

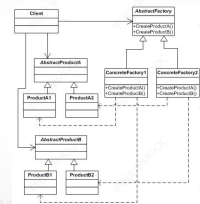
抽象工厂模式

» 定义:

- 定义了创建一系列相关的或依赖的对象的接口，没有规定其具体的类。

» 抽象工厂是一个可以创建其他工厂的超级工厂，是工厂的工厂。

抽象工厂模式



抽象工厂模式

» 参与者

» AbstractFactory

- 定义一个创建抽象产品的操作接口

抽象工厂模式

» ConcreteFactory

- 实现创建具体产品对象的操作接口

» AbstractProduct

- 声明一类产品对象的接口

» Product

- 定义由相应ConcreteFactory创建的产品对象。
- 实现抽象产品接口

抽象工厂模式

» Client

- 用户接口类

例子

» 花园布局程序

» 供选择植物

- 一年生，蔬菜，多年生植物

» 问题

- 花园边界种植的植物类型
- 花园中心种植的植物类型
- 阴凉处种植的植物类型
-

例子: 基类

»» A base garden class

```
public abstract class Garden {  
    public abstract Plant getCenter();  
    public abstract Plant getBorder();  
    public abstract Plant getShade();  
}
```

例子

```
public class Plant {  
    String name;  
    public Plant(String pname) {  
        name = pname; //save name  
    }  
    public String getName() {  
        return name;  
    }  
}
```

例子: 子类

```
public class VegieGarden extends Garden {  
    public Plant getShade() {  
        return new Plant("Broccoli");  
    }  
    public Plant getCenter() {  
        return new Plant("Corn");  
    }  
    public Plant getBorder() {  
        return new Plant("Peas");  
    }  
}
```

例子

```
class GardenMaker  
{  
    //Abstract Factory which returns one of three gardens  
    private Garden gd;  
    public Garden getGarden(String gtype)  
    {  
        gd = new VegieGarden(); //default  
        if(gtype.equals("Perennial"))  
            gd = new PerennialGarden();  
        if(gtype.equals("Annual"))  
            gd = new AnnualGarden();  
        return gd;  
    }  
}
```

例子

Garden planner

Garden type

☐ Vegetable

☐ Annual

☒ Perennial

Astilbe

Dicentra

Sedum

Central

Border

Shade

Quit