

```

using System;
using System.Threading;

//回调
//
//数据下载
//需求：数据下载完成后，给用户一个提示
class Person
{
    public void DownloadData(string url)
    {
        DownloadTools tool = new DownloadTools();
        tool.DownloadData(url);
        tool.complete(url);
    }
    public void DealWithData(string result)
    {
        ConsoleColor origin = Console.ForegroundColor;
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine(result);
        Console.ForegroundColor = origin;
    }
}
//数据下载完成后的委托
//result： 下载得到的数据
public delegate void DownLoadComplete(string result);
//下载工具类（模拟迅雷的下载工具）
class DownloadTools
{
    //不要使用传user的方式来实现需求
    //因为一旦牵扯到扩展 例如再来一个新的类需要下载数据
    //那么代码就得大改
    //public Person user
    public DownLoadComplete complete;
    //下载数据

```

```

public void DownloadData(string url)
{

    //模拟下载数据
    for (int i = 0; i < 100; i++)
    {
        Thread.Sleep(5);
        Console.Clear();
        Console.WriteLine("下载进度为{0}%", i);
    }
    Console.WriteLine("下载完成");
    complete("Hello World!!");
}
}

class Program
{
    public static void Main(string [] arg)
    {
        Person xiaoming = new Person();
        xiaoming.DealWithData("http://www._____.com");
    }
}

using System;
using System.Threading;

//回调
//
//数据下载
//需求：数据下载完成后，给用户一个提示
class Person
{
    public void DownloadData(string url)
    {
        DownloadTools tool = new DownloadTools();
        tool.DownloadData(url);
    }
}

```

```

        tool.complete(url);
    }
    public void DealWithData(string result)
    {
        ConsoleColor origin = Console.ForegroundColor;
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine(result);
        Console.ForegroundColor = origin;
    }
}

//数据下载完成后的委托
//result: 下载得到的数据
public delegate void DownloadComplete(string result);
//下载工具类（模拟迅雷的下载工具）
class DownloadTools
{
    //不要使用传user的方式来实现需求
    //因为一旦牵扯到扩展 例如再来一个新的类需要下载数据
    //那么代码就得大改
    //public Person user
    public DownloadComplete complete;
    //下载数据
    public void DownloadData(string url)
    {

        //模拟下载数据
        for (int i = 0; i < 100; i++)
        {
            Thread.Sleep(5);
            Console.Clear();
            Console.WriteLine("下载进度为{0}%", i);
        }
        Console.WriteLine("下载完成");
        complete("Hellow World!!");
    }
}

```

```

}
class Program
{
    public static void Main(string [] arg)
    {
        Person xiaoming = new Person();
        xiaoming.DealWithData("http://www._____.com");
    }
}
using System;
using System.Threading;

//回调
//
//数据下载
//需求：数据下载完成后，给用户一个提示
class Person
{
    public void DownloadData(string url)
    {
        DownloadTools tool = new DownloadTools();
        tool.DownloadData(url);
        tool.complete(url);
    }
    public void DealWithData(string result)
    {
        ConsoleColor origin = Console.ForegroundColor;
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine(result);
        Console.ForegroundColor = origin;
    }
}
//数据下载完成后的委托
//result： 下载得到的数据
public delegate void DownLoadComplete(string result);

```

```

//下载工具类（模拟迅雷的下载工具）
class DownloadTools
{
    //不要使用传user的方式来实现需求
    //因为一旦牵扯到扩展 例如再来一个新的类需要下载数据
    //那么代码就得大改
    //public Person user
    public DownloadComplete complete;
    //下载数据
    public void DownloadData(string url)
    {

        //模拟下载数据
        for (int i = 0; i < 100; i++)
        {
            Thread.Sleep(5);
            Console.Clear();
            Console.WriteLine("下载进度为{0}%", i);
        }
        Console.WriteLine("下载完成");
        complete("Hellow World!!");
    }
}

class Program
{
    public static void Main(string [] arg)
    {
        Person xiaoming = new Person();
        xiaoming.DealWithData("http://www._____.com");
    }
}

using System;
using System.Threading;

//回调

```

```

//
//数据下载
//需求：数据下载完成后，给用户一个提示
class Person
{
    public void DownloadData(string url)
    {
        DownloadTools tool = new DownloadTools();
        tool.DownloadData(url);
        tool.complete(url);
    }
    public void DealWithData(string result)
    {
        ConsoleColor origin = Console.ForegroundColor;
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine(result);
        Console.ForegroundColor = origin;
    }
}
//数据下载完成后的委托
//result： 下载得到的数据
public delegate void DownLoadComplete(string result);
//下载工具类（模拟迅雷的下载工具）
class DownloadTools
{
    //不要使用传user的方式来实现需求
    //因为一旦牵扯到扩展 例如再来一个新的类需要下载数据
    //那么代码就得大改
    //public Person user
    public DownLoadComplete complete;
    //下载数据
    public void DownloadData(string url)
    {
        //模拟下载数据
    }
}

```

```
        for (int i = 0; i < 100; i++)
        {
            Thread.Sleep(5);
            Console.Clear();
            Console.WriteLine("下载进度为{0}%", i);
        }
        Console.WriteLine("下载完成");
        complete("Hellow World!!");
    }
}

class Program
{
    public static void Main(string [] arg)
    {
        Person xiaoming = new Person();
        xiaoming.DealWithData("http://www._____.com");
    }
}
```