

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет радіоелектроніки

Кафедра ЕОМ

ЛАБОРАТОРНА РОБОТА №4  
LINQ TO OBJECTS AND DATA SERIALIZATION

Виконав:

ст. гр. КІУКІ-20-5

Мавринський О.Д.

Прийняла:

Ляшова А.О.

Харків 2022

## 1.1 Мета роботи

Дослідження можливостей, які надаються технологією LINQ to Objects при роботі з колекціями, поєднаними до елементів інтерфейсу користувача при використанні механізму прив'язки даних.

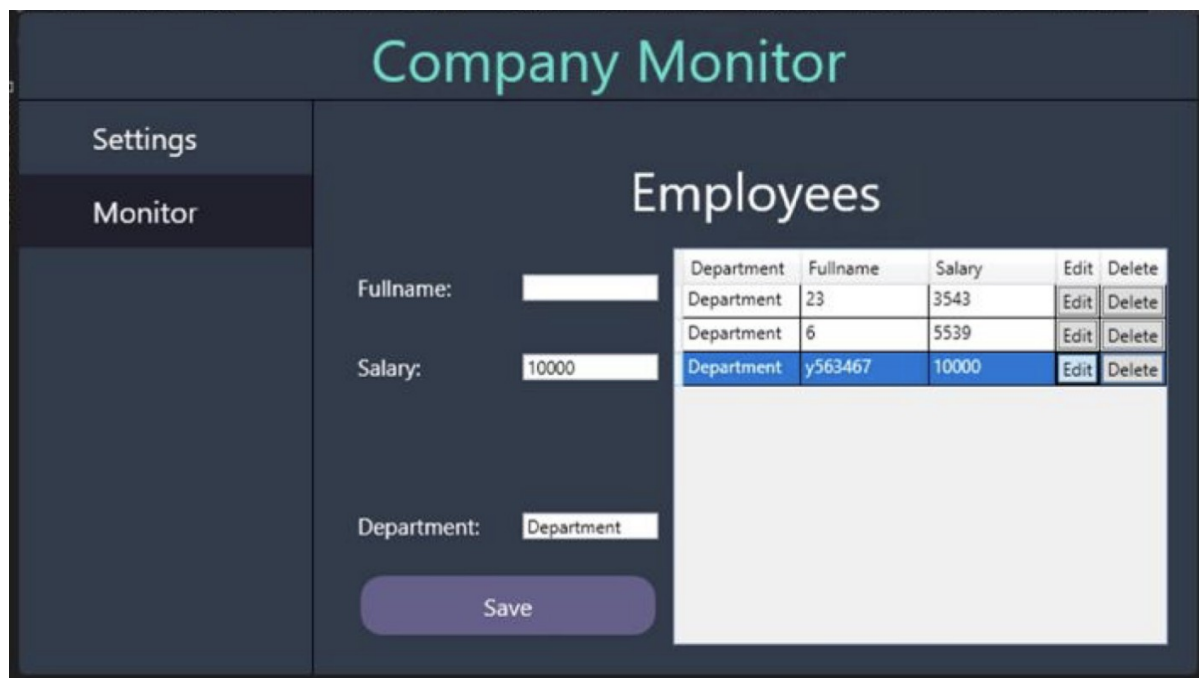
Вивчення можливостей серіалізації даних.

## 1.2 Хід роботи

### ■ Варіант 4

- Додаток необхідно реалізувати за допомогою Windows Forms або WPF. Для візуалізації даних з колекції необхідно використовувати елемент DataGridView з встановленням складної прив'язки даних. Також необхідно реалізовувати можливості зберігання та завантаження інформації за допомогою xml або json-серіалізації.
- Додаток зберігає інформацію про співробітників та відділи компанії. За допомогою діалогових вікон забезпечити можливість редагування інформації про відділи: додавання, видалення, змін назви; та співробітників: додавання видалення, зміна ПІБ, зміна заробітної плати.
- Забезпечити вивід списку співробітників всіх відділів та списку відділів компанії.
- Забезпечити можливість сортування списку співробітників за ПІБ, по відділам, по заробітній платі за збільшенням / у черзі алфавіту та проти алфавітом.
- Забезпечити вибір списку всіх співробітників, що працюють в обраному відділі компанії.

DepartmentName	View	Edit	Delete
Department	View	Edit	Delete
Department2	View	Edit	Delete
Department4	View	Edit	Delete
Department6	View	Edit	Delete




---

EmployeeModel.cs

```
using System.Runtime.Serialization;
namespace CompanyMonitor.MVVM.Model.DataProviders{
    [DataContract]
    public class EmployeeModel {
        [DataMember]
        public string Fullname { get; set; }
        [DataMember]
        public string Salary { get; set; }
        [DataMember]
        public string Department { get; set; }
    }
}
```

---

DepartmentModel.cs

```
using System;
using System.Runtime.Serialization;
namespace CompanyMonitor.MVVM.Model.DataProviders{
    [DataContract]
    public class DepartmentModel{
        [DataMember]
        public string Name { get; set; }
    }
}
```

---

DataEmployee.cs

```
using System;
using System.Collections.Generic;
```

```

using CompanyMonitor.MVVM.Model.DataProviders;
namespace CompanyMonitor.MVVM.Model.DataControllers{
    internal class DataEmployee {
        public List<EmployeeModel> CreateEmployees(int total = 2) {
            List<EmployeeModel> output = new List<EmployeeModel>();
            for (int i = 0; i < total; i++){output.Add(CreateEmployee()); }
            return output;}

        public EmployeeModel addEmployee(string sfullname, string sSalary, string
sDepartment) {
            EmployeeModel output = new EmployeeModel();
            output.Fullname = sfullname;
            output.Salary = sSalary;
            output.Department = sDepartment;
            return output;
        }
        private EmployeeModel CreateEmployee() {
            EmployeeModel output = new EmployeeModel();
            Random rnd = new Random();
            output.Fullname = rnd.Next(1, 25).ToString();
            output.Salary = rnd.Next(1, 10000).ToString();
            output.Department = "Department";
            return output;
        }
    }
}
}

```

#### DataDepartment.cs

---

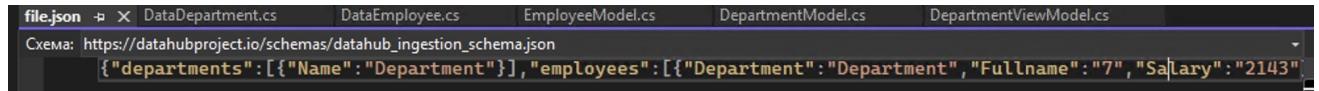
```

using System.Collections.Generic;
using CompanyMonitor.MVVM.Model.DataProviders;
namespace CompanyMonitor.MVVM.Model.DataControllers{
    internal class DataDepartment{
        readonly string depName = "Department";
        public List<DepartmentModel> CreateDepartments(){
            List<DepartmentModel> output = new List<DepartmentModel>();
            output.Add(AddDepartment(depName));
            return output;}

        public DepartmentModel AddDepartment(string dName){
            DepartmentModel output = new DepartmentModel();
            output.Name = dName;
            return output;
        }
    }
}

```

Результуючий файл:



```
file.json  DataDepartment.cs  DataEmployee.cs  EmployeeModel.cs  DepartmentModel.cs  DepartmentViewModel.cs
Схема: https://datahubproject.io/schemas/datahub_ingestion_schema.json
{"departments":[{"Name":"Department"}],"employees":[{"Department":"Department","Fullname":"7","Salary":"2143"}]}
```

GitLab: <https://gitlab.com/oleksandr.mavrynskyi/lb4-companymonitor.git>

### 1.3 Висновки

Під час виконання цієї лабораторної роботи ми навчилися працювати із даними, серіалізувати їх за допомогою вбудованих бібліотек типу JSON, XML.

Також опрацювали технологію Linq та на практиці застосували сполучені навички.