

对象组合

北京理工大学计算机学院
金旭亮

概述

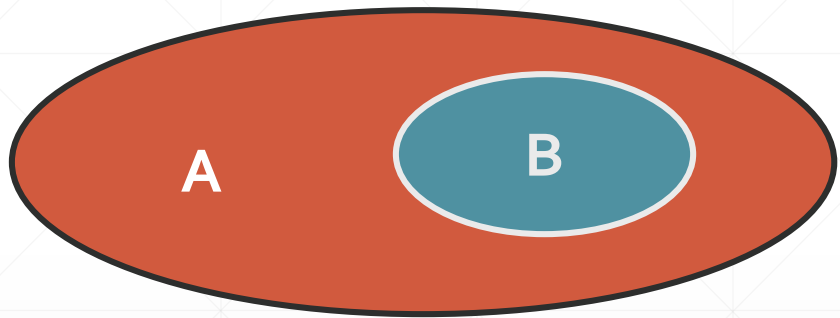
在面向对象成为主流软件开发方式的年代：



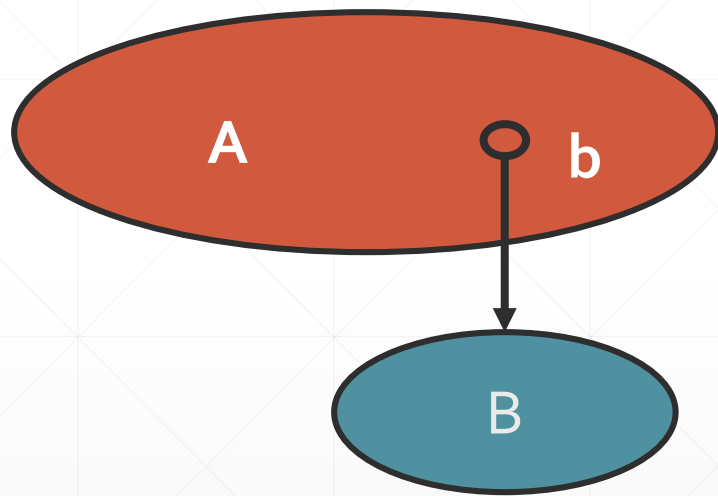
“对象组合”是对象合作关系中的一种，其含义是“一个对象包含另一个对象”。

“一对一”对象组合的两种类型

有两种典型的对象组合方式：

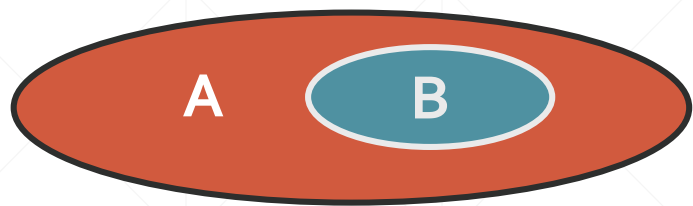


方式一： A对象完全包容B对象，容器对象管理被包容对象的从生到死整个阶段，即“**对象的生命周期**”。

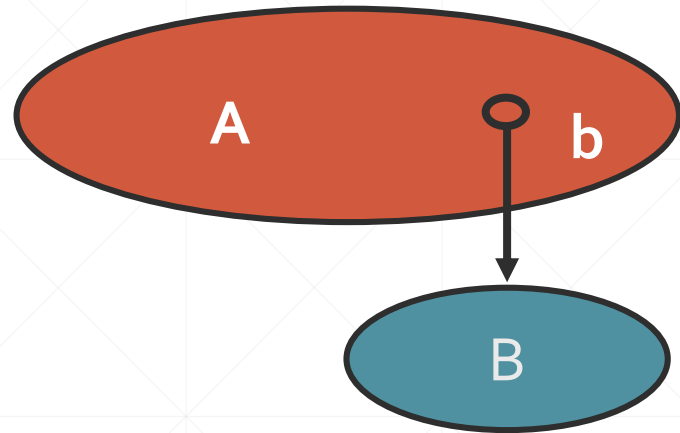


方式二： A和B对象是相互独立的，A对象引用现成的B对象，两者之间是一种比较松散的协作关系。

实现一对一组合



```
/// <summary>
/// 包容方式一
/// 一般情况下，内部对象不能被外界直接访问
/// （当然，根据需要也可以设为Public的）
/// 要在包容类的构造函数中创建被包容对象
/// 如果此种对象仅在本类中使用，还可以把其类的定义为内部类
/// </summary>
3 个引用
class OneToOneClass
{
    private InnerClass obj;
    1 个引用
    public OneToOneClass()
    {
        obj = new InnerClass();
    }
}
```



```
/// <summary>
/// 包容方式二
/// 包容的对象由外界负责创建，通常采用对象注入的方式
/// </summary>
3 个引用
class OneToOneClass2
{
    private InnerClass obj = null;
    1 个引用
    public OneToOneClass2(InnerClass outerObj)
    {
        this.obj = outerObj;
    }
}
```

对象间的“关联”

当两个对象之间是组合与被组合关系时，



这两个对象之间就存在着一种“**关联**”：



关联线上的箭头标明“导航”方向。
意思是：得到一个A对象，就能通过它的属性或字段访问到一个B对象。

对象组合应用示例



示例：OneToOneWinForm

frmMain.Designer.cs

```
partial class frmMain
{
    /// <summary> Required designer variable.
    private System.ComponentModel.IContainer components = null;

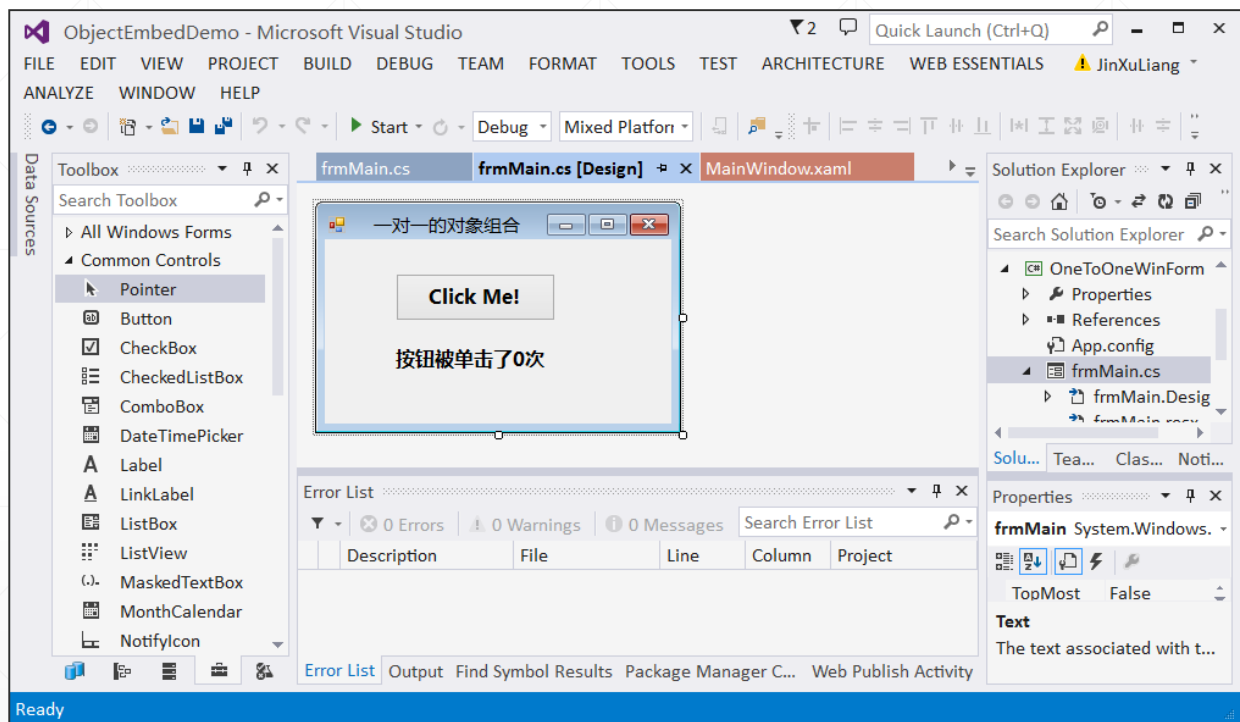
    /// <summary> Clean up any resources being used.
    13 个引用
    protected override void Dispose(bool disposing)...

    Windows Form Designer generated code

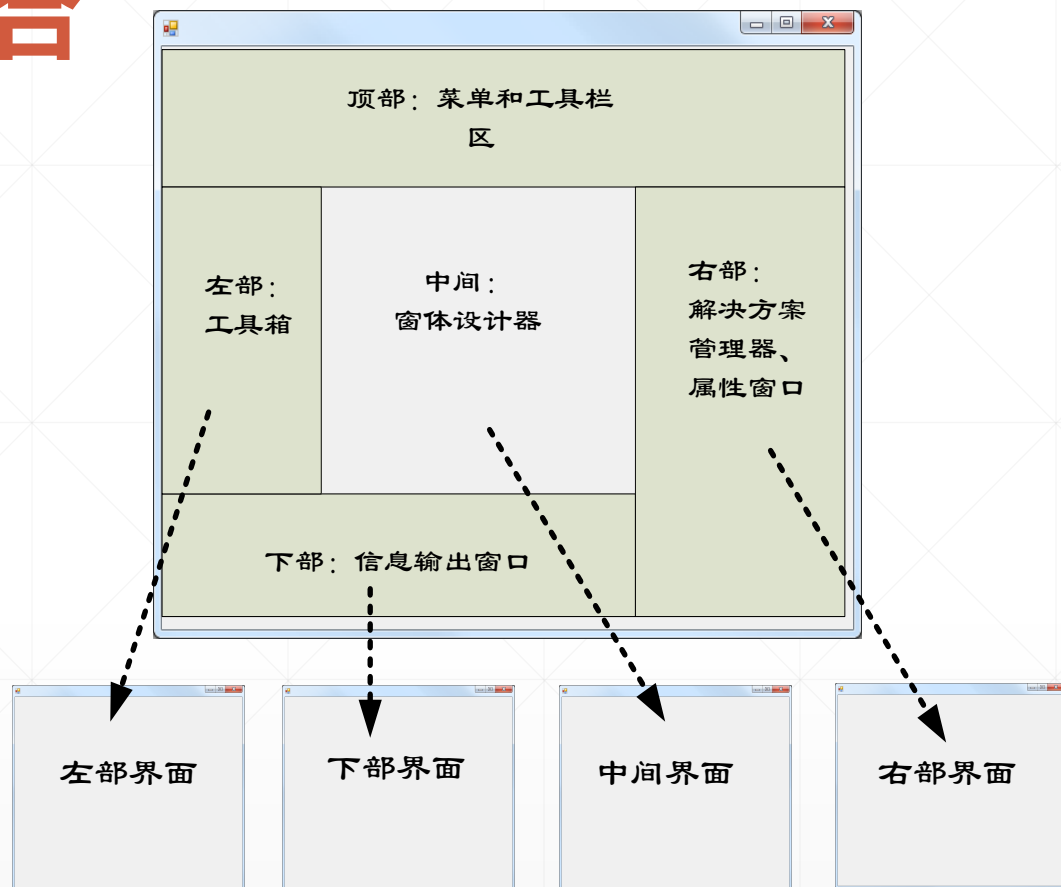
    private System.Windows.Forms.Button btnClickMe;
    private System.Windows.Forms.Label lblInfo;
}
```

本例中，窗体与按钮、标签控件之间的关系就是一对一的组合关系。

对象组合应用实例——窗体组合



Visual Studio主界面包容诸多的控件，非常复杂……

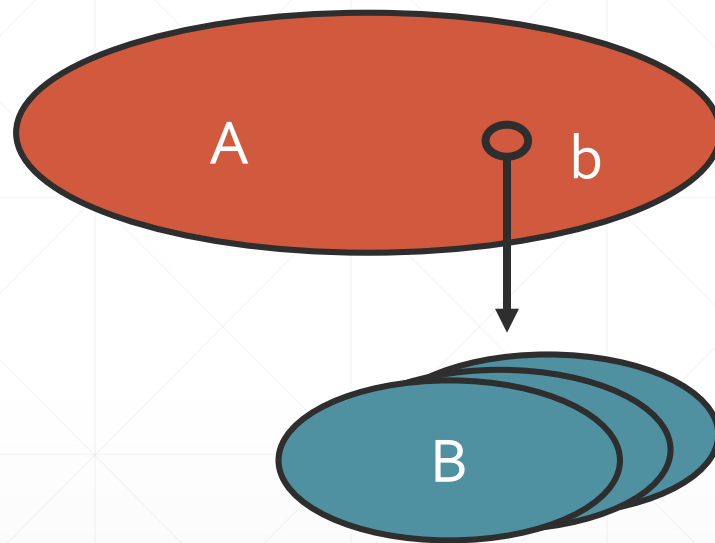


使用延时动态对象组合的方式，可以用于实现类似的界面

一对多的对象组合方式



实例：一辆轿车有四个轮子



实例：机场停机坪上停着多架飞机

编程实现“一对多”的对象组合关系

```
/// <summary>
/// 一对多的对象组合方式一
/// 通常会定义一些公有方法实现向其内部集合中增删对象
/// </summary>
2 个引用
class OneToManyClass
{
    /// <summary>
    /// 内部对象集合
    /// </summary>
    private List<InnerClass> objs=new List<InnerClass>();

    1 个引用
    public void Add(InnerClass obj)
    {
        if (obj != null)
        {
            objs.Add(obj);
        }
    }

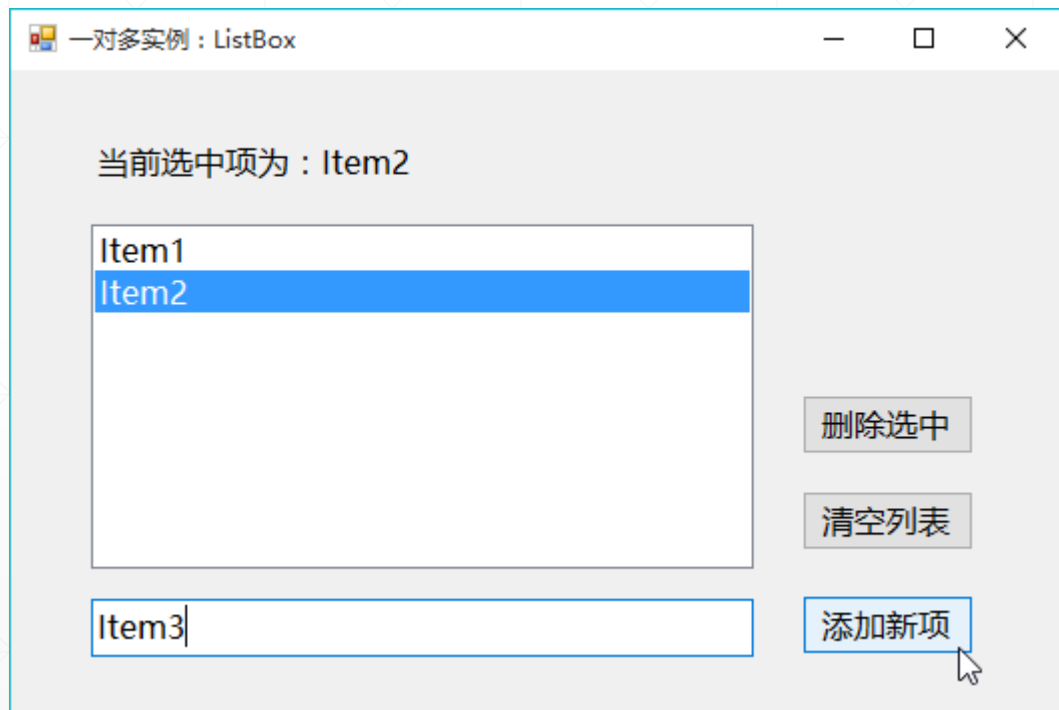
    1 个引用
    public void Remove(int index)
    {
        if (index >= 0 && index < objs.Count)
        {
            objs.RemoveAt(index);
        }
    }
}
```

```
/// <summary>
/// 一对多的对象组合方式二
/// 一般使用对象注入的方式关联上外部对象集合
/// 对象集合放在外部，本身通常不需要提供向此对象集合中增删对象的方法
/// </summary>
/// <typeparam name="T"></typeparam>
3 个引用
class OneToManyClass2<T>
{
    /// <summary>
    /// 内部对象集合
    /// </summary>
    private IEnumerable<T> objs = null;

    1 个引用
    public OneToManyClass2(IEnumerable<T> objCollections)
    {
        objs = objCollections;
    }
}
```

一对多的应用实例——列表框控件

列表框（ListBox）一次可以显示多个列表项。



使用它的**Items**属性添加和移除列表项

如何得到用户选中的项：

SelectedIndex/SelectedItem/SelectValue

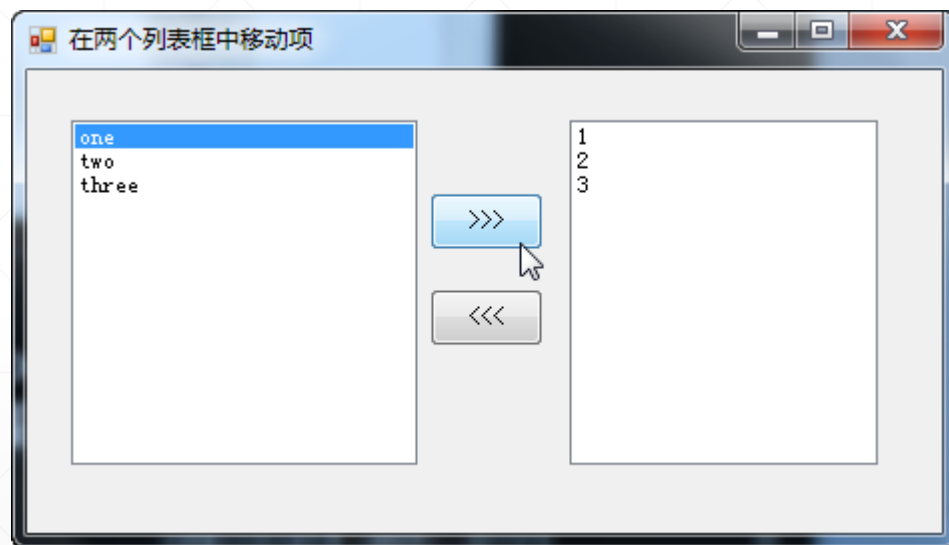
用户选择不同项时，激发事件：

SelectedIndexChanged

示例：OneToManyWinForm

动动手

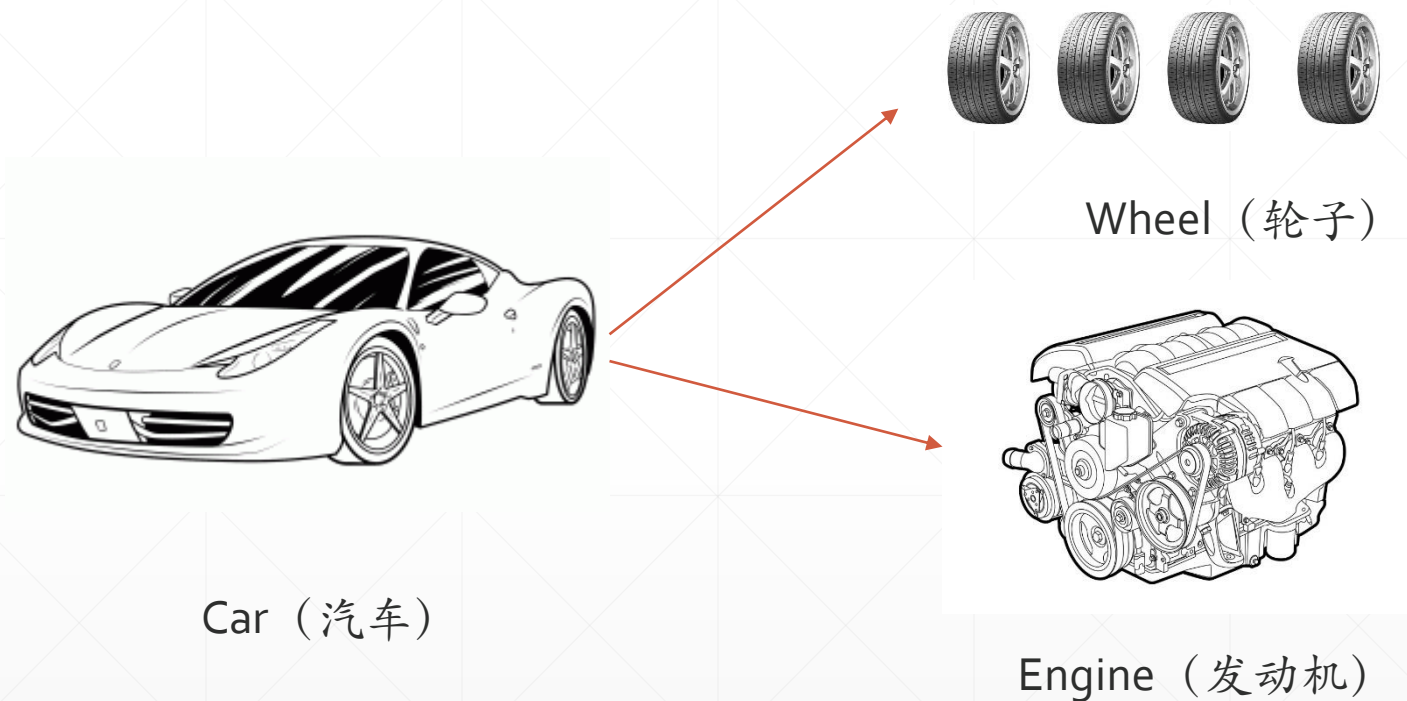
编写程序实现以下功能：



要求：

用户点击“>>>”或“<<<”，将当前选中的指定的列表项移到另一个列表控件中

使用对象组合思想为现实事物建模



示例：CarFactory

开发技能训练

一台个人电脑由以下几个主要部件组成：
CPU、内存、硬盘、主板、机箱、电源、显示器、键盘、鼠标等。

请为每个部件设计相应的类，并编程模拟出以下场景：

使用各个部件“组装”出一台个人电脑。

对象组合的特殊形式——自引用类

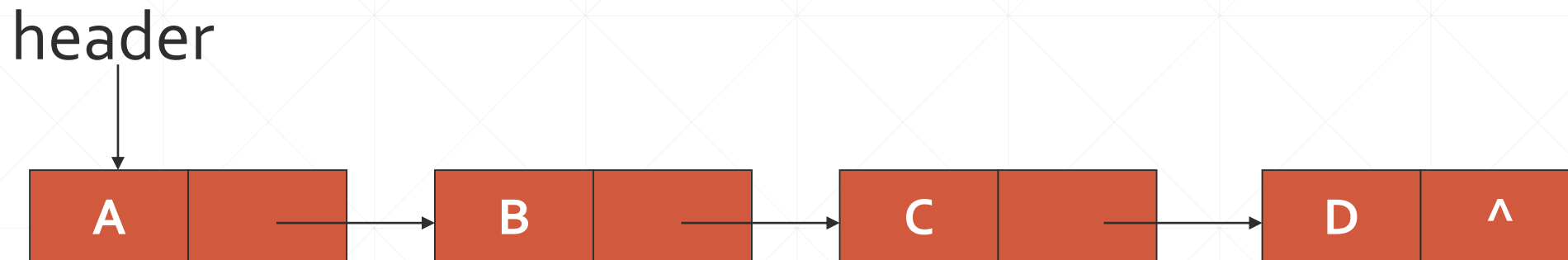
在MyClass类的内部，再定义一个MyClass的字段或属性

```
class MyClass  
{  
    MyClass obj;  
}
```

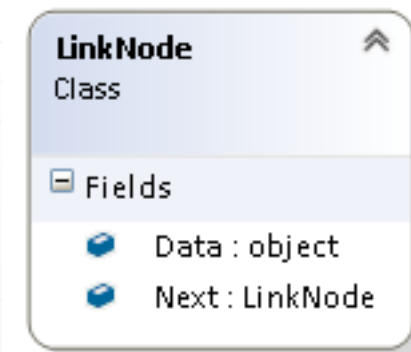
利用obj字段引用下一个MyClass对象，可以在多个MyClass对象之间建立关联关系。它可以用来模拟现实生活中诸如“火车”这样“连成一串”的事物。

面向对象编程语言中，自引用类通常用于建立数据之间的关联。

实例：使用自引用类实现链表



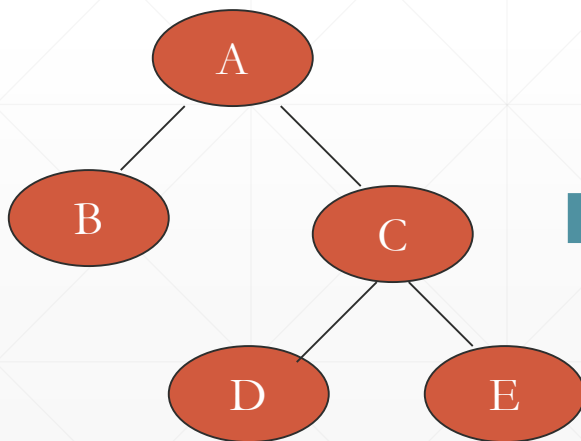
“链表 (Link List)” 是软件开发中一种常用的数据结构，每个节点都有一个可以保存数据的字段，另外有一个指向下一个关联数据的引用。



示例：LinkedListExample

动手编程 - “树”的设计

使用自引用类实现一个二叉树（一个父节点带两个子节点，有左右之分），并且对此树进行深度前序遍历（即从根节点出发，先访问根节点，如果有左右节点，先访问左节点，再访问右节点，对每个节点重复此过程，直到所有节点访问完毕，遍历结束）。



前序遍历结果：

ABCDE