**代理模式（Proxy Pattern）**

**简单介绍：**

代理模式（Proxy Pattern）：

当无法直接访问某个对象或访问某个对象存在困难时可以通过一个代理对象来间接访问，为了保证客户端使用的透明性，所访问的真实对象与代理对象需要实现相同的接口。根据代理模式的使用目的不同，代理模式又可以分为多种类型，例如保护代理、远程代理、虚拟代理、缓冲代理等，它们应用于不同的场合，满足用户的不同需求。

**用户------>代理------->对象**

生活中其实也有许多的例子比如买一部手机，国内购买比较贵，因为涉及到上税的问题，所以好多人从香港或者美国等其他地方代购，

**客户----->代购------->香港**

大概就是这么个意思。

代理模式主要组成部分：

Proxy 维持一个引用，使得代理可以访问Subject。

　　提供一个与Subject的接口相同的接口，这样代理就可以替代Subject。

　　控制对Subject的访问，并可能负责对Subject的创建和删除。

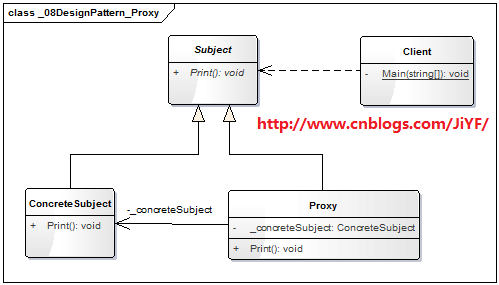
Subject：定义ConcreteSubject与Proxy的共用接口，从而在任何使用ConcreteSubject的地方都可以使用Proxy。

　　ConcreteSubject：定义Proxy所代表的Subject。

　　Client：维持一个对Subject的引用

　　在代理模式中，由于Proxy与ConcreteSubject继承同一接口，所以Client调用ConcreteSubject可以转化为Client调用Proxy在调用ConcreteSubject，类Proxy为中间代理。

**代理模式类图：**



**代理模式C#代码举例：**

**Subject类 对外提供的接口类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 代理类和对象的共用接口

3 /// </summary>

4 public abstract class Subject

5 {

6 public abstract void Print();

7 }

[复制代码](javascript:void(0);)

**ConcreteSubject类代理类所创建代表的对象**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 代理类所创建的对象

3 /// </summary>

4 public class ConcreteSubject :Subject

5 {

6 public override void Print()

7 {

8 Console.WriteLine("ConcreteSubject");

9 }

10 }

[复制代码](javascript:void(0);)

**Proxy代理类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 代理类

3 /// </summary>

4 public class Proxy : Subject

5 {

6 private ConcreteSubject \_concreteSubject;

7 public override void Print()

8 {

9 if (\_concreteSubject == null)

10 {

11 \_concreteSubject = new ConcreteSubject();

12 }

13 \_concreteSubject.Print();

14 }

15 }

[复制代码](javascript:void(0);)

**用户测试类：**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 客户端测试类

3 /// </summary>

4 class Client

5 {

6 static void Main(string[] args)

7 {

8 //创建代理类

9 Proxy proxy = new Proxy();

10 proxy.Print();

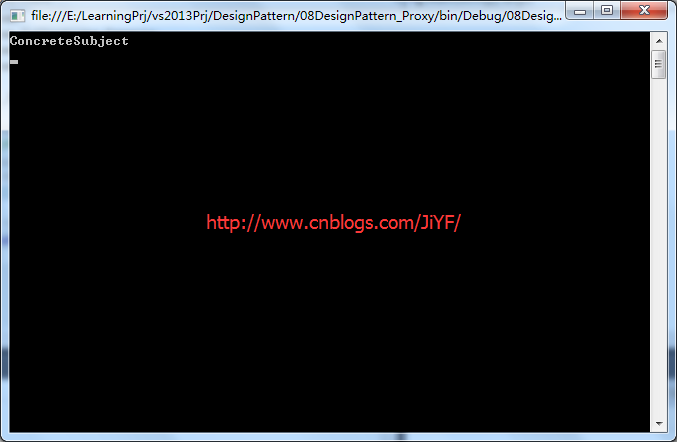
11 Console.Read();

12 }

13 }

[复制代码](javascript:void(0);)

**运行结果：**



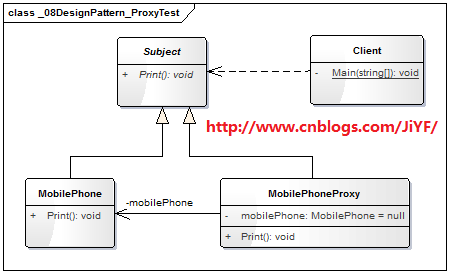
**代理模式源代码工程文件下载**

**代理模式生活例子举例：**

生活中其实也有许多的例子比如买一部手机，国内购买比较贵，因为涉及到上税的问题，所以好多人从香港或者美国等其他地方代购，

**客户----->代购------->香港**

**类图：**

****

**代理模式生活例子C#代码举例：**

**Subject类 ：对外接口类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 对象接口

3 /// </summary>

4 public abstract class Subject

5 {

6 public abstract void Print();

7 }

[复制代码](javascript:void(0);)

**MobilePhone类 具体手机类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 手机类

3 /// </summary>

4 public class MobilePhone : Subject

5 { public override void Print()

6 {

7 Console.WriteLine("我是一部苹果手机!");

8 }

9 }

[复制代码](javascript:void(0);)

**MobilePhoneProxy类 手机代理类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 手机代理类

3 /// </summary>

4 public class MobilePhoneProxy :Subject

5 {

6 private MobilePhone mobilePhone =null;

7

8 public override void Print()

9 {

10 if (mobilePhone == null)

11 {

12 mobilePhone = new MobilePhone();

13 }

14 mobilePhone.Print();

15 }

16 }

[复制代码](javascript:void(0);)

**用户测试类**

[复制代码](javascript:void(0);)

1 /// <summary>

2 /// 客户端调用测试类

3 /// </summary>

4 class Client

5 {

6 static void Main(string[] args)

7 {

8 //创建代理类既可以拿到手机

9 MobilePhoneProxy mobilePhoneProxy = new MobilePhoneProxy();

10 mobilePhoneProxy.Print();

11 Console.Read();

12 }

13 }

[复制代码](javascript:void(0);)

**运行结果**

