Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

Язык LINQ и методы-расширения

тема

Вариант 14

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ А.А. Чикизов

подпись, дата инициалы, фамилия

Студент КИ15–16Б, 031510065 \_\_\_\_\_\_\_\_\_\_\_ Т.В. Радионов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2017

# Задание

Описать класс для работы с одномерным массивом:

Описать класс Note {поля: SurnameName (string); Phone (string); DOB (string), CardIndexNumber (int)}

Описать класс CardIndex {поля: Name (string), Number (int)}

Заполнить список обоих классов (мин 5).

Создать запрос LINQ по типу inner join для классов и отсортировать их по именам.

Реализовать собственный метод-расширения для класса Note, который позволяет вычислить длину строки поля SurnameName.

Переделать запрос: сгруппировать Note в CardIndex, при этом внутри группы сортировать SurnameName (используя собственный метод-расширение) по возрастанию длины строки, само число вывести рядом с остальной информацией.

# Исходный код

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

// Заполняем списки записей и картотек

Console.WriteLine("\nЗаполняем списки записей и картотек\n");

List<Note> notes = Note.GetCompletedList();

for (int i = 0; i < notes.Count; i++)

{

Console.WriteLine(notes[i]);

}

List<CardIndex> cardIndexes = CardIndex.GetCompletedList();

for (int i = 0; i < cardIndexes.Count; i++)

{

Console.WriteLine(cardIndexes[i]);

}

// Связываем записи с картотеками

Console.WriteLine("\nСвязываем записи с картотеками\n");

int counter = 0;

for (int i = 0; i < notes.Count; i++, counter++)

{

if (counter > cardIndexes.Count - 1)

counter = 0;

notes[i].CardIndex = cardIndexes[counter];

Console.WriteLine(notes[i] + " --- связан с картотекой " + notes[i].CardIndexNumber);

}

// Запрос Inner Join

Console.WriteLine("\nЗапрос Inner Join\n");

var innerJoinQuery = from n in notes

join ci in cardIndexes on n.CardIndexNumber equals ci.Number

select new { SurnameName = n.SurnameName, Number = ci.Number };

foreach (var item in innerJoinQuery)

Console.WriteLine("{0} - {1}", item.SurnameName, item.Number);

// Запрос Group Join и использование метода-расширения

Console.WriteLine("\nЗапрос Group Join и использование метода-расширения\n");

var groupJoin = cardIndexes.GroupJoin(

notes,

ci => ci.Number,

n => n.CardIndexNumber,

(cardIndex, nts) => new

{

Name = cardIndex.Name,

Number = cardIndex.Number,

Notes = nts.Select(n => n.SurnameName).OrderBy(s => s.StringLength())

});

foreach (var cardIndex in groupJoin)

{

Console.WriteLine(cardIndex.Name);

foreach (string note in cardIndex.Notes)

{

Console.WriteLine(note + ", длина строки: " + note.StringLength());

}

}

Console.ReadKey();

}

}

class Note

{

public string SurnameName { get; set; }

public string Phone { get; set; }

public string DOB { get; set; }

CardIndex cardIndex = null;

public CardIndex CardIndex

{

set

{

cardIndex = value;

}

}

public int ? CardIndexNumber

{

get

{

if (cardIndex is null)

{

return null;

}

else

{

return cardIndex.Number;

}

}

}

static List<Note> notes = new List<Note>();

/// <summary>

/// Конструктор

/// </summary>

/// <param name="note"></param>

public Note(string surnameName, string phone, string DOB)

{

SurnameName = surnameName;

Phone = phone;

this.DOB = DOB;

}

/// <summary>

/// Заполнение и получение заполненного списка

/// </summary>

public static List<Note> GetCompletedList()

{

notes.Add(new Note("Андрей Андреевич", "+9 (999) 999-99-99)", "11.01.1990"));

notes.Add(new Note("Константин Константинович", "+2 (323) 245-38-75)", "11.01.1994"));

notes.Add(new Note("Тимур Тимурович", "+2 (873) 145-35-84)", "11.01.1995"));

notes.Add(new Note("Федр Федорович", "+2 (982) 244-31-95)", "11.01.1992"));

notes.Add(new Note("Василий Васильевич", "+2 (923) 655-05-23)", "11.01.1991"));

notes.Add(new Note("Петр Петрович", "+1 (456) 123-03-00)", "11.01.1995"));

notes.Add(new Note("Сергей Сергеевич", "+3 (333) 547-47-12)", "11.01.1991"));

notes.Add(new Note("Виталий Витальевич", "+1 (789) 010-11-99)", "11.01.1993"));

notes.Add(new Note("Иван Иванович", "+1 (234) 567-89-00)", "11.01.2001"));

return notes;

}

/// <summary>

/// Перегрузка метода ToString()

/// </summary>

/// <returns></returns>

public override string ToString()

{

return String.Format("Фамилия Имя: {0}; Телефон: {1}; Дата рождения: {2}", SurnameName, Phone, DOB);

}

}

class CardIndex

{

public string Name { get; set; }

public int Number { get; set; }

static List<CardIndex> cardIndexes = new List<CardIndex>();

/// <summary>

/// Конструктор

/// </summary>

/// <param name="name"></param>

/// <param name="number"></param>

public CardIndex(string name, int number)

{

Name = name;

Number = number;

}

/// <summary>

/// Заполнение и получение списка

/// </summary>

public static List<CardIndex> GetCompletedList()

{

cardIndexes.Add(new CardIndex("Картотека 1", 1));

cardIndexes.Add(new CardIndex("Картотека 2", 2));

cardIndexes.Add(new CardIndex("Картотека 3", 3));

cardIndexes.Add(new CardIndex("Картотека 4", 4));

cardIndexes.Add(new CardIndex("Картотека 5", 5));

return cardIndexes;

}

/// <summary>

/// Перегрузка метода ToString()

/// </summary>

/// <returns></returns>

public override string ToString()

{

return String.Format("Название: {0}; Номер: {1}", Name, Number);

}

}

public static class ExtentionMethods

{

/// <summary>

/// Метод-расширение для вычисления длины строки

/// </summary>

/// <param name="s"></param>

/// <returns></returns>

public static int StringLength(this string s)

{

int result = s.Length;

return result;

}

}

}

# Результы

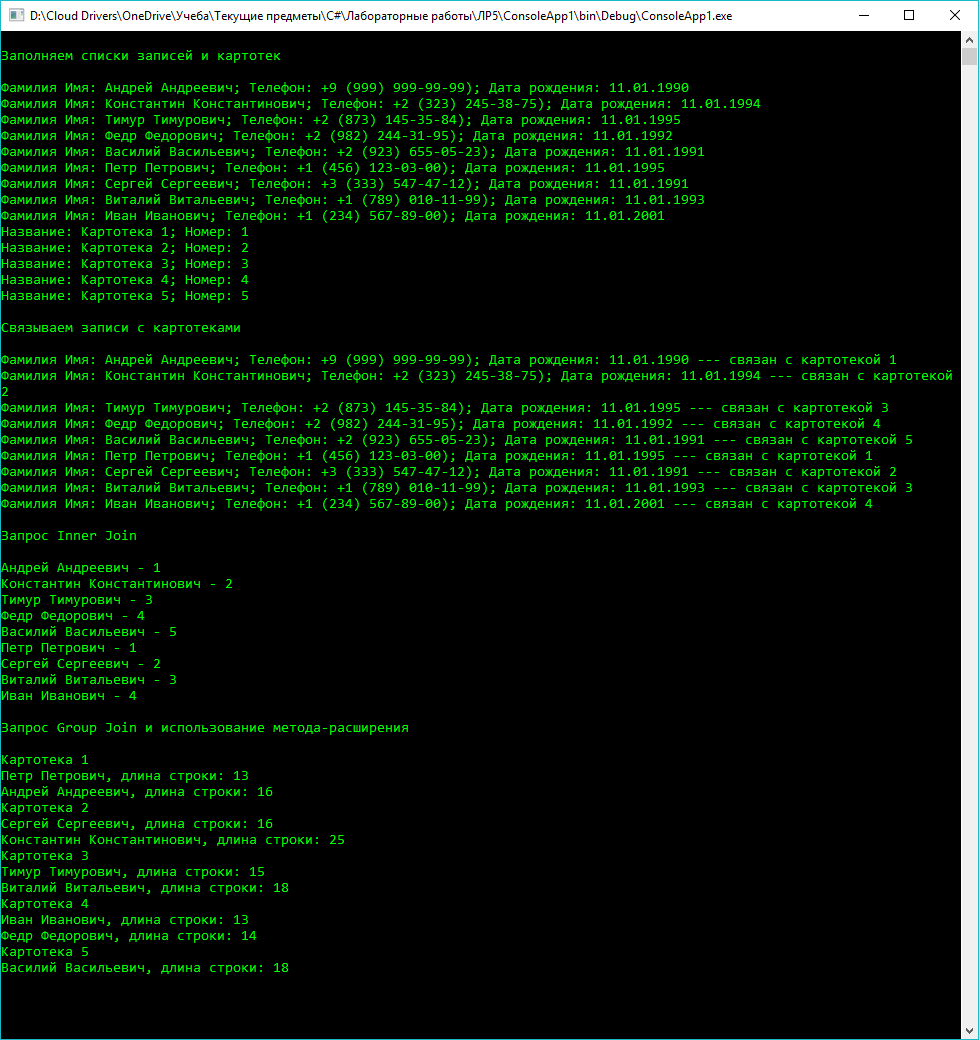


Рисунок 1 – Пример работы программы

ПРИЛОЖЕНИЕ А

Дополнительное задание: выполнить группировку по ключу и применить Key.

Реализация метода в классе Program:

Console.WriteLine("\nЗапрос GroupBy с применением .Key\n");

var groupBy = notes.GroupBy(n => n.CardIndexNumber);

foreach (var groupNotes in groupBy)

{

Console.WriteLine("Картотека: " + groupNotes.Key);

foreach (Note note in groupNotes)

{

Console.WriteLine(note + ", длина строки: " + note.SurnameName.StringLength());

}

}

Console.ReadKey();