

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

Практикум №2

з курсу «Основи розробки програмного забезпечення на платформі
Microsoft.NET»

на тему: « LINQ to XML»

Викладач:

Крамар Ю.М.

Виконав:

студент 2 курсу
групи ПІ-15 ФІОТ
Костін В.А.

Київ-2023

Комп'ютерний практикум № 2.

Тема: LINQ to XML.

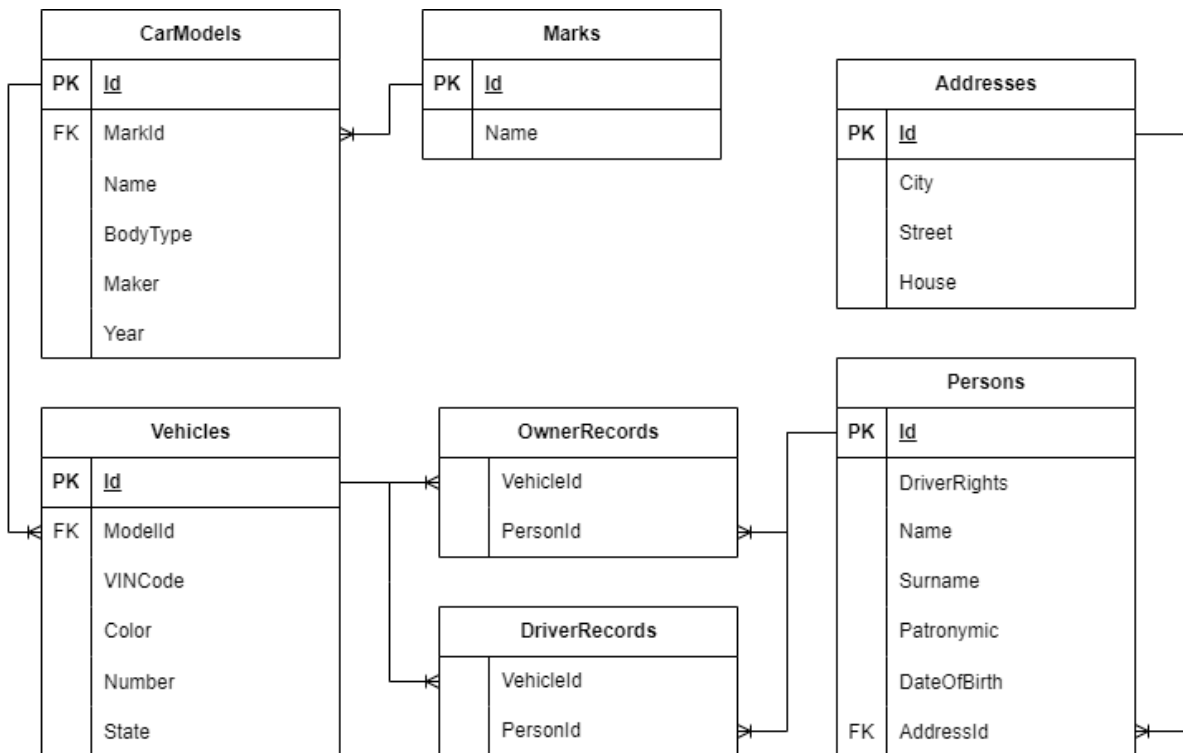
Мета: ознайомитися з обробкою XML документів з використанням технології LINQ to XML.

Постановка задачі комп'ютерного практикуму № 2:

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

- 1) Розробити структуру XML для зберігання даних згідно варіантів, наведених нижче.
 - 2) Створити XML-файл з використанням XmlWriter. Дані необхідно вводити з консолі, зберегти.
 - 3) Завантажити створені файли (або заздалегідь створені) з використанням XmlDocument та XmlSerializer.
 - 4) LINQ to XML:
 - a. Вивести зміст файлу, створеного в п.2
 - b. Для файлу, створеного в п.2 розробити як мінімум 15 різних запитів, використовуючи різні дії над отриманими даними. Запити не повинні повторюватись.
 - 5) Створити програмне забезпечення, котре реалізує обробку даних з використання бібліотеки LINQ to XML.
 - 6) Програмне забезпечення необхідно розробити у вигляді консольного застосування на мові C#.
 - 7) Звіт повинен містити: опис архітектури проекту, словесний опис запитів, текст програмного коду, скріншоти результатів виконання.
- 2) Розробити структуру даних для зберігання інформації про реєстрацію транспортних засобів. Для кожного транспортного засобу зберігається як мінімум марка авто, виробник, модель, тип кузова, рік випуску, номер шасі (VIN-код), колір, номерний знак, технічний стан, власник автомобіля, перелік водіїв, котрі мають право керувати транспортним засобом, тощо. Для власників та тих персон, котрі мають право керувати транспортним засобом, - номер прав водія, прізвище, ім'я, по батькові, дата народження, адреса реєстрації. Необхідно врахувати, що транспортний засіб може мати декілька власників (тобто бути зареєстрованим декілька разів).

Діаграма класів:



Текст запитів

Код програми можна переглянути в репозиторії:

<https://github.com/VadimkaKostin/.Net>

1) Запит для виведення усіх людей з прізвищем Ткаченко:

```
var q1 = from person in documentPerson.Root.Elements("Person")
         where person.Element("Surname").Value == "Ткаченко"
         select person;

q1.Foreach(person => Console.WriteLine(string.Join(" ",
person.Element("Name").Value, person.Element("Surname").Value)));
```

```
1 Запит
Валерій Ткаченко
Кирило Ткаченко
Вадим Ткаченко
```

Рисунок 1.2 – результат 1-го запиту.

2) Запит для виведення усіх людей з таблиці Person, які народились раніше 1990:

```
var q2 = from person in documentPerson.Root.Elements("Person")
         where
Convert.ToDateTime(person.Element("DateOfBirth").Value).Year < 1990
         select person;

q2.Foreach(person =>
```

```
Console.WriteLine(string.Join(" ", person.Element("Name").Value,
person.Element("Surname").Value,
person.Element("DateOfBirth").Value.Split(' ')[0]]));
```

```
2 Запит
Данило Марченко 07.10.1987
Борис Борисенко 09.01.1989
Володимир Романченко 21.10.1986
Петро Захарченко 22.06.1983
Данило Іващенко 16.10.1981
Сергій Савенко 07.10.1982
Ілля Яценко 14.08.1980
Гліб Пономаренко 24.03.1988
Федір Романченко 20.01.1980
Богдан Гейченко 25.08.1981
Максім Кузьменко 08.08.1989
Данило Симоненко 21.03.1988
Данило Гейченко 23.07.1987
Ілля Пономаренко 22.11.1988
Гліб Пархоменко 01.02.1988
Валерій Бондаренко 13.06.1987
Богдан Терещенко 26.09.1988
Федір Кравченко 07.05.1982
Валерій Ткаченко 01.07.1984
Богдан Гуліченко 01.01.1989
Григорій Захарченко 14.08.1985
```

Рисунок 1.3 – результат 2-го запиту.

3) Запит, що виводить ім'я людини в алфавітному порядку:

```
documentPerson.Root.Elements("Person")
.Select(person => string.Join(" ",
person.Element("Surname").Value,
person.Element("Name").Value,
person.Element("Patronymic").Value))
.OrderBy(name => name)
.Foreach(name => Console.WriteLine(name));
```

```
3 Запит
Бондаренко Борис Данилович
Бондаренко Валерій Петрович
Бондаренко Євген Вадимович
Бондаренко Сергій Олександрович
Борисенко Богдан Петрович
Борисенко Борис Вадимович
Борисенко Борис Ілліч
Борисенко Кирило Сергійович
Борисенко Максим Дмитрович
Василенко Вадим Сергійович
Гейченко Богдан Владиславович
Гейченко Володимир Вадимович
Гейченко Данило Дмитрович
Гуліченко Богдан Максимович
Гуліченко Григорій Сергійович
Гуліченко Кирило Володимирович
Єременко Валерій Олексійович
Єременко Федір Максимович
Захарченко Григорій Федорович
Захарченко Кирило Петрович
Захарченко Петро Анатолійович
Іваненко Анатолійович Федорович
Іваненко Владислав Маркович
Іваненко Володимир Вадимович
Іваненко Данило Анатолійович
Іваненко Євген Петрович
Іващенко Данило Володимирович
Іващенко Максим Володимирович
Кириленко Борис Богданович
```

Рисунок 1.4 – результат 3-го запиту.

```
documentPerson.Root.Elements("Person")
    .Select(person =>
        new
        {
            Name = string.Join(" ", person.Element("Surname").Value,
                person.Element("Name").Value),
            Age =
                GetAge(Convert.ToDateTime(person.Element("DateOfBirth").Value))
        })
    .Foreach(person => Console.WriteLine(person.Name + " " +
        person.Age));
```

4 Запит

Кравченко Валерій	26
Гуліченко Григорій	24
Романченко Петро	25
Петренко Федір	23
Коваленко Максим	30
Марченко Данило	35
Солодченко Данило	26
Скляренко Ілля	30
Тимошенко Максим	23
Симоненко Володимир	26
Борисенко Борис	34
Шевченко Федір	28
Романченко Володимир	36
Яценко Олександр	26
Савенко Владислав	25
Іваненко Владислав	29
Романченко Дмитро	24
Худенко Вадим	25
Терещенко Олексій	24
Захарченко Петро	39
Іващенко Данило	41
Кириленко Борис	28
Овчаренко Володимир	25
Ярошенко Валерій	26
Яценко Анатолійович	24
Савенко Сергій	40
Яценко Ілля	42
Пархоменко Гліб	31
Іваненко Анатолійович	33

5) Запит для підрахунку середнього віку усіх людей:

```
Console.WriteLine("\n5 Запит");
    double average = documentPerson.Root.Elements("Person")
        .Average(person =>
            GetAge(Convert.ToDateTime(person.Element("DateOfBirth").Value)));
    Console.WriteLine(average);
```

5 Запит
31,17

6) Запит для виведення людей, вік яких менший за середній вік всіх людей:

```
documentPerson.Root.Elements("Person")  
    .Where(person =>  
GetAge(Convert.ToDateTime(person.Element("DateOfBirth").Value)) < average)  
    .Select(person =>  
        new  
        {  
            Name = string.Join(" ", person.Element("Surname").Value,  
                person.Element("Name").Value),
```

```

        Age =
GetAge(Convert.ToDateTime(person.Element("DateOfBirth").Value))
    })
    .Foreach(person => Console.WriteLine(person.Name + " " +
        person.Age));

```

```

6 Запит
Кравченко Валерій 26
Гуліченко Григорій 24
Романченко Петро 25
Петренко Федір 23
Коваленко Максим 30
Солодченко Данило 26
Склярченко Ілля 30
Тимошенко Максим 23
Симоненко Володимир 26
Шевченко Федір 28
Яценко Олександр 26
Савенко Владислав 25
Іваненко Владислав 29
Романченко Дмитро 24
Худенко Вадим 25

```

Рисунок 1.7 – результат 6-го запиту.

7) Запит для виведення тих людей, які є власниками транспортних засобів:

```

documentOwnersRec.Root.Elements("OwnersRecord")
    .Select(record => int.Parse(record.Element("PersonId").Value))
    .Distinct()
    .OrderBy(personId => personId)
    .Join(documentPerson.Root.Elements("Person"),
        personId => personId,
        person => int.Parse(person.Element("Id").Value),
        (personId, person) =>
        new
        {
            Id = personId,
            Name = person.Element("Surname").Value + " " +
                person.Element("Name").Value
        })
    .Foreach(person => Console.WriteLine(person.Id + " " +
        person.Name));

```

```

7 Запит
2 Гуліченко Григорій
8 Склярченко Ілля
9 Тимошенко Максим
16 Іваненко Владислав
19 Терещенко Олексій
20 Захарченко Петро
21 Іващенко Данило
22 Кириленко Борис
25 Яценко Анатолійович
28 Пархоменко Гліб
33 Гейченко Богдан
36 Симоненко Данило
37 Яковенко Данило
39 Гейченко Данило
40 Пономаренко Ілля
44 Шевченко Богдан
46 Проценко Олексій
49 Ткаченко Валерій
50 Худенко Марк

```

Рисунок 1.8 – результат 7-го запиту.

8) Запит для виведення тих людей, які є водіями транспортних засобів:

```

documentDriversRec.Root.Elements("DriversRecord")
    .Select(record => int.Parse(record.Element("PersonId").Value))
    .Distinct()
    .OrderBy(personId => personId)
    .Join(documentPerson.Root.Elements("Person"),

```

```

personId => personId,
person => int.Parse(person.Element("Id").Value),
(personId, person) =>
new
{
    Id = personId,
    Name = person.Element("Surname").Value + " " +
person.Element("Name").Value
})
.Foreach(person => Console.WriteLine(person.Id + " " +
person.Name));

```

```

8 Запит
1 Кравченко Валерій
2 Гуліченко Григорій
3 Романченко Петро
4 Петренко Федір
5 Коваленко Максим
6 Марченко Данило
7 Солодченко Данило
8 Склярєнко Ілля
9 Тимошенко Максим
10 Симоненко Володимир

```

Рисунок 1.9 – результат 8-го запиту.

9) Запит для виведення тих людей, які є і власниками і водіями транспортних засобів:

```

documentOwnersRec.Root.Elements("OwnersRecord")
.Select(record => int.Parse(record.Element("PersonId").Value))
.Distinct()
.Intersect(
    documentDriversRec.Root.Elements("DriversRecord")
    .Select(record =>
int.Parse(record.Element("PersonId").Value))
    .Distinct()
)
.OrderBy(personId => personId)
.Join(documentPerson.Root.Elements("Person"),
personId => personId,
person => int.Parse(person.Element("Id").Value),
(personId, person) =>
new
{
    Id = personId,
    Name = person.Element("Surname").Value + " " +
person.Element("Name").Value
})
.Foreach(person => Console.WriteLine(person.Id + " " + person.Name));

```

```

9 Запит
2 Гуліченко Григорій
8 Склярєнко Ілля
9 Тимошенко Максим
16 Іваненко Владислав
19 Терещенко Олексій
20 Захарченко Петро
21 Івашченко Данило
22 Кириленко Борис
25 Яценко Анатолійович
28 Пархоменко Гліб
33 Гейченко Богдан

```

Рисунок 1.10 – результат 9-го запиту.

10) Запит для підрахунку кількості транспортних засобів у володінні власників:

```

documentOwnersRec.Root.Elements("OwnersRecord")
    .Select(record =>
        new
        {
            PersonId = int.Parse(record.Element("PersonId").Value),
            VehicleId = int.Parse(record.Element("VehicleId").Value),
        })
    .GroupBy(record => record.PersonId)
    .Join(documentPerson.Root.Elements("Person"),
        group => group.Key,
        person => int.Parse(person.Element("Id").Value),
        (group, person) =>
            new
            {
                Name = person.Element("Surname").Value + " " +
                    person.Element("Name").Value,
                VehicleAmount = group.Count()
            })
    .Foreach(person => Console.WriteLine(person.Name + " - " +
        person.VehicleAmount));

```

```

10 Запит
Борисенко Максим - 1
Худенко Марк - 1
Пархоменко Сергій - 1
Борисенко Богдан - 1
Борисенко Борис - 1
Захарченко Петро - 1
Ткаченко Кирило - 1
Скляренко Ілля - 2
Кириленко Борис - 1
Гуліченко Кирило - 1
Пархоменко Гліб - 1
Гейченко Володимир - 1
Скляренко Петро - 1
Коваленко Дмитро - 1
Ткаченко Валерій - 1
Захарченко Кирило - 1
Тимошенко Максим - 1
Бондаренко Сергій - 1
Гейченко Богдан - 1
Іващенко Данило - 1
Захарченко Григорій - 1
Шевченко Богдан - 1
Пономаренко Ілля - 2
Яценко Ілля - 2
Кулішенко Володимир - 1
Яценко Борис - 1
Бондаренко Борис - 1
Терещенко Олексій - 1
Гейченко Данило - 1

```

Рисунок 1.11 – результат 10-го запиту.

11) Запит для виведення VIN коду транспортних засобів у кожного власника:

```

documentOwnersRec.Descendants("OwnersRecord")
    .GroupBy(record => record.Element("PersonId").Value)
    .Foreach(group =>
        {
            XElement person = documentPerson.Descendants("Person")
                .First(p => p.Element("Id").Value == group.Key);

            Console.WriteLine(string.Join(" ",
                person.Element("Surname").Value,
                person.Element("Name").Value,
                person.Element("Patronymic").Value));

            documentOwnersRec.Descendants("OwnersRecord")
                .Where(record => record.Element("PersonId").Value ==
                    group.Key)

```



```

.Foreach(record =>
{
    XElement vehicleId = record.Element("VehicleId");

    XElement vehicle = documentVehicle.Descendants("Vehicle")
.First(v => v.Element("Id").Value == vehicleId.Value);

    Console.WriteLine("\t" + string.Join(" ",
vehicle.Element("VINCode").Value));
});

});

```

```

11 Запит
Борисенко Максим Дмитрович
c42d707b-4a83-484b-a3aa-9b5d0a31d001
Худенко Марк Борисович
c42d707b-4a83-484b-a3aa-9b5d0a31d001
Пархоменко Сергій Олексійович
33275987-4ed3-4bb0-aa29-0f114ac355ad
Борисенко Богдан Петрович
fcb90294-7410-4799-8215-0cc0f07221cc
Борисенко Борис Ілліч
fcb90294-7410-4799-8215-0cc0f07221cc
Захарченко Петро Анатолійович
c9d88f42-756e-421c-9455-af56c1b86438
Ткаченко Кирило Григорійович
c9d88f42-756e-421c-9455-af56c1b86438
Скляренко Ілля Владиславович
1083cd07-713c-4e71-9964-0dcdaea07e38
cb0c440e-0ea9-4f0e-9a93-fddfae829318
Кириленко Борис Богданович
1083cd07-713c-4e71-9964-0dcdaea07e38
Гуліченко Кирило Володимирович
be3cc27e-f1bc-489a-9bd8-3d1ac4c7c89f
Пархоменко Гліб Данилович
be3cc27e-f1bc-489a-9bd8-3d1ac4c7c89f
Гейченко Володимир Вадимович
d45ab23c-7e5d-4f91-99d5-1327b8c81d86
Скляренко Петро Богданович
d45ab23c-7e5d-4f91-99d5-1327b8c81d86

```

Рисунок 1.12 – результат 11-го запиту.

12) Запит для виведення всіх людей, що живуть на бульварі Лесі Українки:

```

documentPerson.Descendants("Person")
.Join(documentAddress.Descendants("Address"),
person => int.Parse(person.Element("AddressId").Value),
address => int.Parse(address.Element("Id").Value),
(person, address) => new
{
    Name = string.Join(" ", person.Element("Surname").Value,
person.Element("Name").Value),
    Address = address.Element("Street").Value + ", " +
address.Element("House").Value
})
.Where(person => person.Address.Contains("бул.Лесі Українки"))

.Foreach(person => Console.WriteLine(person.Name + " - " +
person.Address));

```

```

12 Запит
Пархоменко Гліб - бул.Лесі Українки, 17
Павленко Григорій - бул.Лесі Українки, 8
Гейченко Володимир - бул.Лесі Українки, 12
Коваленко Богдан - бул.Лесі Українки, 10

```

Рисунок 1.13 – результат 12-го запиту.

13) Запит для виведення усіх транспортних засобів з інформацією про марку, модель та VINCode:

```
var vehicles = documentVehicle.Descendants("Vehicle")
    .Select(vehicle =>
        new
        {
            Id = int.Parse(vehicle.Element("Id").Value),
            ModelId = int.Parse(vehicle.Element("ModelId").Value),
            VINCode = vehicle.Element("VINCode").Value,
        });

var models = documentModel.Descendants("CarModel")
    .Select(model =>
        new
        {
            Id = int.Parse(model.Element("Id").Value),
            Name = model.Element("Name").Value,
            MarkId = int.Parse(model.Element("MarkId").Value)
        });

var marks = documentMark.Descendants("Mark")
    .Select(mark =>
        new
        {
            Id = int.Parse(mark.Element("Id").Value),
            Name = mark.Element("Name").Value,
        });

vehicles.Join(models,
    vehicle => vehicle.ModelId,
    model => model.Id,
    (vehicle, model) =>
        new
        {
            Id = vehicle.Id,
            MarkId = model.MarkId,
            Model = model.Name,
            VINCode = vehicle.VINCode
        })
    .Join(marks,
    vehicle => vehicle.MarkId,
    mark => mark.Id,
    (vehicle, mark) =>
        new
        {
            Id = vehicle.Id,
            Mark = mark.Name,
            Model = vehicle.Model,
            VINCode = vehicle.VINCode
        })
    .Foreach(vehicle =>
        Console.WriteLine(string.Join(" ", vehicle.Id, vehicle.Mark,
            vehicle.Model, vehicle.VINCode)));
```

```

13 Запит
1 Toyota Camry c42d707b-4a83-484b-a3aa-9b5d0a31d001
2 Toyota Camry 33275987-4ed3-4bb0-aa29-0f114ac355ad
3 Toyota Camry fcb90294-7410-4799-8215-0cc0f07221cc
4 Toyota Rav4 c9d88f42-756e-421c-9455-af56c1b86438
5 Toyota Rav4 1083cd07-713c-4e71-9964-0dcdaea07e38
6 Toyota Rav4 be3cc27e-f1bc-489a-9bd8-3d1ac4c7c89f
7 Toyota Rav3 d45ab23c-7e5d-4f91-99d5-1327b8c81d86
8 Toyota Rav3 5cd72d4f-829d-4dc4-a842-df9f74927829
9 Toyota Rav3 f22f3287-87f3-4634-b8c3-c8bb4df48d0e
10 BMW GLA SUV 8484575e-962d-4b5a-b25e-1866f4c30e96
11 BMW GLA SUV ff60f25a-8657-4663-ad0b-5bcbacfc4348
12 BMW GLA SUV c0959f3f-b487-43a9-a926-6f1cae023a2d
13 BMW GLB SUV cb0c440e-0ea9-4f0e-9a93-fddfae829318
14 BMW GLB SUV f878ca74-19e5-4aa5-9484-41b0531c27f5
15 BMW GLB SUV 680807d2-c758-45b4-880a-90347a93013a
16 BMW GLC SUV 404ccea-3ca1-4bfb-a915-e60fd811b2b1
17 BMW GLC SUV 66c57e6c-0138-4fb0-878d-5d248f8a2293
18 BMW GLC SUV e54705bf-bdc1-478c-b5b2-3b1bef054e30
19 Mercedes i4 e447107b-bd75-48b6-8de2-3e90be2a9d64
20 Mercedes i4 15dea608-97ce-48aa-86ec-70de26178ca9
21 Mercedes i4 06b3f2bd-f22e-48e3-b168-ab1c3b949a97
22 Mercedes i4 M50 381ec6d3-249e-4126-aa99-f392376d1ed4
23 Mercedes i4 M50 326bac95-ccb9-4fc0-9602-e21f91969576
24 Mercedes i4 M50 ae545ab5-03f0-4d54-87bb-a150729b35f4
25 Mercedes iX 45985d45-a27b-479d-8a19-326517827b60
26 Mercedes iX ef733c55-f7c3-4b2c-ab3c-d20d68ee83e6
27 Mercedes iX 16b065c7-a1a8-427b-abea-04996a7b0bbb

```

Рисунок 1.14 – результат 13-го запиту.

14) Запит для виведення ієрархії XML документу від кореню до елемента ім'я людини:

```
int count = 0;
```

```
documentPerson.Descendants("Name").First().AncestorsAndSelf().Reverse()
    .Foreach(element => Console.WriteLine(Tabs(count++) + $"-
{element.Name}"));
```

```

14 Запит
-Persons
- Person
- Name

```

Рисунок 1.15 – результат 14-го запиту.

15) Запит для виведення повної ієрархії XML документу:

```
documentPerson.Root.DescendantsAndSelf()
    .Select(element => element.Name)
    .Distinct()
    .Foreach(element =>
    {
        int tabs =
documentPerson.Descendants(element).First().Ancestors().Count();
        Console.WriteLine(Tabs(tabs) + "-" + element);
    });
```

```

15 Запит
-Persons
- Person
- Id
- DriverRights
- Name
- Surname
- Patronymic
- DateOfBirth
- AddressId

```

Рисунок 1.16 – результат 15-го запиту.

Висновок

Протягом другого комп'ютерного практикуму ми ознайомитися з роботою з XML: зберіганням даних, записом, читанням, серіалізацією, LINQ to XML. Для створення запитів використовувались класи XDocument та XElement, за допомогою яких можна отримати доступ до вмісту файлу XML, діставати данні з файлу та заносити нові елементи.