

LINQ —— 查询操作

简介

- LINQ不但提供了基本查询表达式,
- 而且还提供了数十个查询操作:
筛选操作、投影操作、集合操作、聚合操作等。

通过这些操作, 用户更加方便、快捷操作序列, 并对序列实现筛选、投影、排序、聚合、联接等功能

筛选操作Where

- 筛选操作Where能够处理由逻辑运算符（如逻辑“与”、逻辑“或”）组成的逻辑表达式，并从数据源中筛选数据，它和where子句的功能相似。
- Enumerable类的Where()方法的原型如下
 - `public static IEnumerable<TSource> Where<TSource>(this IEnumerable<TSource> source, Func<TSource, bool> predicate)`
- 其中，source参数表示被筛选的数据源，该数据源的类型为IEnumerable<T>。predicate参数表示筛选函数，该函数指定元素满足的条件。该函数返回数据类型为IEnumerable<T>的序列。

投影操作

- 在LINQ中，投影操作和SQL语句中的SELECT子句功能相似，它能够选择数据源中的元素，并指定元素的表现形式。
- 投影操作包括以下2种操作
 - Select操作，将数据源中的元素投影到新序列中，并指定元素的类型和表现形式。
 - SelectMany操作，也可以将数据源中的元素投影到新序列中，并指定元素的类型和表现形式。但是，SelectMany操作可以将一个函数应用到多个序列之上，并将结果合并为一个序列。

排序操作

- 在LINQ中，排序操作可以按照一个或多个关键字对序列中的元素进行排序。其中，第一个排序关键字为主要关键字，第二个排序关键字为次要关键字。
- 排序操作共包括以下5个操作
 - OrderBy操作，根据关键字对序列中的元素按升序排列。
 - OrderByDescending操作，根据关键字对序列中的元素按降序排列。
 - ThenBy操作，根据次要关键字对序列中的元素按升序排列。
 - ThenByDescending操作，根据次要关键字对序列中的元素按降序排列。
 - Reverse操作，将序列中的元素的顺序进行反转。

聚合操作

- 在LINQ中，聚合操作是指根据集合计算一个单个值，如集合的最大值、最小值、平均值等，包括以下7种操作。
 - Count操作，计算序列中元素的数量，或者计算序列满足一定条件的元素的数量。
 - Sum操作，计算序列中元素的和。
 - Max操作，计算序列中元素的最大值。
 - Min操作，计算序列中元素的最小值。
 - Average操作，计算序列中元素的平均值。
 - Aggregate操作，对集合中的元素进行自定义的聚合计算。
 - LongCount操作，计算序列中元素的数量，或者计算序列满足一定条件的元素的数量。一般计算大型集合中的元素的数量。

集合操作

- 在LINQ中，集合操作是指对一个序列或多个序列本身的操作，如去掉重复元素、计算两个集合的交集等操作
- 包括以下4种操作。
 - Distinct操作，可以去掉将数据源中重复的元素，并返回一个新序列。另外，它还可以指定一个比较器来比较两个元素是否相同。
 - Except操作，可以计算两个集合的差集（在一个集合中而不在另外一个集合中元素组成的集合）。
 - Intersect操作，可以计算两个集合的交集（由既在一个集合中，又在另外一个集合中的元素组成的集合）。
 - Union操作，可以计算两个集合的并集（由在一个集合中，或者在另外一个集合中的元素组成的集合）。

元素操作

- 在LINQ中，元素操作可以获取序列中一个特定的元素
- 包括以下8种操作
 - ElementAt操作，返回集合中指定索引处的元素。
 - ElementAtOrDefault操作，返回集合中指定索引处的元素。如果索引超出集合的返回，则返回默认值。
 - First操作，返回集合的第一个元素，或者返回集合的满足指定条件的第一个元素。
 - FirstOrDefault操作，返回集合的第一个元素，或者返回集合的满足指定条件的第一个元素。如果不存在满足该条件的元素，则返回默认元素。
 - Last操作，返回集合的最后一个元素，或者返回集合的满足指定条件的最后一个元素。
 - LastOrDefault操作，返回集合的最后一个元素，或者返回集合的满足指定条件的最后一个元素。如果不存在满足该条件的元素，则返回默认元素。
 - Single操作，返回集合的惟一元素，或者返回集合的满足指定条件的惟一元素。
 - SingleOrDefault操作，返回集合的惟一元素，或者返回集合的满足指定条件的惟一元素。如果不存在满足该条件的元素，则返回默认元素。

数据类型转换操作

- 在LINQ中，数据类型转换操作可以将数据源的类型或者其元素的类型转换为用户指定的类型，包括以下8种操作。
 - AsEnumerable操作，可以将数据源转换为IEnumerable<T>类型的序列。
 - AsQueryable操作，可以将数据源转换为IQueryable<T>或者IQueryable类型的序列。
 - Cast操作，将序列中的元素的类型转换为指定的类型（由TResult参数指定）。
 - OfType操作，从序列中筛选指定类型的元素，并构建为一个序列。
 - ToList操作，将IEnumerable<T>类型的序列转换为List<T>类型的序列。
 - ToArray操作，将IEnumerable<T>类型的序列转换为T[]类型的数组。
 - ToDictionary操作，按照键值将序列中的元素放入一对一的字典序列（Dictionary<TKey,TValue>）中。
 - ToLookup操作，按照键值将序列中的元素放入一对多的字典序列（Lookup<TKey,TValue>）中。

生成操作

- 在LINQ中，生成操作能够产生指定的新序列
- 包括以下4种操作。
 - DefaultIfEmpty操作，返回IEnumerable<T>类型的序列。如果序列为空，则返回只包含一个元素（值为默认值或者指定的值）的序列。
 - Empty操作，返回IEnumerable<T>类型的空序列。
 - Range操作，返回指定范围的数字序列。
 - Repeat操作，返回IEnumerable<T>类型的包含一个重复值的序列。

限定符操作

- 在LINQ中，限定符操作可以检测序列中是否存在满足指定条件的元素，或者检测序列中的所有元素满足指定的条件。限定符操作返回一个布尔值，包含以下3种操作
 - All操作，检测序列中的所有元素是否都满足指定的条件。如果满足，则返回true，否则返回false。
 - Any操作，检测序列中是否存在满足指定条件的元素。如果存在这样的元素，则返回true，否则返回false。
 - Contains操作，检测序列中是否存在指定的元素。如果存在这样的元素，则返回true，否则返回false。

数据分区操作

- 在LINQ中，数据分区操作可以在不改变序列元素之间的顺序条件下，将序列的元素划分为两个部分，把其中一个部分的元素构建为一个新序列并返回。数据分区操作包括以下4个操作。
 - Skip操作，跳过数据源（序列）中指定数量的元素，然后返回由数据源（序列）剩余的元素组成的序列。
 - SkipWhile操作，跳过数据源（序列）中满足指定条件的元素，然后返回由数据源（序列）剩余的元素组成的序列。
 - Take操作，从数据源（序列）开头开始提取指定数量的元素，然后返回由这些元素组成的序列。
 - TakeWhile操作，从数据源（序列）的开头开始提取满足指定条件的元素，然后返回由这些元素组成的序列。

联接操作

- 熟悉SQL语句的读者，可能了解SQL中的联接操作。LINQ中的联接操作和SQL中的联接操作比较相似。SQL语句提供3种联接方式：INNER JOIN、LEFT JOIN和RIGHT JOIN，如图5.6所示。然而，LINQ只提供了两种联接：JOIN和GroupJoin。这两种联接都属于相等联接，即根据两个数据源的键是否相等来匹配这两个数据源的联接，具体说明如下。
 - Join联接，要求元素的联接关系必须同时满足被联接的两个数据源，和SQL语句中的INNER JOIN子句相似。
 - GroupJoin联接，它产生分层数据结构，将第一个集合中的每个元素与第二个集合中的一组相关元素进行匹配。在查询结果中，第一个集合中的元素都会出现在查询结果中。如果第一个集合中的元素在第二个集合中找到相关元素，则使用被找到的元素，否则使用空。

相等操作SequenceEqual

- 给定两个序列，如果这两个序列相等，则必须满足以下两个条件。
 - 序列元素的数量相等，即序列的长度相等。
 - 两个序列的对应元素相等。
- SequenceEqual操作可以判断两个序列是否相等。
Enumerable类的SequenceEqual()方法的原型如下。
 - (1) `public static bool SequenceEqual<TSource>(this IEnumerable<TSource> first, IEnumerable<TSource> second)`
 - (2) `public static bool SequenceEqual<TSource>(this IEnumerable<TSource> first, IEnumerable<TSource> second, IEqualityComparer<TSource> comparer)`
- 其中，first和second参数表示被比较的两个序列，comparer参数表示比较两个元素相等的函数。TSource指定序列元素的类型。

串联操作Contact

- 序列的串联操作是指将一个序列的元素全部追加到另一个序列中，并构成一个新的序列。序列的串联操作由Contact操作实现。Enumerable类的Contact()方法的原型如下。
 - `public static IEnumerable<TSource> Concat<TSource>(this IEnumerable<TSource> first, IEnumerable<TSource> second)`
- 其中，first和second参数表示被串联的两个序列，TSource指定序列元素的类型。

总结

序号	查询操作	查询表达式 (C#)	说明
1	Aggregate	-	自定义的聚合计算。
2	All	-	检测序列中的所有元素是否都满足指定的条件。
3	Any	-	检测序列中是否存在满足指定条件的元素。
4	Average	-	计算序列中元素的平均值。
5	Cast	-	将序列中元素的类型转换为指定的类型（由TResult参数指定）。
6	Contact	-	将一个序列的元素全部追加到另一个序列中，并构成一个新的序列。
7	Contains	-	检测序列中是否存在指定的元素。
8	Count	-	计算序列中元素的数量，或者计算序列满足一定条件的元素的数量。
9	DefaultIfEmpty	-	返回IEnumerable<T>类型的序列。如果序列为空，则返回只包含一个元素（值为默认值或者指定的值）的序列。
10	Distinct	-	可以去掉将数据源中重复的元素，并返回一个新序列。另外，它还可以指定一个比较器来比较两个元素是否相同。
11	Element	-	返回集合中指定索引处的元素。
12	ElementAtOrDefault	-	返回集合中指定索引处的元素。如果索引超出集合的返回，则返回默认值。
13	Empty	-	返回IEnumerable<T>类型的空序列。
14	EqualAll/SequenceEqual	-	判断两个序列是否相等。
15	Except	--	可以计算两个集合的差集（由在一个集合中而不在另外一个集合中的元素组成的集合）。
16	First	-	返回集合的第一个元素，或者返回集合的满足指定条件的第一个元素。
17	FirstOrDefault	-	返回集合的第一个元素，或者返回集合的满足指定条件的第一个元素。如果不存在满足该条件的元素，则返回默认元素。

总结

18	GroupBy	group子句	对序列中的元素进行分组。
19	GroupJoin	join子句	它产生分层数据结构，将第一个集合中的每个元素与第二个集合中的一组相关元素进行匹配。在查询结果中，第一个集合中的元素都会出现在查询结果中。如果第一个集合中的元素在第二个集合中找到相关元素，则使用被找到的元素，否则使用空。
20	Intersect	-	可以计算两个集合的交集（由既在一个集合中，又在另外一个集合中的元素组成的集合）。
21	Join	join子句	要求元素的联接关系必须同时满足被联接的两个数据源，和SQL语句中的INNER JOIN子句相似。
22	Last	-	返回集合的最后一个元素，或者返回集合的满足指定条件的最后一个元素。
23	LastOrDefault	-	返回集合的最后一个元素，或者返回集合的满足指定条件的最后一个元素。如果不存在满足该条件的元素，则返回默认元素。
24	LongCount	-	计算序列中元素的数量，或者计算序列满足一定条件的元素的数量。一般计算大型集合中的元素的数量。
25	Max	-	计算序列中元素的最大值。
26	Min	-	计算序列中元素的最小值。
27	OfType	-	从序列中筛选指定类型的元素，并构建为一个序列。
28	OrderBy	orderby子句	根据关键字对序列中的元素按升序排列。
29	OrderByDescending	orderby子句	根据关键字对序列中的元素按降序排列。
30	Range	-	返回指定范围的数字序列。
31	Repeat	-	返回IEnumerable<T>类型的包含一个重复值的序列。
32	Reverse	-	将序列中的元素的顺序进行反转。
33	Select	select子句	将数据源中的元素投影到新序列中，并指定元素的类型和表现形式。
34	SelectMany	多个from子句	也可以将数据源中的元素投影到新序列中，并指定元素的类型和表现形式。但是，SelectMany操作可以将一个函数应用到多个序列之上，并将结果合并为一个序列。

总结

35	Single	-	返回集合的惟一元素，或者返回集合满足指定条件的惟一元素。
36	SingleOrDefault	-	返回集合的惟一元素，或者返回集合的满足指定条件的惟一元素。如果不存在满足该条件的元素，则返回默认元素。
37	Skip	-	跳过数据源（序列）中指定数量的元素，然后返回由数据源（序列）剩余的元素组成的序列。
38	SkipWhile	-	跳过数据源（序列）中满足指定条件的元素，然后返回由数据源（序列）剩余的元素组成的序列。
39	Sum	-	计算序列中元素的和。
40	Take	-	从数据源（序列）的开头开始提取指定数量的元素，然后返回由这些元素组成的序列。
41	TakeWhile	-	从数据源（序列）的开头开始提取满足指定条件的元素，然后返回由这些元素组成的序列。
42	ThenBy	orderby子句	根据次要关键字对序列中的元素按升序排列。
43	ThenByDescending	orderby子句	根据次要关键字对序列中的元素按降序排列。
44	ToArray	-	将IEnumerable<T>类型的序列转换为T[]类型的数组。
45	ToDictionary	-	按照键值将序列中的元素放入一对一的字典序列（Dictionary<TKey,TValue>）中。
46	ToList	-	将IEnumerable<T>类型的序列转换为List<T>类型的序列。
47	ToLookup	-	按照键值将序列中的元素放入一对多的字典序列（Lookup<TKey,TValue>）中。
48	Union	-	可以计算两个集合的并集（由在一个集合中，或者在另外一个集合中的元素组成的集合）。
49	Where	where子句	处理由逻辑运算符（如逻辑“与”、逻辑“或”）组成的逻辑表达式，并从数据源中筛选数据。