**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №7

«Трек курса «Функциональное программирование»»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студентка группы ИУ5-32 |  | преподаватель каф. ИУ5 |
| Федосеева Е.Ю. |  | Ю.Е. Гапанюк. |
| Подпись и дата: |  | Подпись и дата: |

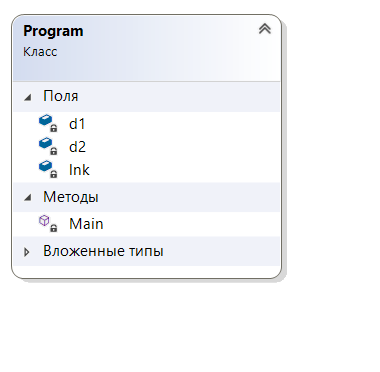
Москва, 2020 г.

**Описание задания**

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

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создайте класс «Сотрудник», содержащий поля:
   * ID записи о сотруднике;
   * Фамилия сотрудника;
   * ID записи об отделе.
3. Создайте класс «Отдел», содержащий поля:
   * ID записи об отделе;
   * Наименование отдела.
4. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением один-ко-многим разработайте следующие запросы:
   * Выведите список всех сотрудников и отделов, отсортированный по отделам.
   * Выведите список всех сотрудников, у которых фамилия начинается с буквы «А».
   * Выведите список всех отделов и количество сотрудников в каждом отделе.
   * Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы «А».
   * Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А».
5. Создайте класс «Сотрудники отдела», содержащий поля:
   * ID записи о сотруднике;
   * ID записи об отделе.
6. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением много-ко-многим с использованием класса «Сотрудники отдела» разработайте следующие запросы:
   * Выведите список всех отделов и список сотрудников в каждом отделе.
   * Выведите список всех отделов и количество сотрудников в каждом отделе.

**Диаграмма классов**

****

**Текст программы**

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Lab7.Models2;

namespace Lab7

{

class Program

{

static void Main()

{

var model = new Model1Container();

var Workers = (from Employee in model.EmployeeSet select Employee).ToList();

var Departments = (from Department in model.DepartmentSet select Department).ToList();

// Tasks

Console.WriteLine("\nВыведите список всех сотрудников и отделов, отсортированный по отделам.");

Console.WriteLine("ID) Department Name | ID) Surname");

foreach (var dep in Departments.OrderBy(e => e.Name))

{

Console.WriteLine($"Department: {dep.Name}");

foreach (var worker in dep.Employee)

{

Console.WriteLine($"{worker.Surname}");

}

Console.WriteLine();

}

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

Console.WriteLine("ID) Surname");

foreach (var worker in Workers.Where(w => w.Surname.ToUpper().StartsWith("A")))

{

Console.WriteLine($"{worker.Id}) {worker.Surname}");

}

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

Console.WriteLine();

foreach (var dep in Departments)

{

Console.WriteLine($"Department: {dep.Name} -> {dep.Employee.Count} worker.");

}

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

Console.WriteLine("ID) Department Name");

foreach (var dep in Departments

.Where(x => x.Employee.All(y => y.Surname.ToUpper().StartsWith("A"))))

{

Console.WriteLine($"{dep.Id}) {dep.Name}");

}

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

Console.WriteLine("ID) Department Name");

foreach (var dep in Departments

.Where(x => x.Employee.Any(y => y.Surname.ToUpper().StartsWith("A"))))

{

Console.WriteLine($"{dep.Id}) {dep.Name}");

}

Console.ReadLine();

}

}

}

Анализ результатов

