## 复合的from子句

如果需要根据对象的一个成员进行过滤，而该成员本身是一个**集合**，就可以使用复合的 from 子句。

Racer 类定义了一个属性 Cars， Cars  是一个字符串数组。

要过滤驾驶 Ferrari  的所有冠军， 可以使用如下所示的 LINQ 查询。

第一个 from  子句访问从 Formyla1.GetChampions()返回的Racer  对象，

第二个 from  子句访问 Racer 类的属性 Cars，返回**所有的**string  类型的赛车。

接着在 Where  子句中使用这些赛车过滤驾驶 Ferrari 的所有车手

并使用LastName升序

private static void CompoundFrom()

{

var fd = from r in Formula1.GetChampions()

from c in r.Cars

where c == "Ferrari"

orderby r.LastName

select r.FirstName + " " + r.LastName;

foreach (var racer in fd)

{

Console.WriteLine(racer);

}

}

结果把Cas中带有Ferrari 的9个车手名字列了出来：

C#编译器把复合的from子句和LINQ查询转换为SelectMany扩展方法。SelectMany方法用于迭代序列的序列。

[IMG_258](https://www.cnblogs.com/z888/p/javascript:void(0);)

var ferrariDrivers = Formula1.GetChampions().

SelectMany( r => r.Cars ,

(r,c) => new { Racer = r, Car = c }).

Where(r => r.Car == "Ferrari").

OrderBy(r => r.Racer.LastName).

Select(r => r.Racer.FirstName + " " + r.Racer.LastName);