次只使用其中某一产品族；属于同一个产品族的产品将在一起使用；系统提供一个产品类的库，所有的产品以同样的接口出现，从而使客户端不依赖于具体实现。



感谢观赏

QQ：253692170

13755115139@139.com