**抽象工厂模式（Abstract factory pattern）**

**在前几篇文章张分别介绍了工厂方法模式，简单工厂模式**

**简单工厂模式：**

简单工厂模式的工厂类随着产品类的增加需要增加额外的代码

**工厂方法模式：**

工厂方法模式每个具体工厂类只完成单个实例的创建,所以它具有很好的可扩展性

但是在实际应用中，一个工厂不止会创建单一的产品，因为现在的工厂都多元化了，一个工厂创建一系列的产品，这就引入了抽象工厂的概念。

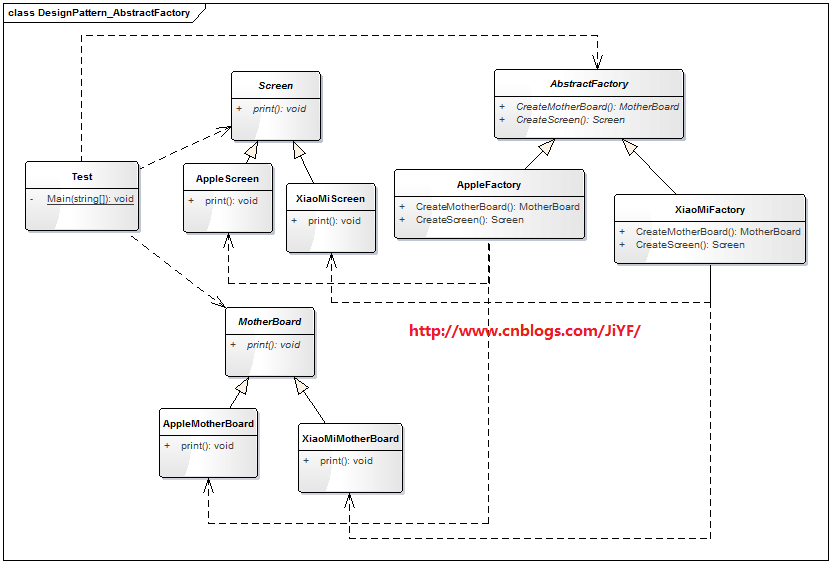
**抽象工厂模式**

抽象工厂模式是围绕一个超级工厂创建其他工厂。该超级工厂又称为其他工厂的工厂。这种类型的设计模式属于创建型模式，它提供了一种创建对象的最佳方式。在抽象工厂模式中，接口是负责创建一个相关对象的工厂，不需要显式指定它们的类。每个生成的工厂都能按照工厂模式提供对象。

**实例：**

下面以一个生产手机主要配件为例子，假设手机工厂需要生产手机屏幕和手机主板，俩条生产线，那么不同品牌的手机工厂生产出的屏幕和主板各不相同。

**工厂方法模式类图：**



**抽象工厂模式C#代码举例：**

**AbstractFactory抽象工厂类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 抽象工厂类：提供创建不同品牌的手机屏幕和手机主板

3 /// </summary>

4 public abstract class AbstractFactory

5 {

6 //工厂生产屏幕

7 public abstract Screen CreateScreen();

8 //工厂生产主板

9 public abstract MotherBoard CreateMotherBoard();

10 }

[复制代码](javascript:void(0);)

**手机屏幕和主板基类**

**Screen手机屏幕类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 屏幕抽象类：提供每一品牌的屏幕的继承

3 /// </summary>

4 public abstract class Screen

5 {

6 public abstract void print();

7 }

[复制代码](javascript:void(0);)

**MotherBoard手机主板类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 主板抽象类：提供每一品牌的主板的继承

3 /// </summary>

4 public abstract class MotherBoard

5 {

6 public abstract void print();

7 }

[复制代码](javascript:void(0);)

**苹果手机屏幕和主板类**

**AppleScreen苹果手机屏幕类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 苹果手机屏幕

3 /// </summary>

4 public class AppleScreen:Screen

5 {

6 public override void print()

7 {

8 Console.WriteLine("苹果手机屏幕！");

9 }

10 }

[复制代码](javascript:void(0);)

**AppleMotherBoard苹果手机主板类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 苹果手机主板

3 /// </summary>

4 public class AppleMotherBoard:MotherBoard

5 {

6 public override void print()

7 {

8 Console.WriteLine("苹果手机主板！");

9 }

10 }

[复制代码](javascript:void(0);)

**小米手机屏幕和主板类**

**XiaoMiScreen小米手机屏幕类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 小米手机屏幕

3 /// </summary>

4 public class XiaoMiScreen:Screen

5 {

6 public override void print()

7 {

8 Console.WriteLine("小米手机屏幕！");

9 }

10 }

[复制代码](javascript:void(0);)

**XiaoMiMotherBoard小米手机主板类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 小米手机主板类

3 /// </summary>

4 public class XiaoMiMotherBoard :MotherBoard

5 {

6 public override void print()

7 {

8 Console.WriteLine("小米手机主板！");

9 }

10

[复制代码](javascript:void(0);)

**苹果手机工厂类和小米手机工厂类**

**AppleFactory苹果手机工厂类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 苹果手机工厂

3 /// </summary>

4 public class AppleFactory : AbstractFactory

5 {

6 /// <summary>

7 /// 生产苹果手机屏幕

8 /// </summary>

9 /// <returns></returns>

10 public override Screen CreateScreen()

11 {

12 return new AppleScreen();

13 }

14

15 /// <summary>

16 /// 生产苹果手机主板

17 /// </summary>

18 /// <returns></returns>

19 public override MotherBoard CreateMotherBoard()

20 {

21 return new AppleMotherBoard();

22 }

23 }

[复制代码](javascript:void(0);)

**XiaoMi手机工厂类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 小米手机工厂类

3 /// </summary>

4 public class XiaoMiFactory : AbstractFactory

5 {

6 /// <summary>

7 /// 生产小米手机屏幕

8 /// </summary>

9 /// <returns></returns>

10 public override Screen CreateScreen()

11 {

12 return new XiaoMiScreen();

13 }

14

15 /// <summary>

16 /// 生产小米手机主板

17 /// </summary>

18 /// <returns></returns>

19 public override MotherBoard CreateMotherBoard()

20 {

21 return new XiaoMiMotherBoard();

22 }

23 }

[复制代码](javascript:void(0);)

**工厂测试类**

[复制代码](javascript:void(0);)

1 class Test

2 {

3 static void Main(string[] args)

4 {

5 //小米工厂生产小米手机的屏幕和主板

6 AbstractFactory xiaomiFactory = new XiaoMiFactory();

7 Screen xiaomiScreen = xiaomiFactory.CreateScreen();

8 xiaomiScreen.print();

9 MotherBoard xiaomiMotherBoard = xiaomiFactory.CreateMotherBoard();

10 xiaomiMotherBoard.print();

11

12 //苹果工厂生产苹果手机屏幕和主板

13 AbstractFactory appleFactory = new AppleFactory();

14 Screen appleScreen = appleFactory.CreateScreen();

15 appleScreen.print();

16 MotherBoard appleMotherBoard = appleFactory.CreateMotherBoard();

17 appleMotherBoard.print();

18

19 Console.Read();

20 }

21 }

[复制代码](javascript:void(0);)

**运行结果：**

