# Local Inversion of Control

namespace Train

{

public static class Extension

{

public static void AddTo<T>(this T self, ICollection<T> coll)

{

coll.Add(self);

}

public static bool IsOneOf1<T>(this T self, ICollection<T> coll)

{

return coll.Contains(self);

}

public static TOut With<TIn, TOut>(this TIn self, Func<TIn,TOut> f,TOut FailValue = null)

where TIn:class

where TOut:class

{

return self == null ? null : f(self);

}

}

class Person

{

public Adress Adress { set; get; }

}

public class Adress

{

public string adress { set; get; }

}

class Program

{

static void Main(string[] args)

{

List<int> list = new List<int>();

5.AddTo(list);

7.AddTo(list);

Console.WriteLine(5.IsOneOf1(list));

Console.WriteLine(list[0]);

Person p = new Person();

p.Adress = new Adress();

string str = p.With(x => x.Adress).With(x => x.adress, "azaza");

Console.WriteLine(str);

}

}