**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №7

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Кузнецов Григорий |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

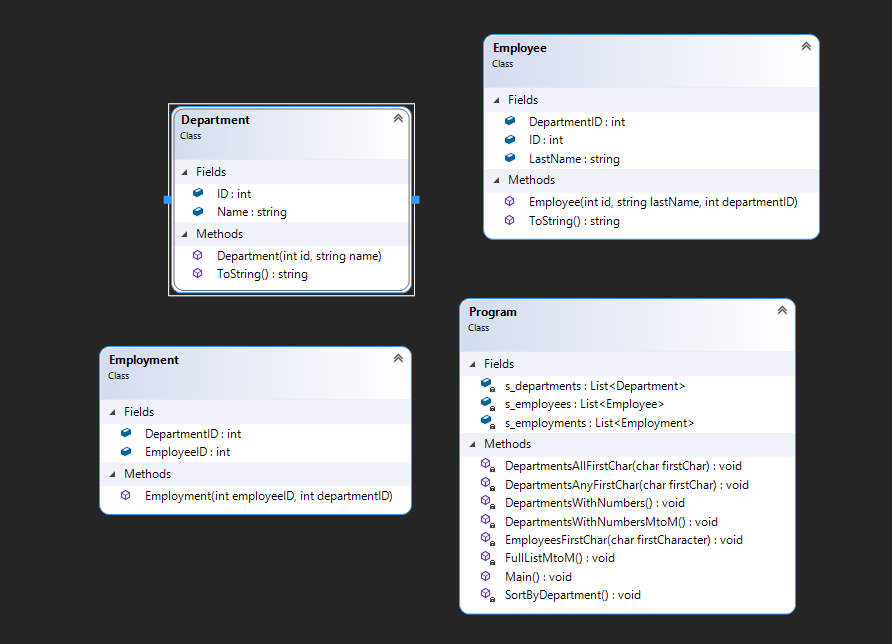
**Описание задания**

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;

using System.Linq;

using System.Collections.Generic;

namespace HMW

{

public class Employee

{

public int ID;

public string LastName;

public int DepartmentID;

public Employee(int id, string lastName, int departmentID) => (ID, LastName, DepartmentID) = (id, lastName, departmentID);

public override string ToString() => $"(ID = {ID}; LastName = {LastName}; DepartmentID = {DepartmentID})";

}

public class Department

{

public int ID;

public string Name;

public Department(int id, string name) => (ID, Name) = (id, name);

public override string ToString() => $"(ID = {ID}; Name = {Name})";

}

public class Employment

{

public int EmployeeID;

public int DepartmentID;

public Employment(int employeeID, int departmentID) => (EmployeeID, DepartmentID) = (employeeID, departmentID);

}

public class Program

{

private static readonly List<Employee> s\_employees = new List<Employee>()

{

new Employee(0, "Иванов", 0),

new Employee(1, "Петров", 0),

new Employee(2, "Сидоров", 0),

new Employee(3, "Семёнов", 1),

new Employee(4, "Сергеев", 1),

};

private static readonly List<Department> s\_departments = new List<Department>()

{

new Department(1, "Бухгалтерия"),

new Department(0, "Логистика"),

};

private static readonly List<Employment> s\_employments = new List<Employment>()

{

new Employment(0, 1),

new Employment(0, 0),

new Employment(1, 1),

new Employment(2, 0),

new Employment(2, 1),

new Employment(3, 0),

new Employment(4, 1),

};

private static void SortByDepartment()

{

Console.WriteLine("Cписок всех сотрудников и отделов, отсортированный по отделам:");

var departments = from department in s\_departments

orderby department.ID ascending

select department;

foreach (var department in departments)

{

Console.WriteLine($"\tОтдел: {department}");

var employees = from employee in s\_employees

where employee.DepartmentID == department.ID

select employee;

foreach (var employee in employees)

Console.WriteLine($"\t\t{employee}");

}

}

private static void EmployeesFirstChar(char firstCharacter)

{

Console.WriteLine($"Список всех сотрудников, чья фамилия начинается на: {firstCharacter}");

var employees = from employee in s\_employees

where employee.LastName[0] == firstCharacter

select employee;

foreach (var employee in employees)

Console.WriteLine("\t" + employee);

}

private static void DepartmentsWithNumbers()

{

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

foreach (var department in s\_departments)

{

Console.Write($"\t{department} : ");

var employees = from employee in s\_employees

where employee.DepartmentID == department.ID

select employee;

Console.WriteLine(employees.Count());

}

}

private static void DepartmentsAllFirstChar(char firstChar)

{

Console.WriteLine($"Cписок отделов, в которых у всех сотрудников фамилия начинается с буквы {firstChar}");

foreach (var department in s\_departments)

{

var employees = from employee in s\_employees

where employee.DepartmentID == department.ID

select employee;

if (employees.All(employee => employee.LastName[0] == firstChar))

Console.WriteLine($"\t{department}");

}

}

private static void DepartmentsAnyFirstChar(char firstChar)

{

Console.WriteLine($"Cписок отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы {firstChar}");

foreach (var department in s\_departments)

{

var employees = from employee in s\_employees

where employee.DepartmentID == department.ID

select employee;

if (employees.Any(employee => employee.LastName[0] == firstChar))

Console.WriteLine($"\t{department}");

}

}

private static void FullListMtoM()

{

Console.WriteLine("Список всех отделов и список сотрудников в каждом отделе (связь много-ко-многим)");

foreach (var department in s\_departments)

{

Console.WriteLine($"\t{department}:");

var employees = from employment in s\_employments

from employee in s\_employees

where employment.DepartmentID == department.ID && employment.EmployeeID == employee.ID

select employee;

foreach (var employee in employees)

Console.WriteLine($"\t\t{employee}");

}

}

private static void DepartmentsWithNumbersMtoM()

{

Console.WriteLine("Список всех отделов и количество сотрудников в каждом отделе (связь много-ко-многим)");

foreach (var department in s\_departments)

{

Console.Write($"\t{department}: ");

var employees = from employment in s\_employments

from employee in s\_employees

where employment.DepartmentID == department.ID && employment.EmployeeID == employee.ID

select employee;

Console.WriteLine(employees.Count());

}

}

public static void Main()

{

SortByDepartment();

EmployeesFirstChar('С');

DepartmentsWithNumbers();

DepartmentsAllFirstChar('С');

DepartmentsAnyFirstChar('С');

FullListMtoM();

DepartmentsWithNumbersMtoM();

}

}

}

**Пример выполнения программы**

