Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної

техніки

Кафедра інформатики та програмної

інженерії

Звіт

з лабораторної роботи № 1 з дисципліни
«Основи розробки програмного забезпечення на платформі Microsoft.NET»

«LINQ to Objects»

Варіант 5

Виконав студент ІП-13 Григоренко Родіон Ярославович
(шифр, прізвище, ім'я, по батькові)

Перевірила Ліщук Катерина Ігорівна
(прізвище, ім'я, по батькові)

Київ 2023

Лабораторна робота 2

LINQ to Objects

Мета – ознайомитися з обробкою даних з використанням бібліотеки LINQ to Objects.

Постановка задачі:
При виконанні комп‘ютерного практикуму необхідно виконати наступні дії:

1) Розробити структуру даних для зберігання згідно варіантів, наведених
нижче. У кожному з варіантів має бути як мінімум 4 класи. В рамках реалізації повинні бути продемонстровані зв'язки між класами: один-до-багатьох і багато-до-багатьох.

2) Розробити як мінімум 20 різних запитів, використовуючи різні дії над множинами: групування, сортування, фільтрацію, об'єднання результатів
декількох запитів в один (join, concat) та інше. Крім того, необхідно використовувати обидва можливі варіанти реалізації LINQ-запитів (класичний варіант та з використанням методів розширення), причому запити не повинні повторюватись.

Наприклад (предметне середовище Кінофільми):
a. Вивести перелік всіх кінофільмів
b. Вивести перелік акторів, котрі грають у кінофільмах, котрі починаються з літери «А»
c. Вивести перелік всіх акторів та кінофільмів, в яких вони грають
d. Вивести перелік всіх акторів, згрупувавши дані по рокам народження
e. Вивести перелік кінофільмів, в яких хоча б у одного актора прізвище починається на літеру "А"
f. Вивести всі кінофільми, відсортувавши їх по роках
g. Вивести всіх акторів, згрупувавши по амплуа та роком народженняh. З'єднати джерела даних Кінофільм» і «Актор». Вивести назву фільму, прізвище автора, прізвище актора в головній ролі.

3) Створити програмне забезпечення, котре реалізує обробку даних з

використання бібліотеки LINQ to Objects.

4) Програмне забезпечення необхідно розробити у вигляді консольного
застосування на мові C#.

5) Створити звіт. Звіт повинен містити: опис архітектури проекту, словесний опис запитів, текст програмного коду, скріншоти результатів виконання

Варіант 5 Розробити структуру даних для зберігання інформації для відстеження фінансових показників роботи пункту прокату автомобілів. В автопарк компанії входить кілька автомобілів різних марок, вартостей і типів. Кожен автомобіль має свою ціну прокату. В пункт прокату звертаються клієнти. Всі клієнти проходять обов'язкову реєстрацію - про них збирається стандартна інформація (ПІБ, адреса, телефон). Кожен клієнт може звертатися в пункт прокату кілька разів. Всі звернення клієнтів фіксуються, при цьому по кожній угоді запам'ятовуються дата видачі та очікувана дата повернення. Перед отриманням автомобіля клієнт залишає деяку заставну суму, яка йому повністю повертається після успішного повернення автомобіля. Вартість прокату автомобіля залежить не тільки від самого автомобіля, але і від терміну його прокату, а також від року випуску.

1. Архітектура Проекту.

Виконуючи першу лабораторну роботу, було створено консольне застосування на мові C#, котре реалізує котре реалізує обробку даних з використанням бібліотеки LINQ to Objects. Було створено 5 класів, а саме:

Program.cs - власне клас програми з виконуваною функцією Main

Agency.cs - клас агенції який містить списки автівок, клієнтів та угод

Auto.cs - клас, який відображає сутність автомобіль

Client.cs - клас, який відображає сутність Клієнт

Deal.cs - клас, який відображає сутність Угода

Код класів:

Agency.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab1DotNet
{
    class Agency
    {
        public List<Auto> AutoPark = new List<Auto>();
        public List<Client> Clients = new List<Client>();
        public List<Deal> Deals = new List<Deal>();
    }
}
```

Auto.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;

namespace Lab1DotNet
{
    class Auto
    {
        public string Brand;
        public string Model;
        public int Cost;
        public string Type;
        public int RentalPrice;
        public int YearOfProduction;
        public int RentalPeriod;

        public Auto(string brand, string model, int cost, string type, int
yearOfProduction, int rentalPeriod)
        {
            Brand = brand;
            Model = model;
            Cost = cost;
            Type = type;
            YearOfProduction = yearOfProduction;
            RentalPeriod = rentalPeriod;
            RentalPrice = (int)(((float)Cost / 100f) * (1f - (float)rentalPeriod / 5000f)
* (((float)YearOfProduction - 1970f)/53f));
        }
    }
}
```

Client.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab1DotNet
{
    class Client
    {
        public string Name;
        public string Address;
        public string PhoneNumber;

        public Client(string name, string address, string phoneNumber)
        {
            Name = name;
            Address = address;
            PhoneNumber = phoneNumber;
        }
    }
}
```

Deal.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab1DotNet
{
    class Deal
    {
        public Client Client;
        public Auto Auto;
        public DateTime DateOfIssue;
        public DateTime DateOfReturn;


        public Deal(Client client, Auto auto, DateTime dateOfIssue, DateTime
dateOfReturn)
        {
            Client = client;
            Auto = auto;
            DateOfIssue = dateOfIssue;
            DateOfReturn = dateOfReturn;
        }
```

```
    }
}
```

Program.cs

```
using System;
using System.Linq;

namespace Lab1DotNet
{
    class Program
    {
        static void Main(string[] args)
        {
            Agency agency = new Agency();


            Auto auto1 = new Auto("Ford", "Raptor", 150000, "off-road",2023, 0);
            Auto auto2 = new Auto("Mercedes", "G-Class", 200000, "off-road",
2023, 0);
            Auto auto3 = new Auto("Audi", "RS Q8", 210000, "off-road", 2023, 0);
            Auto auto4 = new Auto("BMW", "X6", 112000, "crossover", 2020,
500);
            Auto auto5 = new Auto("Volkswagen", "Touareg", 25000, "crossover",
2017, 1000);
            Auto auto6 = new Auto("Lada", "Priora", 10000, "sedan", 2008, 2000);
            Auto auto7 = new Auto("Nissan", "Qashqai", 14500, "crossover", 2017,
1500);
```

```
        Auto auto8 = new Auto("Toyota", "Corolla", 7000, "sedan", 2010, 3000);
        Auto auto9 = new Auto("Chrysler", "300 C", 11000, "sedan", 2008, 2300);
        Auto auto10 = new Auto("Porsche", "Cayenne", 160000, "crossover", 2023, 0);

        agency.AutoPark.Add(auto1);
        agency.AutoPark.Add(auto2);
        agency.AutoPark.Add(auto3);
        agency.AutoPark.Add(auto4);
        agency.AutoPark.Add(auto5);
        agency.AutoPark.Add(auto6);
        agency.AutoPark.Add(auto7);
        agency.AutoPark.Add(auto8);
        agency.AutoPark.Add(auto9);
        agency.AutoPark.Add(auto10);

        Client client1 = new Client("Radulf Schulte", "Mecklenburg-Vorpommern Greifswald Kurfürstendamm 23", "+49 0383 37 68 78");
        Client client2 = new Client("Dietrich Frei", "Niedersachsen Werpeloh Rudolstaedter Strasse 45", "+49 05952 53 26 46");
        Client client3 = new Client("Helmfried Schreier", "Baden-Württemberg Heilbronn Sontheim Kurfuerstendamm 85", "+49 079 99 33 63");
        Client client4 = new Client("Leon Nagel", "Nordrhein-Westfalen Oberhausen Stadtmitte Friedrichstrasse 36", "+49 0208 77 56 75");
        Client client5 = new Client("Claus Esser", "Hamburg  Hamburg Bergedorf Waßmannsdorfer Chaussee 69", "+49 040 44 81 77");
        Client client6 = new Client("Karl Stark", "Rheinland-Pfalz Meckenheim Invalidenstrasse 65", "+49 06326 49 53 30");
```

```
Client client7 = new Client("Emmerich Schumacher", "Niedersachsen Norden Kastanienallee 68", "+49 04931 60 37 51");

Client client8 = new Client("Wilfried Knopf", "Baden-Württemberg Pforzheim Buckenberg Pasewalker Straße 38", "+49 07231 59 54 52");

Client client9 = new Client("Astor Arnold", "Freistaat Thüringen Saalfeld Eschenweg 6", "+49 03671 26 79 41");

Client client10 = new Client("Mathis Groß", "Freistaat Bayern Biberbach Alt Reinickendorf 3", "+49 08271 22 11 44");


agency.Clients.Add(client1);

agency.Clients.Add(client2);

agency.Clients.Add(client3);

agency.Clients.Add(client4);

agency.Clients.Add(client5);

agency.Clients.Add(client6);

agency.Clients.Add(client7);

agency.Clients.Add(client8);

agency.Clients.Add(client9);

agency.Clients.Add(client10);


Deal deal1 = new Deal(client8,auto2, new DateTime(2021, 6, 18), new DateTime(2021,6, 20));

Deal deal2 = new Deal(client3, auto7, new DateTime(2020, 5, 18), new DateTime(2020, 6, 10));

Deal deal3 = new Deal(client5, auto4, new DateTime(2020, 9, 16), new DateTime(2020, 9, 28));

Deal deal4 = new Deal(client7, auto6, new DateTime(2020, 6, 7), new DateTime(2020, 8, 8));

Deal deal5 = new Deal(client9, auto5, new DateTime(2020, 3, 26), new DateTime(2020, 7, 25));
```

```
        Deal deal6 = new Deal(client2, auto9, new DateTime(2022, 5, 15), new DateTime(2022, 6, 20));
        Deal deal7 = new Deal(client4, auto3, new DateTime(2020, 3, 17), new DateTime(2020, 3, 29));
        Deal deal8 = new Deal(client1, auto8, new DateTime(2022, 2, 5), new DateTime(2023, 1, 1));
        Deal deal9 = new Deal(client10, auto10, new DateTime(2021, 6, 18), new DateTime(2021, 7, 5));
        Deal deal10 = new Deal(client6, auto1, new DateTime(2021, 1, 31), new DateTime(2021, 3, 10));

        agency.Deals.Add(deal1);
        agency.Deals.Add(deal2);
        agency.Deals.Add(deal3);
        agency.Deals.Add(deal4);
        agency.Deals.Add(deal5);
        agency.Deals.Add(deal6);
        agency.Deals.Add(deal7);
        agency.Deals.Add(deal8);
        agency.Deals.Add(deal9);
        agency.Deals.Add(deal10);

        //this query selects all clients' names
        Console.WriteLine("1.Clients' names:\n");
        var query1 = from c in agency.Clients select c.Name;
        foreach (var name in query1) Console.WriteLine($"\t{name}");

        //this query selects all autos' names and rental prices
        Console.WriteLine("\n2.Autos and their rental prices");
```

```
        var query2 = from a in agency.AutoPark
                     select new
                     {
                        brand = a.Brand,
                        model = a.Model,
                        rentalPrice = a.RentalPrice
                     };
        foreach (var auto in query2)
Console.WriteLine($"\t{auto.brand}\t{auto.model}\t{auto.rentalPrice}");

        //this query selects all autos whose type is crossover
        Console.WriteLine("\n3.Autos whose type is crossover:");
        var query3 = agency.AutoPark.Select(a => a).Where(a => a.Type ==
"crossover");
        foreach (var auto in query3)
Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.Type}");

        //this query calculates and selects average rental price for all autos
        Console.WriteLine("\n4.Average rantal price");
        var query4 = agency.AutoPark.Select(a => a).Average(a =>
a.RentalPrice);
        Console.WriteLine($"\t{Math.Round(query4, 2)}");

        //this query selects all autos and their costs
        Console.WriteLine("\n5.Autos and their costs:");
        var query5 = from a in agency.AutoPark
                     orderby a.Cost descending
                     select new
                     {
```

```
                    brand = a.Brand,
                    model = a.Model,
                    cost = a.Cost
                };
        foreach (var auto in query5)
Console.WriteLine($"\t{auto.brand}\t{auto.model}\t{auto.cost}");

        //this query selects all clients' names and phone numbers that begins with +49
        Console.WriteLine("\n6.Clients whose phone numbers starts with +49");
        var query6 = agency.Clients.Select(c => c).Where(n => n.PhoneNumber.StartsWith("+49"));
        foreach (var client in query6)
Console.WriteLine($"\t{client.Name}\t{client.PhoneNumber}");

        //this query group autos by types and calculates their amount in each type
        Console.WriteLine("\n7.Amount of autos by types:");
        var query7 = from a in agency.AutoPark
                group a by a.Type into type
                select new
                {
                    name = type.Key,
                    amount = type.Count()
                };
        foreach (var item in query7)
Console.WriteLine($"\t{item.name}\t{item.amount}");

        //this query selects all autos and their year of production
        Console.WriteLine("\n8.Autos and their years of production:");
```

```
        var query8 = agency.AutoPark.Select(a => a).OrderByDescending(c =>
c.Cost).OrderByDescending(y => y.YearOfProduction);
        foreach (var auto in query8)
Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.YearOfProduction}"
);


        //this query selects dates of issues and returns of all deals
        Console.WriteLine("\n9.List of deals(dates of issue and return)");
        var query9 = from d in agency.Deals
                     select new
                     {
                        dateOfIssue = d.DateOfIssue,
                        dateOfReturn = d.DateOfReturn
                     };
        foreach (var deal in query9)
Console.WriteLine($"\t{deal.dateOfIssue.ToString("d")}\t{deal.dateOfReturn.T
oString("d")}");


        //this query selects clients and autos they rent
        Console.WriteLine("\n10.Clients and autos they rent");
        var query10 = from a in agency.AutoPark from c in agency.Clients from
d in agency.Deals
                      where (d.Auto == a && d.Client == c)
                      select new
                      {
                         name = c.Name,
                         brand = a.Brand,
                         model = a.Model
                      };
```

```
        foreach (var item in query10)
Console.WriteLine($"\t{item.name}\t{item.brand}\t{item.model}");


        //this query selects autos whose cost is over 100000
        Console.WriteLine("\n11.Autos whose cost is over 100000:");
        var query11 = agency.AutoPark.Select(a => a).Where(c => c.Cost >
100000);
        foreach (var auto in query11)
Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.Cost}");


        //this query selects all clients and put them in alphabetical order by their
names
        Console.WriteLine("\n12.List of clients in alphabetical order:");
        var query12 = from c in agency.Clients
                      orderby c.Name
                      select new
                      {
                          name = c.Name
                      };
        foreach (var client in query12) Console.WriteLine($"\t{client.name}");


        //this query selects all autos and put them in descending order by their
costs
        Console.WriteLine("\n13.List of autos in their costs descending order:");
        var query13 = agency.AutoPark.Select(a => a).OrderByDescending(c
=> c.Cost);
        foreach (var auto in query13)
Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.Cost}");


        //this query selects all clients from Niedersachsen
```

```
        Console.WriteLine("\n14.Clients in Niedersachsen Land:");
        var query14 = from c in agency.Clients
                      where c.Address.Contains("Niedersachsen")
                      select new
                      {
                         name = c.Name,
                         address = c.Address
                      };
        foreach (var client in query14)
Console.WriteLine($"\t{client.name}\t{client.address}");


        //this query group all autos by their year of production and calculates
their amount in each group
        Console.WriteLine("\n15.Autos grouped by their year of production:");
        var query15 = from a in agency.AutoPark
                              group a by a.YearOfProduction into year
                              select new
                              {
                                 name = year.Key,
                                 amount = year.Count()
                              };
        foreach (var item in query15)
Console.WriteLine($"\t{item.name}\t{item.amount}");


        //this query selects all clients whose name or surname contains sch
        Console.WriteLine("\n16.Clients whose name or surname contains
sch:");
        var query16 = agency.Clients.Select(c => c).Where(n =>
n.Name.ToUpper().Contains("SCH"));
```

```
        foreach (var client in query16) Console.WriteLine($"\t{client.Name}");


        //this query selects all autos and pt them in alphabetical order by their brands' names
        Console.WriteLine("\n17.List of autos' brands names in alphabetical order:");
        var query17 = from a in agency.AutoPark
                      orderby a.Brand
                      select new
                      {
                          name = a.Brand
                      };
        foreach (var auto in query17) Console.WriteLine($"\t{auto.name}");


        //this query selects all autos produced in 2010 or earlier
        Console.WriteLine("\n18.Autos produced in 2010 or earlier:");
        var query18 = agency.AutoPark.Select(a => a).Where(y => y.YearOfProduction <= 2010);
        foreach (var auto in query18) Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.YearOfProduction}");


        //this query selects all clients whose phone numbers contain 53
        Console.WriteLine("\n19.Clients whose phone numbers contain 53:");
        var query19 = from c in agency.Clients
                      where c.PhoneNumber.Contains("53")
                      select new
                      {
                          name = c.Name,
```

```
                phoneNimber = c.PhoneNumber
            };
        foreach (var client in query19)
Console.WriteLine($"\t{client.name}\t{client.phoneNimber}");


        //this query selects full infirmation about clients that were renting premium autos and autos they've been renting
        Console.WriteLine("\n20.Full information about clients that rent premium autos and the autos:");
        var query20 = from a in agency.AutoPark
                      from c in agency.Clients
                      from d in agency.Deals
                      where (d.Auto == a && d.Client == c && a.Cost > 100000)
                      select new
                      {
                          name = c.Name,
                          address = c.Address,
                          phoneNumber = c.PhoneNumber,
                          brand = a.Brand,
                          model = a.Model,
                          cost = a.Cost,
                          type = a.Type,
                          year = a.YearOfProduction,
                          rentalPeriod = a.RentalPeriod,
                          rentalPrice = a.RentalPrice
                      };
        foreach (var item in query20)
Console.WriteLine($"\t{item.name}\t{item.address}\t{item.phoneNumber}" +
```

$"\t{item.brand}\t{item.model}\t{item.cost}\t{item.type}\t{item.year}\t{item.rentalPeriod}\t{item.rentalPrice}");

    }
  }
}

2.Запити

1. this query selects all clients' names

```
Console.WriteLine("1.Clients' names:\n");
var query1 = from c in agency.Clients select c.Name;
foreach (var name in query1) Console.WriteLine($"\t{name}");
```

```
1.Clients' names:

        Radulf Schulte
        Dietrich Frei
        Helmfried Schreier
        Leon Nagel
        Claus Esser
        Karl Stark
        Emmerich Schumacher
        Wilfried Knopf
        Astor Arnold
        Mathis Gro?
```

2. this query selects all autos' names and rental prices

```
Console.WriteLine("\n2.Autos and their rental prices");
var query2 = from a in agency.AutoPark
             select new
             {
                 brand = a.Brand,
                 model = a.Model,
                 rentalPrice = a.RentalPrice
             };
foreach (var auto in query2) Console.WriteLine($"\t{auto.brand}\t{auto.model}\t{auto.rentalPrice}");
```

```
2.Autos and their rental prices
        Ford    Raptor  1500
        Mercedes        G-Class 2000
        Audi    RS Q8   2100
        BMW     X6      950
        Volkswagen      Touareg 177
        Lada    Priora  43
        Nissan  Qashqai 90
        Toyota  Corolla 21
        Chrysler        300 C   42
        Porsche Cayenne 1600
```

3. this query selects all autos whose type is crossover

```
Console.WriteLine("\n3.Autos whose type is crossover:");
var query3 = agency.AutoPark.Select(a => a).Where(a => a.Type == "crossover");
foreach (var auto in query3) Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.Type}");
```

```
3.Autos whose type is crossover:
        BMW     X6      crossover
        Volkswagen      Touareg crossover
        Nissan  Qashqai crossover
        Porsche Cayenne crossover
```

4. this query calculates and selects average rental price for all autos

```
Console.WriteLine("\n4.Average rantal price");
var query4 = agency.AutoPark.Select(a => a).Average(a => a.RentalPrice);
Console.WriteLine($"\t{Math.Round(query4, 2)}");
```

```
4.Average rantal price
        852,3
```

5. this query selects all autos and their costs

```
Console.WriteLine("\n5.Autos and their costs:");
var query5 = from a in agency.AutoPark
             orderby a.Cost descending
             select new
             {
                 brand = a.Brand,
                 model = a.Model,
                 cost = a.Cost
             };
foreach (var auto in query5) Console.WriteLine($"\t{auto.brand}\t{auto.model}\t{auto.cost}");
```

```
5.Autos and their costs:
        Audi    RS Q8   210000
        Mercedes        G-Class 200000
        Porsche Cayenne 160000
        Ford    Raptor  150000
        BMW     X6      112000
        Volkswagen      Touareg 25000
        Nissan  Qashqai 14500
        Chrysler        300 C   11000
        Lada    Priora  10000
        Toyota  Corolla 7000
```

6. this query selects all clients' names and phone numbers that begins with +49

```
Console.WriteLine("\n6.Clients whose phone numbers starts with +49");
var query6 = agency.Clients.Select(c => c).Where(n => n.PhoneNumber.StartsWith("+49"));
foreach (var client in query6) Console.WriteLine($"\t{client.Name}\t{client.PhoneNumber}");
```

```
6.Clients whose phone numbers starts with +49
        Radulf Schulte  +49 0383 37 68 78
        Dietrich Frei   +49 05952 53 26 46
        Helmfried Schreier      +49 079 99 33 63
        Leon Nagel      +49 0208 77 56 75
        Claus Esser     +49 040 44 81 77
        Karl Stark      +49 06326 49 53 30
        Emmerich Schumacher     +49 04931 60 37 51
        Wilfried Knopf  +49 07231 59 54 52
        Astor Arnold    +49 03671 26 79 41
        Mathis Gro?     +49 08271 22 11 44
```

7. this query group autos by types and calculates their amount in each type

```
Console.WriteLine("\n7.Amount of autos by types:");
var query7 = from a in agency.AutoPark
             group a by a.Type into type
             select new
             {
                 name = type.Key,
                 amount = type.Count()
             };
foreach (var item in query7) Console.WriteLine($"\t{item.name}\t{item.amount}");
```

```
7.Amount of autos by types:
        off-road        3
        crossover       4
        sedan   3
```

8. this query selects all autos and their year of production

```
Console.WriteLine("\n8.Autos and their years of production:");
var query8 = agency.AutoPark.Select(a => a).OrderByDescending(c => c.Cost).OrderByDescending(y => y.YearOfProduction);
foreach (var auto in query8) Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.YearOfProduction}");
```

```
8.Autos and their years of production:
        Audi    RS Q8   2023
        Mercedes        G-Class 2023
        Porsche Cayenne 2023
        Ford    Raptor  2023
        BMW     X6      2020
        Volkswagen      Touareg 2017
        Nissan  Qashqai 2017
        Toyota  Corolla 2010
        Chrysler        300 C   2008
        Lada    Priora  2008
```

9. this query selects dates of issues and returns of all deals

```
Console.WriteLine("\n9.List of deals(dates of issue and return)");
var query9 = from d in agency.Deals
             select new
             {
                 dateOfIssue = d.DateOfIssue,
                 dateOfReturn = d.DateOfReturn
             };
foreach (var deal in query9) Console.WriteLine($"\t{deal.dateOfIssue.ToString("d")}\t{deal.dateOfReturn.ToString("d")}");
```

```
9.List of deals(dates of issue and return)
        18.06.2021      20.06.2021
        18.05.2020      10.06.2020
        16.09.2020      28.09.2020
        07.06.2020      08.08.2020
        26.03.2020      25.07.2020
        15.05.2022      20.06.2022
        17.03.2020      29.03.2020
        05.02.2022      01.01.2023
        18.06.2021      05.07.2021
        31.01.2021      10.03.2021
```

10. this query selects clients and autos they rent

```
Console.WriteLine("\n10.Clients and autos they rent");
var query10 = from a in agency.AutoPark from c in agency.Clients from d in agency.Deals
              where (d.Auto == a && d.Client == c)
              select new
              {
                  name = c.Name,
                  brand = a.Brand,
                  model = a.Model
              };
foreach (var item in query10) Console.WriteLine($"\t{item.name}\t{item.brand}\t{item.model}");
```

```
10.Clients and autos they rent
        Karl Stark      Ford    Raptor
        Wilfried Knopf  Mercedes        G-Class
        Leon Nagel      Audi    RS Q8
        Claus Esser     BMW     X6
        Astor Arnold    Volkswagen      Touareg
        Emmerich Schumacher     Lada    Priora
        Helmfried Schreier      Nissan  Qashqai
        Radulf Schulte  Toyota  Corolla
        Dietrich Frei   Chrysler        300 C
        Mathis Gro?     Porsche Cayenne
```

11. this query selects autos whose cost is over 100000

```
Console.WriteLine("\n11.Autos whose cost is over 100000:");
var query11 = agency.AutoPark.Select(a => a).Where(c => c.Cost > 100000);
foreach (var auto in query11) Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.Cost}");
```

```
11.Autos whose cost is over 100000:
        Ford    Raptor  150000
        Mercedes        G-Class 200000
        Audi    RS Q8   210000
        BMW     X6      112000
        Porsche Cayenne 160000
```

12. this query selects all clients and put them in alphabetical order by their names

```
Console.WriteLine("\n12.List of clients in alphabetical order:");
var query12 = from c in agency.Clients
              orderby c.Name
              select new
              {
                  name = c.Name
              };
foreach (var client in query12) Console.WriteLine($"\t{client.name}");
```

```
12.List of clients in alphabetical order:
        Astor Arnold
        Claus Esser
        Dietrich Frei
        Emmerich Schumacher
        Helmfried Schreier
        Karl Stark
        Leon Nagel
        Mathis Gro?
        Radulf Schulte
        Wilfried Knopf
```

13. this query selects all autos and put them in descending order by their costs

```
Console.WriteLine("\n13.List of autos in their costs descending order:");
var query13 = agency.AutoPark.Select(a => a).OrderByDescending(c => c.Cost);
foreach (var auto in query13) Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.Cost}");
```

```
13.List of autos in their costs descending order:
        Audi    RS Q8   210000
        Mercedes        G-Class 200000
        Porsche Cayenne 160000
        Ford    Raptor  150000
        BMW     X6      112000
        Volkswagen      Touareg 25000
        Nissan  Qashqai 14500
        Chrysler        300 C   11000
        Lada    Priora  10000
        Toyota  Corolla 7000
```

14. this query selects all clients from Niedersachsen

```
Console.WriteLine("\n14.Clients in Niedersachsen Land:");
var query14 = from c in agency.Clients
              where c.Address.Contains("Niedersachsen")
              select new
              {
                  name = c.Name,
                  address = c.Address
              };
foreach (var client in query14) Console.WriteLine($"\t{client.name}\t{client.address}");
```

```
14.Clients in Niedersachsen Land:
        Dietrich Frei   Niedersachsen Werpeloh Rudolstaedter Strasse 45
        Emmerich Schumacher     Niedersachsen Norden Kastanienallee 68
```

15. this query group all autos by their year of production and calculates their amount in each group

```
Console.WriteLine("\n15.Autos grouped by their year of production:");
var query15 = from a in agency.AutoPark
                        group a by a.YearOfProduction into year
                        select new
                        {
                            name = year.Key,
                            amount = year.Count()
                        };
foreach (var item in query15) Console.WriteLine($"\t{item.name}\t{item.amount}");
```

```
15.Autos grouped by their year of production:
        2023    4
        2020    1
        2017    2
        2008    2
        2010    1
```

16. this query selects all clients whose name or surname contains sch

```
Console.WriteLine("\n16.Clients whose name or surname contains sch:");
var query16 = agency.Clients.Select(c => c).Where(n => n.Name.ToUpper().Contains("SCH"));
foreach (var client in query16) Console.WriteLine($"\t{client.Name}");
```

```
16.Clients whose name or surname contains sch:
        Radulf Schulte
        Helmfried Schreier
        Emmerich Schumacher
```

17. this query selects all autos and pt them in alphabetical order by their brands' names

```
Console.WriteLine("\n17.List of autos' brands names in alphabetical order:");
var query17 = from a in agency.AutoPark
              orderby a.Brand
              select new
              {
                  name = a.Brand
              };
foreach (var auto in query17) Console.WriteLine($"\t{auto.name}");
```

```
17.List of autos' brands names in alphabetical order:
        Audi
        BMW
        Chrysler
        Ford
        Lada
        Mercedes
        Nissan
        Porsche
        Toyota
        Volkswagen
```

18. this query selects all autos produced in 2010 or earlier

```
Console.WriteLine("\n18.Autos produced in 2010 or earlier:");
var query18 = agency.AutoPark.Select(a => a).Where(y => y.YearOfProduction <= 2010);
foreach (var auto in query18) Console.WriteLine($"\t{auto.Brand}\t{auto.Model}\t{auto.YearOfProduction}");
```

```
18.Autos produced in 2010 or earlier:
        Lada    Priora  2008
        Toyota  Corolla 2010
        Chrysler        300 C   2008
```

19. this query selects all clients whose phone numbers contain 53

```
Console.WriteLine("\n19.Clients whose phone numbers contain 53:");
var query19 = from c in agency.Clients
              where c.PhoneNumber.Contains("53")
              select new
              {
                  name = c.Name,
                  phoneNimber = c.PhoneNumber
              };
foreach (var client in query19) Console.WriteLine($"\t{client.name}\t{client.phoneNimber}");
```

```
19.Clients whose phone numbers contain 53:
        Dietrich Frei   +49 05952 53 26 46
        Karl Stark      +49 06326 49 53 30
```

20. this query selects full information about clients that were renting premium autos and autos they've been renting

```
Console.WriteLine("\n20.Full information about clients that rent premium autos and the autos:");
var query20 = from a in agency.AutoPark
              from c in agency.Clients
              from d in agency.Deals
              where (d.Auto == a && d.Client == c && a.Cost > 100000)
              select new
              {
                  name = c.Name,
                  address = c.Address,
                  phoneNumber = c.PhoneNumber,
                  brand = a.Brand,
                  model = a.Model,
                  cost = a.Cost,
                  type = a.Type,
                  year = a.YearOfProduction,
                  rentalPeriod = a.RentalPeriod,
                  rentalPrice = a.RentalPrice
              };
foreach (var item in query20) Console.WriteLine($"\t{item.name}\t{item.address}\t{item.phoneNumber}" +
    $"\t{item.brand}\t{item.model}\t{item.cost}\t{item.type}\t{item.year}\t{item.rentalPeriod}\t{item.rentalPrice}");
```

```
20.Full information about clients that rent premium autos and the autos:
        Karl Stark      Rheinland-Pfalz Meckenheim  Invalidenstrasse 65 +49 06326 49 53 30      Ford    Raptor  150000 off-road 2023    0       1500
        Wilfried Knopf  Baden-Wurttemberg Pforzheim Buckenberg Pasewalker Stra?e 38     +49 07231 59 54 52      MercedeG-Class  200000  off-road        2023    0       2000
        Leon Nagel      Nordrhein-Westfalen Oberhausen Stadtmitte Friedrichstrasse 36   +49 0208 77 56 75       Audi    RS Q8   210000  off-road        2023    0       2100
        Claus Esser     Hamburg  Hamburg Bergedorf Wa?mannsdorfer Chaussee 69   +49 040 44 81 77        BMW     X6      112000  crossover       2020    500     950
        Mathis Gro?     Freistaat Bayern Biberbach Alt Reinickendorf 3  +49 08271 22 11 44      Porsche Cayenne 160000 crossover        2023    0       1600
```

Висновок

Виконуючи дану лабораторну роботу, я покращив свої навички створення програмного забезпечення на мові C#, створив структуру даних, навчився обробляти дані за допомогою бібліотеки LINQ to Objects, та закріпив теоретичні знання на практиці.