МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра автоматизації та інформаційних систем

Звіт з лабораторної роботи №3

З навчальної дисципліни «Організація баз даних та знань»

Виконав: студент групи КН-19-1

Данілочкін Данііл

Перевірив: старший викладач

Васильєв Денис Олегович

КРЕМЕНЧУК 2022

**Тема:** **LINQ to Object**

Завдання: Создать массив объектов, содержащий объекты близкие по контексту индивидуального задания, при этом необходимо создать класс объектов, создать массив, коллекцию и словарь объектов класса и написать запросы LINQ с использованием всех предложных стандартных операций запросов.

**Хід роботи**

Код програми:

namespace Lab3

{

class Employee

{

public int registrationNumber, salary;

public long telephoneNumber;

public string surname, name, patronymic, team, position, additionalDescription;

public DateTime dateStartWork, dateStopWork;

public static int lastRegistrationNumber = 0;

public Employee( int salary, long telephoneNumber, string surname, string name, string patronymic, string team, string position, string additionalDescription, DateTime dateStartWork, DateTime dateStopWork)

{

registrationNumber = ++lastRegistrationNumber;

this.salary = salary;

this.telephoneNumber = telephoneNumber;

this.surname = surname;

this.name = name;

this.patronymic = patronymic;

this.team = team;

this.position = position;

this.additionalDescription = additionalDescription;

this.dateStartWork = dateStartWork;

this.dateStopWork = dateStopWork;

}

public string ToShortString()

{

return surname + " " + name + " " + patronymic + ", посада: " + position + ", відділ: " + team;

}

public override string ToString()

{

return surname + " " + name + " " + patronymic + " +" + telephoneNumber + "\nПосада: " + position + ", відділ: " + team + ", зарплата: " + salary + " грн\nПрацював з " + dateStartWork.ToShortDateString() + " по " + dateStopWork.ToShortDateString() + "\nДодатковий опис: " + additionalDescription;

}

}

}

namespace Lab3

{

class Program

{

static void Main(string[] args)

{

var connection = OpenOrCreateDataBase();

var readerEmployees = new SqlCommand("select \* from Employe", connection).ExecuteReader();

List<Employee> employeesL = new List<Employee>();

while(readerEmployees.Read())

{

DateTime dateStartWork, dateStopWork;

DateTime.TryParse(readerEmployees.GetValue(8).ToString(), out dateStartWork);

DateTime.TryParse(readerEmployees.GetValue(9).ToString(), out dateStopWork);

Employee t = new Employee(Convert.ToInt32(readerEmployees.GetValue(7)),

(long)readerEmployees.GetValue(11),

(string)readerEmployees.GetValue(1),

(string)readerEmployees.GetValue(2),

(string)readerEmployees.GetValue(3),

(string)readerEmployees.GetValue(4),

(string)readerEmployees.GetValue(5),

readerEmployees.GetValue(10) is System.DBNull?"":(string)readerEmployees.GetValue(10),

dateStartWork,

dateStopWork==DateTime.MinValue?DateTime.MaxValue:dateStopWork);

employeesL.Add(t);

}

readerEmployees.Close();

CloseConnection(connection);

Console.WriteLine("\nСписок працівників:");

PrintCollection(employeesL);

Employee[] employeesA = employeesL.ToArray();

Dictionary<int, Employee> employeesD = employeesL.ToDictionary(e => e.registrationNumber);

var neRobViddil = employeesL.Where(e => e.team == "невиробничий відділ");

Console.WriteLine("Список працівників невиробничого відділу:");

PrintCollection(neRobViddil);

var salaries = employeesL.Select(e => new { e.position, e.salary });

Console.WriteLine("Список зарплат:");

foreach (var t in salaries)

{ Console.WriteLine($"{t.position} {t.salary}"); }

var brigadirs = employeesL.Where(e => e.position.Contains("бригадир"));

var directors = employeesL.Where(e => e.position.Contains("директор"));

var bd = brigadirs.Union(directors);

var lowerEmployees = employeesL.Except(bd);

var employeAndHisChief = bd.SelectMany(e => lowerEmployees.Where(le => le.team == e.team).Select(le => new { le, e }));

if (directors.Count() == 1)

{

var brigadirAndHisChief = directors.SelectMany(e => brigadirs.Select(le => new { le, e }));

employeAndHisChief = employeAndHisChief.Union(brigadirAndHisChief);

}

Console.WriteLine("\nСписок виду: працівник - безпосередній керівник");

foreach (var t in employeAndHisChief)

{

Console.WriteLine($"{t.le.surname} {t.le.name} {t.le.patronymic} ({t.le.position}) - {t.e.surname} {t.e.name} {t.e.patronymic} ({t.e.position})\n");

}

var first5Employees = employeesL.Take(5);

Console.WriteLine("Перші п`ять працівників:");

PrintCollection(first5Employees);

var allBeforFirstNoneKameniar = employeesL.TakeWhile(e => e.position.Contains("каменяр"));

Console.WriteLine("Працівники до першого, хто не каменяр:");

PrintCollection(first5Employees);

Console.ReadKey();

}

static void PrintCollection(IEnumerable<Employee> collection)

{

foreach (var t in collection)

{

Console.WriteLine(t.ToString());

Console.WriteLine();

}

}

static SqlConnection OpenOrCreateDataBase()

{

string connStr = @"Data Source=DESKTOP-M6OMTPE;

Initial Catalog=buildingCompany;

Integrated Security=True";

SqlConnection conn = new SqlConnection(connStr);

try

{

conn.Open();

Console.WriteLine("БД вiдкрита");

}

catch (SqlException se)

{

if (se.Number == 4060)

{

Console.WriteLine("Зачекайте, йде створення БД");

conn.Close();

conn = new SqlConnection(@"Data Source=DESKTOP-M6OMTPE;Integrated Security=True");

SqlCommand cmdCreateDataBase = new SqlCommand(string.Format("CREATE DATABASE [{0}]", "buildingCompany"), conn);

conn.Open();

Console.WriteLine("Посилаємо запит");

cmdCreateDataBase.ExecuteNonQuery();

conn.Close();

Thread.Sleep(5000);

conn = new SqlConnection(connStr);

conn.Open();

Console.WriteLine("БД створена");

}

}

return conn;

}

static void CloseConnection(SqlConnection connection)

{

connection.Close();

connection.Dispose();

Console.WriteLine("З’єднання успiшно завершено");

}

}

}

Зображення, що містить текст, монітор, знімок екрана, екран

Автоматично згенерований опис

Рисунок 1 – результат роботи коду

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 2 – результат роботи коду

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 3 – результат роботи коду

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 4 – результат роботи коду

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 5 – результат роботи коду

**Висновок:** в результаті виконання лабораторної роботи було вивчено і протестовано на практиці знання з використання LINQ для об`єктів, інтерфейсів IEnumerable, IQueryable.

**Контрольні питання**

1. Что представляет собой интерфейс IEnumerable? Какое его предназаначение?

IEnumerable – интерфейс который предоставляет перечислитель, который поддерживает простой перебор элементов неуниверсальной коллекции. Классы реализовавшие этот интерфейс можно рассматривать как колекции, в том числе для них доступны стандартные способы перечисления.

1. Что представляет собой интерфейс IEnumerable<T>? Какое его предназаначение?

Интерфейс IEnumerable<T> отличается от IEnumerable только тем, что колекция реализующая его может вмещать любой указанный тип.

1. Что представляет собой интерфейс IQueryable? Какое его предназаначение?

IQueryable – интерфейс который предоставляет функциональные возможности расчета запросов к конкретному источнику данных, для которого не указан тип данных.

1. Что такое отложеная операция?

Отложеная операция - это операция, которая возвращает тип IEnumerable<T>, IOrderedEnumerable<T>.

1. Какое предназначение операции Where, Select, SelectMany, Take, TakeWhile, ToArray и ToList?

Where – выбор только элементов коллекции , для которых функция возвращает истину;

Select – выбор указанных значений элементов коллекции;

SelectMany – создание выходной последовательности с проекцией "один ко многим" из входной последовательности;

Take – выбор первых n элементов коллекции;

TakeWhile – выбор всех элементов коллекции до первого, для которого заданная функция возвращает false;

ToArray – приведение коллекции к типу массив;

ToList – приведение коллекции к типу список.