Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»   
(МГТУ им. Н.Э. Баумана)

**Отчет**

**по дисциплине**

**“Базовые компоненты интернет технологий”**

**по лабораторной работе №7**

Выполнил: Окопный Марк Олегович

Группа: ИУ5 – 34

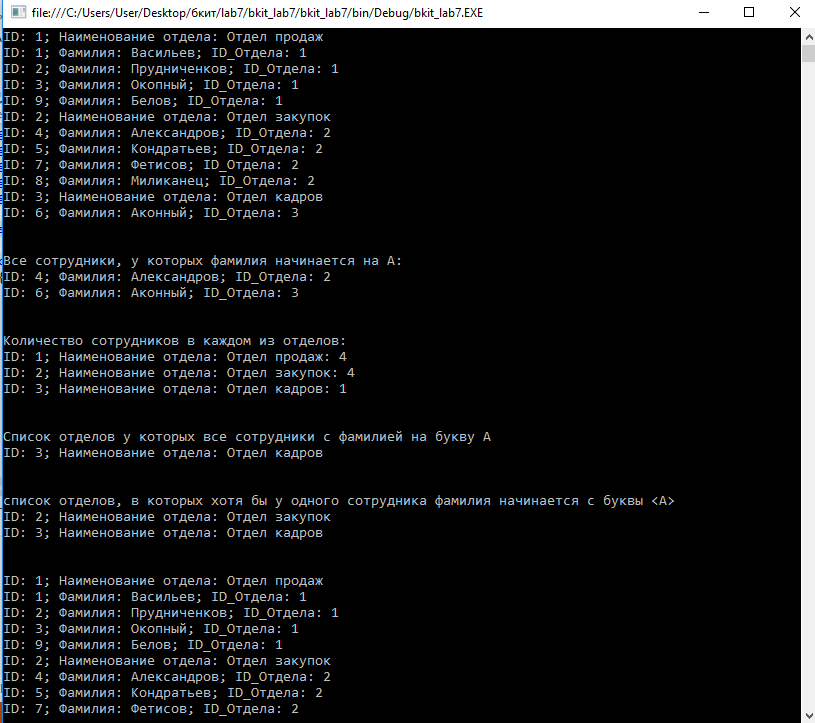
**Цель работы**

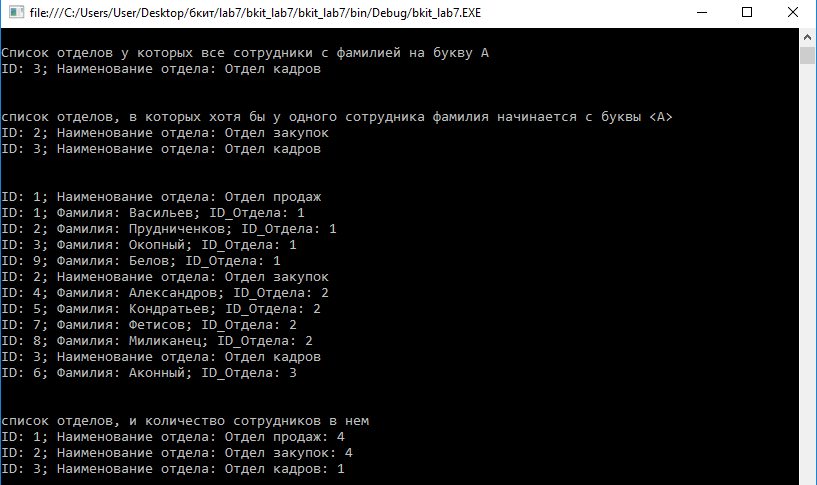
Разработать программу, реализующую работу с LINQ to Objects. В качестве примера используйте проект «SimpleLINQ» из примера «Введение в LINQ».

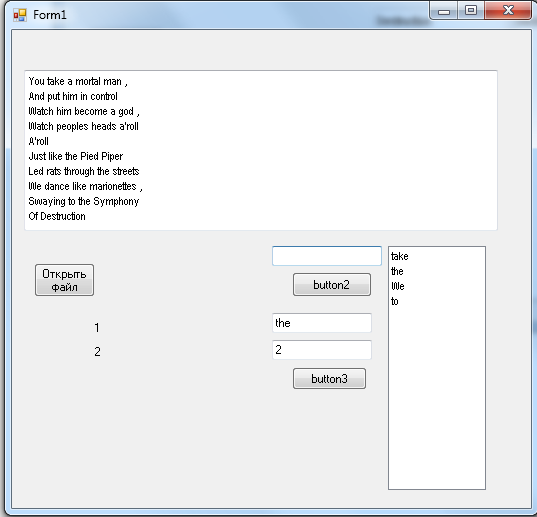
**Текст программы:**

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using System.Text; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace bkit\_lab7 |
|  | { |
|  | class Worker |
|  | { |
|  |  |
|  | public int ID; |
|  | public string Surname; |
|  | public int DepartmentID; |
|  | public Worker(int i, string sn, int d) |
|  | { |
|  | this.ID = i; |
|  | this.Surname = sn; |
|  | this.DepartmentID = d; |
|  | } |
|  | public override string ToString() |
|  | { |
|  | return "ID: " + this.ID + "; Фамилия: " + this.Surname + |
|  | "; ID\_Отдела: " + this.DepartmentID; |
|  | } |
|  | } |
|  |  |
|  | class Department |
|  | { |
|  |  |
|  | public int ID; |
|  | public string name; |
|  | public Department(int i, string n) |
|  | { |
|  | this.ID = i; |
|  | this.name = n; |
|  | } |
|  | public override string ToString() |
|  | { |
|  | return "ID: " + this.ID + "; Наименование отдела: " + this.name; |
|  | } |
|  | } |
|  |  |
|  | class DepartmentWorker |
|  | { |
|  | public int WorkerID; |
|  | public int DepartmentID; |
|  | public DepartmentWorker(int iW, int iD) |
|  | { |
|  | this.WorkerID = iW; |
|  | this.DepartmentID = iD; |
|  | } |
|  | } |
|  |  |
|  |  |
|  | class Program |
|  | { |
|  |  |
|  | static List<Worker> worker = new List<Worker>() |
|  | { |
|  | new Worker(1,"Васильев",1), |
|  | new Worker(2,"Прудниченков",1), |
|  | new Worker(3,"Окопный",1), |
|  | new Worker(4,"Александров",2), |
|  | new Worker(5,"Кондратьев",2), |
|  | new Worker(6,"Аконный",3), |
|  | new Worker(7,"Фетисов",2), |
|  | new Worker(8,"Миликанец",2), |
|  | new Worker(9,"Белов",1) |
|  | }; |
|  |  |
|  | static List<Department> departments = new List<Department>() |
|  | { |
|  | new Department(1, "Отдел продаж"), |
|  | new Department(2, "Отдел закупок"), |
|  | new Department(3, "Отдел кадров") |
|  | }; |
|  |  |
|  | static List<DepartmentWorker> departmentWorkers = new List<DepartmentWorker> |
|  | { |
|  | new DepartmentWorker(1,1), |
|  | new DepartmentWorker(2,1), |
|  | new DepartmentWorker(3,1), |
|  | new DepartmentWorker(4,2), |
|  | new DepartmentWorker(5,2), |
|  | new DepartmentWorker(6,3), |
|  | new DepartmentWorker(7,2), |
|  | new DepartmentWorker(8,2), |
|  | new DepartmentWorker(9,1), |
|  | }; |
|  |  |
|  |  |
|  | static void Main(string[] args) |
|  | { |
|  | foreach (var d in departments) |
|  | { |
|  | //Перебор (укороченный синтаксис цикла) по каждому сотруднику |
|  | var q1 = from x in worker |
|  | where (d.ID == x.DepartmentID) |
|  | select x; |
|  | //Проверка на соответствие сотрудника его отделу (вывод типа: отдел - сотрудники) |
|  | Console.WriteLine(d); |
|  | foreach (var x in q1) Console.WriteLine(x); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | //Сотрудники по первой букве фамилии: "А" |
|  | string lit = "А"; |
|  | Console.WriteLine("Все сотрудники, у которых фамилия начинается на " + lit + ":"); |
|  | var q2 = from x in worker |
|  | where (x.Surname.Substring(0, 1) == lit) |
|  | select x; |
|  | foreach (var x in q2) Console.WriteLine(x); |
|  | if (q2.Count() == 0) |
|  | { |
|  | Console.WriteLine("Ни в одном отделе нет сотрудников с фамилией начинающейся на букву " + lit); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  | //Список всех отделов с количеством сотрудников(через лямбда выражение) |
|  | Console.WriteLine("Количество сотрудников в каждом из отделов:"); |
|  | foreach (var x in departments) |
|  | { |
|  | int num = worker.Count(y => y.DepartmentID == x.ID); |
|  | Console.WriteLine(x + ": " + num); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  | //Список отделов у которых все сотрудники с фамилией на букву А |
|  | Console.WriteLine("Список отделов у которых все сотрудники с фамилией на букву А"); |
|  | bool tr; |
|  | foreach (var x in departments) |
|  | { |
|  | tr = true; |
|  | var q6 = from z in worker |
|  | where (z.DepartmentID == x.ID) |
|  | select z; |
|  | foreach (var q in q6) |
|  | { |
|  | if ((q.Surname.Substring(0, 1) != lit)) tr=false; |
|  | } |
|  | if (tr) Console.WriteLine(x); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  |  |
|  | //Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А». |
|  | Console.WriteLine("список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А»"); |
|  | foreach (var x in departments) |
|  | { |
|  | tr = false; |
|  | var q7 = from c in worker |
|  | where (c.DepartmentID == x.ID) |
|  | select c; |
|  | foreach (var x1 in q7) |
|  | { |
|  | if (x1.Surname.Substring(0, 1) == lit) tr=true; |
|  | } |
|  | if (tr) Console.WriteLine(x); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | //Перебор по отделам |
|  | foreach (var x in departments) |
|  | { |
|  | //Перебор по отделам-сотрудникам |
|  | var q5 = from y in departmentWorkers |
|  | //сравнение номера отдела сотрудника с текущим (по проходу) отделом |
|  | where (y.DepartmentID == x.ID) |
|  | //запоминание его в список q5 |
|  | select y; |
|  |  |
|  | //Перебор по списку сотрудников |
|  | var q6 = from y in worker |
|  | //Перебор по списку q5 |
|  | from z in q5 |
|  | //Сравнение номера сотрудника отдела с текущим работником |
|  | where (z.WorkerID == y.ID) |
|  | //Запоминание его в список й6 |
|  | select y; |
|  | Console.WriteLine(x); |
|  | foreach (var y in q6) Console.WriteLine(y); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  |  |
|  |  |
|  | Console.WriteLine("список отделов, и количество сотрудников в нем"); |
|  | foreach (var x in departments) |
|  | { |
|  | var q7 = from c in departmentWorkers |
|  | where (c.DepartmentID == x.ID) |
|  | select c; |
|  | int num = worker.Count(y => y.DepartmentID == x.ID); |
|  | Console.WriteLine(x + ": " + num); |
|  | } |
|  | Console.WriteLine("\n"); |
|  |  |
|  |  |
|  |  |
|  |  |
|  | Console.ReadKey(); |
|  | } |
|  | } |
|  | } |

**Результаты работы программы**





****