**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №4

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32 |  | преподаватель каф. ИУ5 |
| Ким Р.И. |  | Гапанюк Ю.С. |
| Подпись и дата: |  | Подпись и дата: |

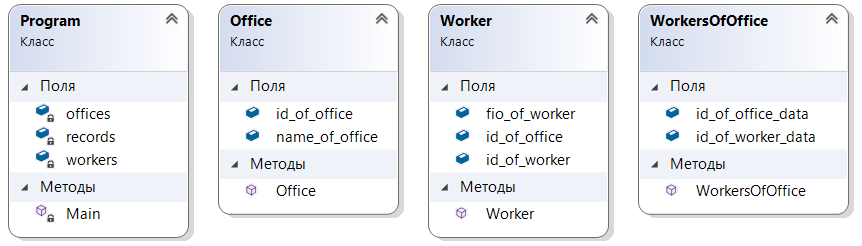
г. Москва, 2018 г.

**Описание задания:**

Разработать программу, реализующую работу с файлами.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. Добавить кнопку, реализующую функцию чтения файла в список слов List<string>.
3. Для выбора имени файла используется класс OpenFileDialog, который открывает диалоговое окно с выбором файла. Ограничить выбор только файлами с расширением «.txt».
4. Для чтения из файла рекомендуется использовать статический метод ReadAllText() класса File (пространство имен System.IO). Содержимое файла считывается методом ReadAllText() в виде одной строки, далее делится на слова с использованием метода Split() класса string. Слова сохраняются в список List<string>.
5. При сохранении слов в список List<string> дубликаты слов не записываются. Для проверки наличия слова в списке используется метод Contains().
6. Вычислить время загрузки и сохранения в список с использованием класса Stopwatch (пространство имен System.Diagnostics). Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).
7. Добавить на форму поле ввода для поиска слова и кнопку поиска. При нажатии на кнопку поиска осуществлять поиск введенного слова в списке. Слово считается найденным, если оно входит в элемент списка как подстрока (метод Contains() класса string).
8. Добавить на форму список (ListBox). Найденные слова выводить в список с использованием метода «название\_списка.Items.Add()». Вызовы метода «название\_списка.Items.Add()» должны находится между вызовами методов «название\_списка.BeginUpdate()» и «название\_списка. EndUpdate()».
9. Вычислить время поиска с использованием класса Stopwatch. Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).

**Диаграмма классов:**



Листинг

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab7

{

class Program

{

static List<Worker> workers = new List<Worker>()

{

new Worker(1,"Алексеев Алексей Алексеевич",1),

new Worker(2,"Фёдоров Фёдор Фёдорович",1),

new Worker(3,"Андреев Андрей Андреевич",2),

new Worker(4,"Александров Александр Александрович",2),

new Worker(5,"Васильев Василий Васильевич",1),

new Worker(6,"Петров Пётр Петрович",3),

new Worker(7,"Иванов Иван Иванович",3)

};

static List<Office> offices = new List<Office>()

{

new Office(1,"Отдел кадров"),

new Office(2,"Отдел маркетинга"),

new Office(3,"Бухгалтерия")

};

static List<WorkersOfOffice> records = new List<WorkersOfOffice>()

{

new WorkersOfOffice(1,1),

new WorkersOfOffice(1,3),

new WorkersOfOffice(2,1),

new WorkersOfOffice(3,2),

new WorkersOfOffice(3,3),

new WorkersOfOffice(4,2),

new WorkersOfOffice(5,1),

new WorkersOfOffice(5,2),

new WorkersOfOffice(5,3),

new WorkersOfOffice(6,2),

new WorkersOfOffice(6,3),

new WorkersOfOffice(7,1),

new WorkersOfOffice(7,3)

};

static void Main(string[] args)

{

//Список всех сотрудников и отделов, отсортированный по отделам

Console.WriteLine("Список всех сотрудников и отделов, отсортированный поотделам");

var q1 = from x in workers

orderby x.id\_of\_office

select x;

foreach (var x in q1)

Console.WriteLine("ID отдела: {0}, ID сотрудника: {1}, ФИО отрудника: {2}",

x.id\_of\_office, x.id\_of\_worker, x.fio\_of\_worker);

//Список всех сотрудников, у которых фамилия начинается с буквы «А»

Console.WriteLine("\nСписок всех сотрудников, у которых фамилия начинается с буквы «А»");

var q2 = from x in workers

where (x.fio\_of\_worker)[0] == 'А'

select x;

foreach (var x in q2)

Console.WriteLine("ID сотрудника: {0}, ФИО сотрудника: {1}, ID отдела: {2}",

x.id\_of\_worker, x.fio\_of\_worker, x.id\_of\_office);

//Список всех отделов и количество сотрудников в каждом отделе

Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе");

var q33 = from x in offices

join y in workers on x.id\_of\_office equals y.id\_of\_office

group x by x.id\_of\_office into g

select new { id = g.Key, count = g.Count() };

foreach (var g in q33) Console.WriteLine("ID отдела: {0} , кол-во сотрудников: {1}", g.id, g.count);

//Список отделов, в которых у всех сотрудников фамилия начинается с буквы «А»

Console.WriteLine("\nСписок отделов, в которых у всех сотрудников фамилия начинается с буквы «А»");

var q4 = from x in workers

group x by x.id\_of\_office into g

where g.All(x => x.fio\_of\_worker[0] == 'А')

select new { id = g.Key };

foreach (var x in q4) Console.WriteLine("ID отдела: " + x.id);

//Список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А»

Console.WriteLine("\nСписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А»");

var q5 = from x in workers

group x by x.id\_of\_office into g

where g.Any(x => x.fio\_of\_worker[0] == 'А')

select new { id = g.Key };

foreach (var x in q5) Console.WriteLine("ID отдела: " + x.id);

//МНОГИЕ КО МНОГИМ

//Список всех отделов и список сотрудников в каждом отделе

Console.WriteLine("\nМНОГИЕ КО МНОГИМ");

Console.WriteLine("\nСписок всех отделов и список сотрудников в каждом отделе");

var m\_to\_m1 = from x in offices

join l in records on x.id\_of\_office equals l.id\_of\_office\_data into temp

from t1 in temp

join y in workers on t1.id\_of\_worker\_data equals y.id\_of\_worker into temp2

from t2 in temp2

group t2 by x.id\_of\_office;

foreach (var x in m\_to\_m1)

{

Console.WriteLine("ID отдела: " + x.Key);

foreach (var y in x) Console.WriteLine("ID сотрудника: {0}, ФИО сотрудника: {1}", y.id\_of\_worker, y.fio\_of\_worker);

}

//Список всех отделов и количество сотрудников в каждом отделе

Console.WriteLine("\nСписок всех отделов и количество сотрудников в каждом отделе");

var m\_to\_m2 = from x in m\_to\_m1

select new { id = x.Key, count = x.Count() };

foreach (var x in m\_to\_m2) Console.WriteLine("ID отдела: {0}, кол-во сотрудников: {1}", x.id, x.count);

Console.ReadKey();

}

}

}

Пример выполнения программы