**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Пояснительная записка по домашнему заданию

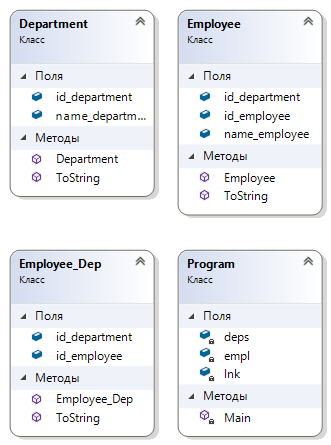
|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б |  | преподаватель каф. ИУ5 |
| Грызин Алексей |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

# 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

/// <summary>

/// Класс отдел

/// </summary>

public class Department

{

//Ключ, название отдела

public int id\_department;

public string name\_department;

public Department(int id, string name)

{

this.id\_department = id;

this.name\_department = name;

}

public override string ToString()

{

return "(ID = " + this.id\_department.ToString() + "; Наименование отдела = " + this.name\_department + ")";

/// <summary>

/// Класс сотрудник

/// </summary>

public class Employee

{

//Ключ, фамилия сотрудника, ключ отдела

public int id\_employee;

public string name\_employee;

public int id\_department;

public Employee(int id, string name, int id\_department)

{

this.id\_employee = id;

this.name\_employee = name;

this.id\_department = id\_department;

}

public override string ToString()

{

return "(ID = " + this.id\_employee.ToString() + "; Фамилия = " + this.name\_employee + "; Отдел = " + this.id\_department + ")";

}

}

/// <summary>

/// Связь между списками

/// </summary>

public class Employee\_Dep

{

public int id\_employee;

public int id\_department;

public Employee\_Dep(int i1, int i2)

{

this.id\_department = i1;

this.id\_employee = i2;

}

public override string ToString()

{

return "(ID Сотрудника = " + this.id\_employee.ToString() + "; ID Отдела = " + this.id\_department + ")";

}

}

class Program

{

//Пример данных

static List<Employee> empl = new List<Employee>()

{

new Employee(1, "Иванов", 1),

new Employee(2, "Петров", 3),

new Employee(3, "Сидоров", 4),

new Employee(4, "Демченко", 5),

new Employee(6, "Абрамов", 1),

new Employee(7, "Колков", 7),

new Employee(8, "Фернандес", 8),

new Employee(9, "Артеев", 2),

new Employee(10, "Аистов", 2),

new Employee(11, "Антонов", 2),

new Employee(12, "Васин", 7),

new Employee(13, "Уткин", 8)

};

static List<Department> deps = new List<Department>()

{

new Department(1, "МейлРу"),

new Department(2, "ОбществоЛюдейСФамилиейНаА"),

new Department(3, "Борщ"),

new Department(4, "Самокат"),

new Department(5, "БургерКинг"),

new Department(6, "КФС"),

new Department(7, "МГТУ"),

new Department(8, "МВД")

};

static List<Employee\_Dep> lnk = new List<Employee\_Dep>()

{

new Employee\_Dep(1,1),

new Employee\_Dep(1,6),

new Employee\_Dep(2,9),

new Employee\_Dep(2,10),

new Employee\_Dep(2,11),

new Employee\_Dep(3,2),

new Employee\_Dep(4,3),

new Employee\_Dep(5,4),

new Employee\_Dep(7,7),

new Employee\_Dep(7,12),

new Employee\_Dep(8,8),

new Employee\_Dep(8,13)

};

static void Main(string[] args)

{

Console.WriteLine("\nГрызин А.Н. РТ5-31");

//Задание 1

Console.WriteLine("\n-------------------------------------------------------------");

Console.WriteLine("Список сотрудников, отсортированный по отделам");

var q1 = from x in empl

orderby x.id\_department ascending

select x;

foreach (var x in q1) Console.WriteLine(x);

//Задание 2

Console.WriteLine("\n-------------------------------------------------------------");

Console.WriteLine("\nСписок сотрудников с фамилией, начинающейся на 'A'");

var q2 = from x in empl

where (x.name\_employee[0].ToString().ToUpper() == "A" || x.name\_employee[0].ToString().ToUpper() == "А")

select x;

foreach (var x in q2) Console.WriteLine(x);

//Задание 3

Console.WriteLine("\n-------------------------------------------------------------");

Console.WriteLine("\nСписок всех отделов с количеством сотрудников");

var q3 = from x in empl.Union(empl)

group x by x.id\_department into g

select new { Key = g.Key, Values = g };

foreach (var x in q3)

{

Department temp\_dep = new Department(0,"");

var f1 = (from y in deps select y).First(y => y.id\_department == x.Key);

temp\_dep = f1;

Console.WriteLine(temp\_dep.name\_department + " - " + x.Values.Count().ToString() + " сотр.");

}

//Задание 4

Console.WriteLine("\n-------------------------------------------------------------");

Console.WriteLine("\nСписок всех отделов, у всех сотрудников которых фамилия начинается на 'A'");

var q4 = from x in q2.Union(q2)

group x by x.id\_department into g

select new { Key = g.Key, Values = g};

foreach (var x in q4)

{

foreach (var y in q3)

if (x.Key == y.Key && x.Values.Count() == y.Values.Count())

{

Department temp\_dep = new Department(0, "");

var f1 = (from z in deps select z).First(z => z.id\_department == y.Key);

temp\_dep = f1;

Console.WriteLine(temp\_dep.name\_department);

}

}

//Задание 5

Console.WriteLine("\n-------------------------------------------------------------");

Console.WriteLine("\nСписок всех отделов, хотя бы у одного из сотрудников которых фамилия начинается на 'A'");

foreach (var x in q4)

{

foreach (var y in q3)

if (x.Key == y.Key)

{

Department temp\_dep = new Department(0, "");

var f1 = (from z in deps select z).First(z => z.id\_department == y.Key);

temp\_dep = f1;

Console.WriteLine(temp\_dep.name\_department);

}

}

//Задание 6

Console.WriteLine("\n-------------------------------------------------------------");

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

var lnk1 = from x in deps

join l in lnk on x.id\_department equals l.id\_department into temp

from t1 in temp

join y in empl on t1.id\_employee equals y.id\_employee into temp2

from t2 in temp2

select new { id1 = x.id\_department, id2 = t2.id\_employee };

var q5 = from x in empl.Union(empl)

group x by x.id\_department into g

select new { Key = g.Key, Values = g };

foreach (var x in q5)

{

Department temp\_dep = new Department(0, "");

var f1 = (from y in deps select y).First(y => y.id\_department == x.Key);

temp\_dep = f1;

Console.WriteLine('\n' + temp\_dep.name\_department + ':');

foreach (var i in x.Values)

{

Console.WriteLine(i.name\_employee);

}

}

//Задание 7

Console.WriteLine("\n-------------------------------------------------------------");

Console.WriteLine("\nСписок всех отделов с количеством сотрудников");

foreach (var x in q5)

{

Department temp\_dep = new Department(0, "");

var f1 = (from y in deps select y).First(y => y.id\_department == x.Key);

temp\_dep = f1;

Console.WriteLine(temp\_dep.name\_department + " - " + x.Values.Count().ToString() + " сотр.");

}

}

}

# Результат работы программы

