

## Лабораторная работа № 7

### “Разработка и исследование программы с использованием LINQ-запросов”

Постановка задачи: разработать программу для обработки последовательностей с использованием LINQ-запросов.

Цель: изучение, использование и исследование языка интегрированных запросов LINQ to Objects.

Проект должен содержать четыре формы:

- 1-я форма – для задания и редактирования значений и параметров последовательности объектов,
- 2-я форма – для визуализации результатов обработки последовательностей,
- 3-я форма – для задания и редактирования значений и параметров последовательности объектов и сравнительной оценки быстродействия методов,
- 4-я форма – для сведений об авторе проекта (модальная форма, которая может вызываться из первых двух форм).

UML диаграмма вариантов использования проекта приведена на рисунке 1.

С целью решения задачи с использованием LINQ реализовать на Visual C# проект, для чего:

1. Используя компоненты *Form*, *TextBox*, *Label*, *Button*, *Panel*, *NumericUpDown*, *DataGridView*, *RadioButton*, *ComboBox*, *GroupBox*, *Chart* создать формы проекта.
2. В соответствии с вариантом (таблица 1) создать класс и последовательность объектов этого класса (число элементов в последовательности – от 10 до 15).
3. Выполнить сортировку элементов последовательности по разным полям объектов – 3-4 различных варианта сортировки.
4. Выполнить фильтрацию элементов последовательности – 4-5 различных вариантов фильтрации.
5. Выполнить группировку элементов последовательности по разным полям объектов – 2-3 различных варианта группировки.
6. Выполнить с помощью методов агрегирования итоговые вычисления над последовательностью – 3-4 различных варианта вычислений.
7. Выполнить сравнительную оценку быстродействия LINQ-методов (последовательный и параллельный (PLINQ) варианты) в соответствии с вариантом (таблица 2) и 2-х произвольных аналогичных методов языка C#. Оценку быстродействия выполнить при 7-10 различных размерностях последовательности. При минимальной размерности последовательности

оценка времени выполнения метода должна быть стабильной. Результаты сравнительной оценки быстродействия представить в виде таблицы и графика (с использованием компонента *Chart*).

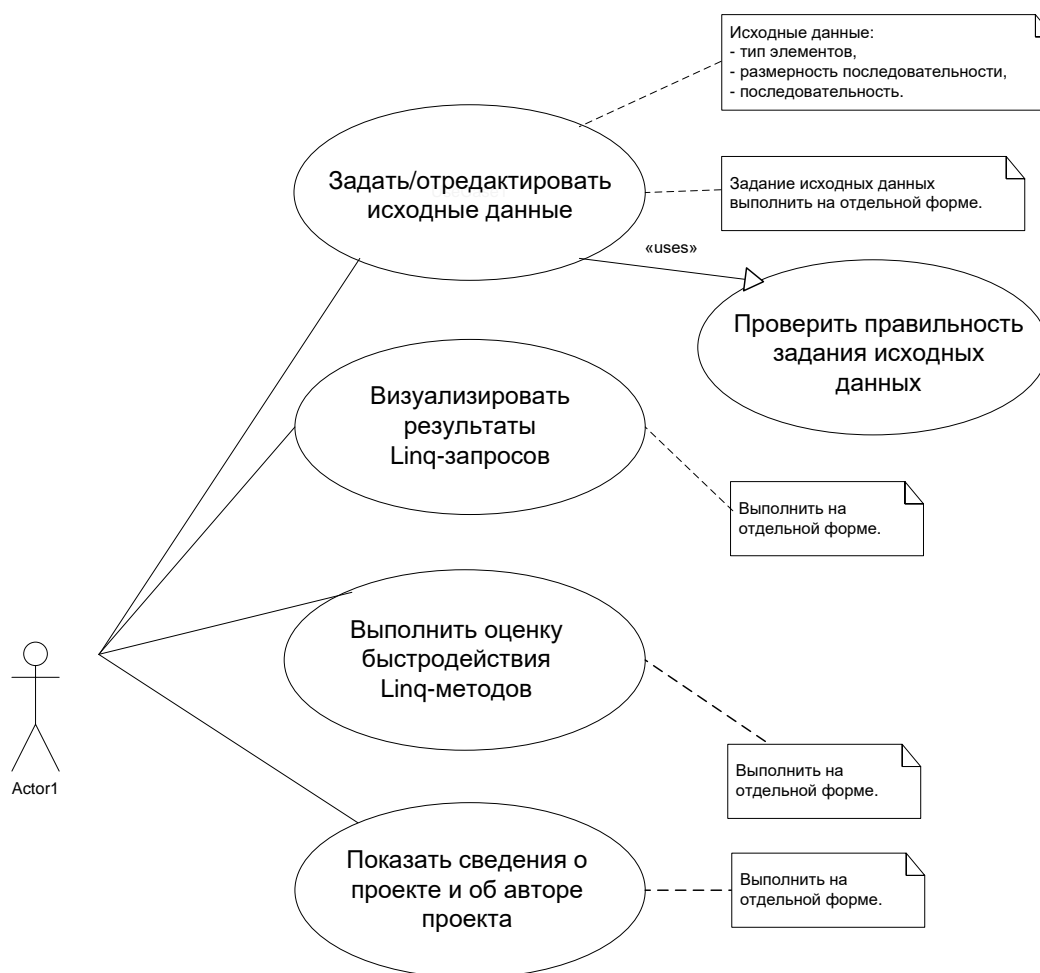


Рис. 1. Диаграмма вариантов использования проекта  
“Linq-запросы”

Таблица 1 – Варианты классов

Номер по списку	Тип последовательности	Имя класса	Поля класса
1 - 6	массив	Студент	- уникальный идентификатор, - фамилия, - имя, - отчество, - дата рождения, - группа, - средний балл, - стипендия
7 - 13	коллекция	Преподаватель	- уникальный идентификатор, - фамилия, - имя, - отчество, - дата рождения, - кафедра, -

			должность, - стаж работы, - зарплата
14 - 20	массив	Программист	- уникальный идентификатор, - фамилия, - имя, - отчество, - дата рождения, - язык программирования, - должность, - стаж работы, - зарплата
21 - 26	коллекция	Тестировщик	- уникальный идентификатор, - фамилия, - имя, - отчество, - дата рождения, - технология тестирования, - должность, - стаж работы, - зарплата

Таблица 2 – Варианты для оценки быстродействия LINQ-методов

Номер по списку	Тип последовательности	Тип элементов	Метод
1 - 2	массив	Int	Сортировка
3 - 4	коллекция	Double	Фильтрация
5 - 6	массив	Byte	Поиск среднего значения
7 - 8	коллекция	String	Сортировка
9 - 10	массив	String	Фильтрация
11 - 12	коллекция	Byte	Поиск дисперсии
13 - 14	массив	Double	Поиск минимума
15 - 16	коллекция	Int	Поиск максимума
17 - 18	массив	Int	Фильтрация
19 - 20	коллекция	Double	Поиск дисперсии
21 - 22	массив	Byte	Фильтрация
23 - 24	коллекция	String	Поиск среднего числа букв
25 - 26	массив	String	Фильтрация

Письменный отчет по лабораторной работе должен содержать:

1. Титульный лист. (Содержащий – название лабораторной работы, фамилию, имя, отчество, номер группы исполнителя, дату сдачи.)
2. Постановку задачи.
3. Диаграммы классов для всех используемых в проекте классов.
4. Таблицу со списком полей, свойств и методов классов проекта и их назначением.

Таблица 1 – Поля, свойства и методы класса А и их назначение (Пример)

№	Поле	Назначение
1		
	...	...
	Свойство	
	...	
	Метод	
	...	...

5. Таблицу со списком обработчиков событий проекта и их назначением.

Таблица 2 – Обработчики событий проекта и их назначение (Пример)

№	Обработчик события	Назначение
1.	...	...

6. Внешний вид форм проекта в режиме выполнения.

7. Раздел “Сортировка” – привести исходные данные, код метода операции, результат операции.

8. Раздел “Фильтрация” – привести исходные данные, код метода операции, результат операции.

9. Раздел “Группировка” – привести исходные данные, код метода операции, результат операции.

10. Раздел “Агрегирование” – привести исходные данные, код метода операции, результат операции.

11. Раздел “Сравнительная оценка быстродействия LINQ-методов”, содержащий следующие материалы: - код LINQ-метода для сравнения, - код PLINQ-метода для сравнения, - код 1-го метода на языке C# для сравнения, - код 2-го метода на языке C# для сравнения, - таблица, - график (с использованием компонента *Chart*).

Таблица 3 – Сравнительная оценка быстродействия методов (в ms) (Пример)

	n <sub>1</sub>	n <sub>2</sub>	n <sub>3</sub>	n <sub>4</sub>	n <sub>5</sub>	n <sub>6</sub>	n <sub>7</sub>
LINQ-метод							
PLINQ-метод							
Метод1							
Метод2							

12. Выводы по лабораторной работе (в выводах отразить ограничения на работу проекта, пути дальнейшей модернизации проекта, результаты сравнительной оценки быстродействия LINQ- и PLINQ-методов, характеристики компьютера, на котором выполнялась оценка быстродействия).

13. Распечатку кода обработчиков событий проектов и классов (*обязательны комментарии*).

### **Вопросы по LINQ to Objects**

1. Как переводится аббревиатура LINQ?
2. Какие разновидности LINQ-языка существуют?
3. Какие источники данных совместимы с LINQ?
4. Какой интерфейс должен поддерживаться совместимыми с LINQ источниками данных?
5. Какие средства предоставляются для перебора элементов в коллекции?
6. Какие синтаксисы используются при формировании LINQ-запросов?
7. Что может возвращать оператор select?
8. Каковы особенности использования анонимного типа в операторе select?
9. Какими способами может быть сформирован LINQ-запрос?
10. Каковы особенности создания методов расширения (методов запроса) и работы с ними?
11. Какие методы поддерживают операцию “Фильтрация” ?
12. Какие методы поддерживают операцию “Проецирование” ?
13. Какие методы поддерживают операцию “Соединение” ?
14. Какие методы поддерживают операцию “Упорядочение” ?
15. Какие методы поддерживают операцию “Группирование” ?
16. Какие методы входят в группу “ Операции над множествами” ?
17. Какие методы поддерживают операцию “Методы преобразования” ?
18. Какие методы поддерживают операцию “Операции элементов” ?
19. Какие методы поддерживают операцию “Методы агрегирования” ?
20. Какие методы входят в группу “Квантификаторы” ?
21. Какие методы входят в группу “Методы генерации” ?
22. Какой обобщенный делегат используется в качестве аргумента в методах запроса?
23. Какие режимы выполнения запросов существуют?
24. Какой тип данных должен иметь источник данных для возможности применения Parallel LINQ?
25. Какие действия выполняются с помощью метода AsParallel()?

26. Как выполняется предохранение порядка результатов в Parallel LINQ?
27. В чем состоит управление параллелизмом в Parallel LINQ?
28. Каковы основные функции технологии LINQ To XML?
29. Каковы особенности вставки элементов в XML –файл в технологии LINQ To XML?
30. Каковы особенности функционального конструирования объектов XML в технологии LINQ To XML?
31. Каковы особенности выборки элементов из Xml-файла в технологии в LINQ To XML?
32. Каковы особенности удаления элементов из Xml-файла в технологии в LINQ To XML?
33. Каковы особенности редактирования элементов в Xml-файле в технологии в LINQ To XML?

#### Задание по Лр 7:

- Выполнить сравнительную оценку быстродействия LINQ-метода в соответствии с вариантом (таблица 2) с использованием Parallel LINQ и без использования Parallel LINQ и 2-х произвольных аналогичных методов языка C#. Оценка быстродействия выполнить при 7-10 различных размерностях последовательности.
- Раздел “Сравнительная оценка быстродействия LINQ-метода”, должен содержать следующие материалы: - код LINQ-метода для сравнения, - код Parallel LINQ-метода для сравнения, - код 1-го метода на языке C# для сравнения, - код 2-го метода на языке C# для сравнения, - таблицы, - графики (с использованием компонента *Chart*).

#### **Примечание:**

1. Формула для вычисления среднего арифметического значения  $Sr$  последовательности из  $n$  элементов

$$Sr = \frac{\sum_{i=0}^{n-1} a_i}{n}$$

2. Формула для вычисления дисперсии  $D$  последовательности из  $n$  элементов

$$D = \frac{\sum_{i=0}^{n-1} (a_i - Sr)^2}{n - 1}$$