**Практическая работа №7**

**Технология LINQ**

**Знакомство с запросами LINQ**

При вводе (выводе) последовательности вначале следует ввести (соответственно, вывести) ее размер, а затем ее элементы. Все входные последовательности являются непустыми. Выходные последовательности могут быть пустыми; в этом случае требуется вывести единственное число 0 – размер данной последовательности.

Если в задании идет речь о *порядковых номерах* элементов последовательности, то предполагается, что нумерация ведется от 1 (таким образом, порядковый номер элемента равен *индексу* этого элемента, *увеличенному на 1*).

Для обработки входной последовательности в большинстве заданий достаточно указать *единственный* оператор, содержащий вызовы нужных запросов LINQ to Objects и другие необходимые конструкции, в частности, операцию ?? языка C#.

При выполнении заданий с использованием задачника Programming Taskbook можно использовать дополнительные методы, определенные в задачнике:

• методы GetEnumerableInt и GetEnumerableString обеспечивают ввод исходных последовательностей с элементами целого и строкового типа соответственно (выполняется ввод размера последовательности и всех ее элементов, возвращается введенная последовательность);

• метод Put является методом расширения для последовательности и обеспечивает вывод этой последовательности (выводится размер последовательности и все ее элементы);

• метод Show также является методом расширения для последовательности; он обеспечивает печать последовательности в разделе отладки окна задачника и возвращает эту же последовательность (отладочная печать может сопровождаться *комментарием*, который указывается в качестве необязательного строкового параметра метода Show).

Использование вспомогательных методов иллюстрируется приведенным ниже фрагментом программы, решающей следующую задачу: извлечь из исходной целочисленной последовательности четные отрицательные числа и заменить порядок их следования на обратный.

// Ввод исходных данных

var a = GetEnumerableInt();

// Обработка

var res = a.Where(e => e % 2 == 0 && e < 0)

 .Reverse();

// Вывод результатов

res.Put();

Все этапы решения можно объединить в одном операторе, состоящем из цепочки последовательно вызываемых методов:

GetEnumerableInt()

 .Where(e => e % 2 == 0 && e < 0)

 .Reverse().Put();

Возможен вариант решения, в котором дополнительно выполняется отладочная печать (в данном случае полученная последовательность четных отрицательных чисел печатается перед изменением порядка следования ее элементов и после этого изменения):

GetEnumerableInt().Where(e => e % 2 == 0 && e < 0)

 .Show().Reverse().Show().Put();

Отладочная печать позволяет увидеть состояние последовательности на различных этапах ее преобразования и тем самым облегчает поиск ошибок.

**ЗАДАЧА 1**

***Поэлементные операции, агрегирование и генерирование последовательностей***

Изучаемые методы LINQ:

• First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault (поэлементные операции);

• Count, Sum, Average, Max, Min, Aggregate (агрегирование);

• Range (генерирование последовательностей).

**Варианты заданий:**

1. Дана цифра D (однозначное целое число) и целочисленная последовательность A. Вывести первый положительный элемент последовательности A, оканчивающийся цифрой D. Если требуемых элементов в последовательности A нет, то вывести 0.

2. Дано целое число L (> 0) и строковая последовательность A. Вывести последнюю строку из A, начинающуюся с цифры и имеющую длину L. Если требуемых строк в последовательности A нет, то вывести строку «Not found».

***Указание.*** *Для обработки ситуации, связанной с отсутствием требуемых строк, использовать операцию ??.*

3. Дан символ С и строковая последовательность A. Если A содержит единственный элемент, оканчивающийся символом C, то вывести этот элемент; если требуемых строк в A нет, то вывести пустую строку; если требуемых строк больше одной, то вывести строку «Error».

***Указание.*** *Использовать try-блок для перехвата возможного исключения.*

4. Дан символ С и строковая последовательность A. Найти количество элементов A, которые содержат более одного символа и при этом начинаются и оканчиваются символом C.

5. Дана строковая последовательность. Найти сумму длин всех строк, входящих в данную последовательность.

6. Дана целочисленная последовательность. Найти количество ее отрицательных элементов, а также их сумму. Если отрицательные элементы отсутствуют, то дважды вывести 0.

7. Дана целочисленная последовательность. Найти количество ее положительных двузначных элементов, а также их среднее арифметическое (как вещественное число). Если требуемые элементы отсутствуют, то дважды вывести 0 (первый раз как целое, второй – как вещественное).

8. Дана целочисленная последовательность. Вывести ее минимальный положительный элемент или число 0, если последовательность не содержит положительных элементов.

9. Дано целое число L (> 0) и строковая последовательность A. Строки последовательности A содержат только заглавные буквы латинского алфавита. Среди всех строк из A, имеющих длину L, найти наибольшую (в смысле лексикографического порядка). Вывести эту строку или пустую строку, если последовательность не содержит строк длины L.

10. Дана последовательность непустых строк. Используя метод Aggregate, получить строку, состоящую из начальных символов всех строк исходной последовательности.

**ЗАДАЧА 2**

***Фильтрация, сортировка, теоретико- множественные операции***

Изучаемые методы LINQ:

• Where, TakeWhile, SkipWhile, Take, Skip (фильтрация);

• OrderBy, OrderByDescending, ThenBy, ThenByDescending (сортировка);

• Distinct, Reverse (удаление повторяющихся элементов и инвертирование);

• Union, Intersect, Except (теоретико-множественные операции).

**Варианты заданий:**

1. Дана целочисленная последовательность. Извлечь из нее все положительные числа, сохранив их исходный порядок следования.

2. Дана целочисленная последовательность. Извлечь из нее все нечетные числа, сохранив их исходный порядок следования и удалив все вхождения повторяющихся элементов, кроме первых.

3. Дана целочисленная последовательность. Извлечь из нее все четные отрицательные числа, поменяв порядок извлеченных чисел на обратный.

4. Дана цифра D (целое однозначное число) и целочисленная последовательность A. Извлечь из A все различные положительные числа, оканчивающиеся цифрой D (в исходном порядке). При наличии повторяющихся элементов удалять все их вхождения, кроме последних.

***Указание.*** *Последовательно применить методы Reverse, Distinct, Reverse.*

5. Дана целочисленная последовательность. Извлечь из нее все положительные двузначные числа, отсортировав их по возрастанию.

6. Дана строковая последовательность. Строки последовательности содержат только заглавные буквы латинского алфавита. Отсортировать последовательность по возрастанию длин строк, а строки одинаковой длины – в лексикографическом порядке по убыванию.

7. Дано целое число K (> 0) и строковая последовательность A. Строки последовательности содержат только цифры и заглавные буквы латинского алфавита. Извлечь из A все строки длины K, оканчивающиеся цифрой, отсортировав их в лексикографическом порядке по возрастанию.

8. Дано целое число K (> 0) и целочисленная последовательность A. Начиная с элемента A с порядковым номером K, извлечь из A все нечетные двузначные числа, отсортировав их по убыванию.

9. Дано целое число K (> 0) и строковая последовательность A. Из элементов A, предшествующих элементу с порядковым номером K, извлечь те строки, которые имеют нечетную длину и начинаются с заглавной латинской буквы, изменив порядок следования извлеченных строк на обратный.

10. Даны целые числа K1 и K2 и целочисленная последовательность A; 1 ≤ K1 < K2 ≤ N, где N – размер последовательности A. Найти сумму положительных элементов последовательности с порядковыми номерами от K1 до K2 включительно.