Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

****

Факультет ИУ

Кафедра ИУ5

**Отчет по лабораторной работе № 7**

Студент Казанцева Ксения Юрьевна

Группа ИУ5-32

Название предмета БКИТ

Руководитель

Гапанюк Ю.Е. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись

Студент

Казанцева К.Ю. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись

# Задание

**Лабораторная работа №7**

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.

# Листинг программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab\_07

{

class Program

{

public class Worker

{

public int id;

public string full\_name;

public int branch\_id;

public Worker(int id, string full\_name, int branch\_id)

{

this.id = id;

this.full\_name = full\_name;

this.branch\_id = branch\_id;

}

public override string ToString()

{

return "(id=" + this.id.ToString() + "; full\_name=" + this.full\_name + "; branch\_id=" + this.branch\_id + ")";

}

}

public class Branch

{

public int id;

public string name;

public Branch(int id, string name)

{

this.id = id;

this.name = name;

}

public override string ToString()

{

return "(id=" + this.id.ToString() + "; name=" + this.name + ")";

}

}

public class BranchWorkers

{

public int id;

public int branch\_id;

public int worker\_id;

public BranchWorkers(int id, int worker\_id, int branch\_id)

{

this.id = id;

this.worker\_id = worker\_id;

this.branch\_id = branch\_id;

}

public override string ToString()

{

return "(worker\_id=" + this.worker\_id + "; branch\_id=" + this.branch\_id + ")";

}

}

/// <summary>

/// Класс данных

/// </summary>

public class Data

{

/// <summary>

/// Ключ

/// </summary>

public int id;

/// <summary>

/// Для группировки

/// </summary>

public string grp;

/// <summary>

/// Значение

/// </summary>

public string value;

/// <summary>

/// Конструктор

/// </summary>

public Data(int i, string g, string v)

{

this.id = i;

this.grp = g;

this.value = v;

}

/// <summary>

/// Приведение к строке

/// </summary>

public override string ToString()

{

return "(id=" + this.id.ToString() + "; grp=" + this.grp + "; value=" + this.value + ")";

}

}

/// <summary>

/// Класс для сравнения данных

/// </summary>

public class DataEqualityComparer : IEqualityComparer<Data>

{

public bool Equals(Data x, Data y)

{

bool Result = false;

if (x.id == y.id && x.grp == y.grp && x.value == y.value) Result = true;

return Result;

}

public int GetHashCode(Data obj)

{

return obj.id;

}

}

public class WorkerEqualityComparer : IEqualityComparer<Worker>

{

public bool Equals(Worker x, Worker y)

{

bool Result = false;

if (x.id == y.id && x.full\_name == y.full\_name && x.branch\_id == y.branch\_id) Result = true;

return Result;

}

public int GetHashCode(Worker obj)

{

return obj.id;

}

}

public class BranchEqualityComparer : IEqualityComparer<Branch>

{

public bool Equals(Branch x, Branch y)

{

bool Result = false;

if (x.id == y.id && x.name == y.name) Result = true;

return Result;

}

public int GetHashCode(Branch obj)

{

return obj.id;

}

}

public class BranchWorkersEqualityComparer : IEqualityComparer<BranchWorkers>

{

public bool Equals(BranchWorkers x, BranchWorkers y)

{

bool Result = false;

if (x.branch\_id == y.branch\_id && x.worker\_id == y.worker\_id) Result = true;

return Result;

}

public int GetHashCode(BranchWorkers obj)

{

return obj.id;

}

}

/// <summary>

/// Связь между списками

/// </summary>

public class DataLink

{

public int d1;

public int d2;

public DataLink(int i1, int i2)

{

this.d1 = i1;

this.d2 = i2;

}

}

static List<Worker> w1 = new List<Worker>()

{

new Worker(1, "Иванов Николай Владимирович", 1),

new Worker(2, "Николаев Владимир Иванович", 2),

new Worker(3, "Савельев Дмитрий Павлович", 1),

new Worker(4, "Петров Алексей Юрьевич", 2),

new Worker(5, "Жириновский Владимир Валерьевич", 3),

new Worker(6, "Укупник Борис Аркадьевич", 3),

new Worker(7, "Алимов Николай Владимирович", 4),

new Worker(8, "Аксенов Владимир Николаевич", 4),

new Worker(9, "Ананьев Евгений Генадьевич", 1)

};

static List<Branch> b1 = new List<Branch>()

{

new Branch(1, "Отдел кадров"),

new Branch(2, "Фронт-енд"),

new Branch(3, "Бэк-енд"),

new Branch(4, "Менеджеры"),

new Branch(5, "Охрана"),

new Branch(6, "Бухгалтерия")

};

static List<BranchWorkers> bw1 = new List<BranchWorkers>()

{

new BranchWorkers(1, 1, 1),

new BranchWorkers(1, 1, 2),

new BranchWorkers(1, 2, 3),

new BranchWorkers(1, 4, 1),

new BranchWorkers(1, 5, 1),

new BranchWorkers(1, 6, 2),

new BranchWorkers(1, 4, 6)

};

//Пример данных

static List<Data> d1 = new List<Data>()

{

new Data(1, "group1", "11"),

new Data(2, "group1", "12"),

new Data(3, "group2", "13"),

new Data(5, "group2", "15")

};

static List<Data> d2 = new List<Data>()

{

new Data(1, "group2", "21"),

new Data(2, "group3", "221"),

new Data(2, "group3", "222"),

new Data(4, "group3", "24")

};

static List<Data> d1\_for\_distinct = new List<Data>()

{

new Data(1, "group1", "11"),

new Data(1, "group1", "11"),

new Data(1, "group1", "11"),

new Data(2, "group1", "12"),

new Data(2, "group1", "12")

};

static List<DataLink> lnk = new List<DataLink>()

{

new DataLink(1,1),

new DataLink(1,2),

new DataLink(1,4),

new DataLink(2,1),

new DataLink(2,2),

new DataLink(2,4),

new DataLink(5,1),

new DataLink(5,2)

};

static void Main(string[] args)

{

Console.WriteLine("Все сотрудники:");

var q1w = from x in w1 select x;

foreach (var x in q1w) Console.WriteLine(x);

Console.WriteLine("\nВсе отделы:");

var q1b = from x in b1 select x;

foreach (var x in q1b) Console.WriteLine(x);

Console.WriteLine("\nСотрудники в отделе:");

var q4bw = from x in b1

from y in w1

where x.id == y.branch\_id

select y.id + " " + y.full\_name + " " + x.name;

foreach (var x in q4bw) Console.WriteLine(x);

Console.WriteLine("\nСотрудники на букву И:");

var w\_with\_N = from x in w1

where x.full\_name[0] == 'И'

select x;

foreach (var x in w\_with\_N) Console.WriteLine(x);

Console.WriteLine("\nОтдел со всеми сотрудниками на букву А:");

var w\_all\_A = from x in b1

from y in w1

where x.id == y.branch\_id && y.full\_name[0] == 'А'

select y.id + " " + y.full\_name + " " + x.name;

foreach (var x in w\_all\_A) Console.WriteLine(x);

Console.WriteLine("\nСотрудники в отделах М-M");

var MM = from x in b1

from y in w1

from z in bw1

where x.id == z.branch\_id && z.worker\_id == y.id

select y.id + " " + y.full\_name + " " + x.name;

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

Console.ReadLine();

}

}

}

# Результаты работы программы

Все сотрудники:

(id=1; full\_name=Иванов Николай Владимирович; branch\_id=1)

(id=2; full\_name=Николаев Владимир Иванович; branch\_id=2)

(id=3; full\_name=Савельев Дмитрий Павлович; branch\_id=1)

(id=4; full\_name=Петров Алексей Юрьевич; branch\_id=2)

(id=5; full\_name=Жириновский Владимир Валерьевич; branch\_id=3)

(id=6; full\_name=Укупник Борис Аркадьевич; branch\_id=3)

(id=7; full\_name=Алимов Николай Владимирович; branch\_id=4)

(id=8; full\_name=Аксенов Владимир Николаевич; branch\_id=4)

(id=9; full\_name=Ананьев Евгений Генадьевич; branch\_id=1)

Все отделы:

(id=1; name=Отдел кадров)

(id=2; name=Фронт-енд)

(id=3; name=Бэк-енд)

(id=4; name=Менеджеры)

(id=5; name=Охрана)

(id=6; name=Бухгалтерия)

Сотрудники в отделе:

1 Иванов Николай Владимирович Отдел кадров

3 Савельев Дмитрий Павлович Отдел кадров

9 Ананьев Евгений Генадьевич Отдел кадров

2 Николаев Владимир Иванович Фронт-енд

4 Петров Алексей Юрьевич Фронт-енд

5 Жириновский Владимир Валерьевич Бэк-енд

6 Укупник Борис Аркадьевич Бэк-енд

7 Алимов Николай Владимирович Менеджеры

8 Аксенов Владимир Николаевич Менеджеры

Сотрудники на букву И:

(id=1; full\_name=Иванов Николай Владимирович; branch\_id=1)

Отдел со всеми сотрудниками на букву А:

9 Ананьев Евгений Генадьевич Отдел кадров

7 Алимов Николай Владимирович Менеджеры

8 Аксенов Владимир Николаевич Менеджеры

Сотрудники в отделах М-M

1 Иванов Николай Владимирович Отдел кадров

4 Петров Алексей Юрьевич Отдел кадров

5 Жириновский Владимир Валерьевич Отдел кадров

1 Иванов Николай Владимирович Фронт-енд

6 Укупник Борис Аркадьевич Фронт-енд

2 Николаев Владимир Иванович Бэк-енд

4 Петров Алексей Юрьевич Бухгалтерия