

## Аналіз методів LINQ для формування навчального розкладу

Вінницький національний технічний університет

### Анотація

Проведено аналіз розширення LINQ для формування навчального розкладу, а також основних методів LINQ.

**Ключові слова:** навчальний розклад, LINQ, LINQ to objects.

### Abstract

The analysis of LINQ extension for formation of the educational schedule, and also the basic LINQ methods is carried out.

**Keywords:** educational schedule, LINQ, LINQ to objects.

### Вступ

Розробка програмного додатку для формування навчального розкладу доволі важка задача [1]. Програмний додаток зберігає та зчитує велику кількість даних про викладачів, їх заняття, навчальні групи та аудиторії. Також при збереженні даних, потрібно провести перевірки, чи не має даний викладач занять в даний час для відсутності колізії. Для прискорення розробки використовується розширення мови C# – LINQ [2].

### Результати дослідження

LINQ – компонент Microsoft .NET Framework, який додає нативні можливості виконання запитів даних до мов, що входять у .NET. LINQ розширює можливості мови, додаючи до неї вирази запитів, що є схожими на твердження SQL та можуть бути використані для зручного отримання та обробки даних масивів, XML документів, реляційних баз даних та сторонніх джерел. LINQ також визначає набір імен методів (що називаються стандартними операторами запитів, або стандартними операторами послідовностей), а також правила перекладу, що має використовувати компілятор для перекладу текучих виразів у звичайні, використовуючи їх назву, лямбда-вирази та анонімні типи [3].

При побудові розкладу в якості вхідних даних виступають навчальне навантаження, навчальні групи, навчальні аудиторії, викладачі та список дисциплін. В програмному додатку усі ці дані є об'єктами і для маніпуляцій із ними використовується LINQ to Objects. Провайдер LINQ to Objects використовується для колекцій, що знаходяться в пам'яті та застосовує для їх обробки локальний функціонал LINQ. Код, згенерований даним провайдером, реалізує архітектурний шаблон Sequence і дозволяє локально виконувати операції над колекціями IEnumerable<T>. Поточна реалізація LINQ to Objects перевіряє, які інтерфейси реалізовані у типі, що входить до колекції, щоб використати їх для швидкого виконання тих, чи інших запитів, якщо вони підтримуються типом під час виконання.

Основний оператор, який використовується для перевірок об'єктів для відсутності колізій, це оператор Any. Any перевіряє чи хоча б один елемент у колекції відповідає предикату та повертає відповідне булеве значення. Виклик Any без предикату поверне істину, якщо в колекції є хоча б один елемент [2]. Приклад використання в коді програмного додатку:

```
cell.Teacher.ScheduleCells.Any(s => s.DayOfWeek == cell.DayOfWeek && s.LessonNumber == cell.LessonNumber
```

В даному прикладі оператор Any застосовується до об'єкта cell типу ScheduleCell, який містить змінну Teacher, тобто вчитель який проводить заняття. За допомогою LINQ to Objects виконується прив'язка до колекції ScheduleCells, тобто до всіх занять, які вже має даний викладач, і виконуємо перевірку, використовуючи анонімні типи, чи проводиться це заняття в той же час, в який хочемо його додати. Якщо у викладача уже є заняття в цей час – оператор поверне значення true, інакше false [3].

Аналогічна перевірка проводиться для аудиторій, чи не проводяться заняття у цей же час, в який хочемо додати заняття:

```
cell.Classroom.ScheduleCells.Any(s => s.DayOfWeek == cell.DayOfWeek && s.LessonNumber == cell.LessonNumber)
```

І також перевірка проводиться для кожної навчальної групи, чи не проводяться заняття у цей же час, в який хочемо додати заняття:

```
foreach (var item in cell.StudentGroups)
    item.ScheduleCells.Any(s => s.DayOfWeek == cell.DayOfWeek && s.LessonNumber == cell.LessonNumber)
```

В результаті виконання усіх перевірок, програмний додаток в залежності від їх результату, або додасть заняття у навчальний розклад, або виведе повідомлення про помилку, і із яким об'єктом виникли колізії.

### Висновки

Отже, використання частини мови C# – LINQ, а саме LINQ to Objects значно полегшує процес розробки, за рахунок використання спрощених синтаксичних конструкцій, лямбда виразів та анонімних типів.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шелепало, М., & Романюк, О. 2021 Mar 3. Аналіз методів автоматизованого формування навчального розкладу. НТКП ВНТУ. Факультет інформаційних технологій та комп'ютерної інженерії.
2. LINQ [Електронний ресурс]. Режим доступу: URL: <https://uk.wikipedia.org/wiki/LINQ>
3. Синтаксис LINQ [Електронний ресурс]. Режим доступу: URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/linq/>
4. Булева функція [Електронний ресурс]. Режим доступу: URL: [https://uk.wikipedia.org/wiki/%D0%91%D1%83%D0%BB%D0%B5%D0%B2%D0%B0\\_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F](https://uk.wikipedia.org/wiki/%D0%91%D1%83%D0%BB%D0%B5%D0%B2%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F)

**Шелепало Максим Олегович** — студент групи ЗПП-176, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, email: [maksym.shelepalo@gmail.com](mailto:maksym.shelepalo@gmail.com).

**Романюк Оксана Володимирівна** – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця email: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)

**Shelepalo Maksym O.** — Department Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: [maksym.shelepalo@gmail.com](mailto:maksym.shelepalo@gmail.com).

**Romaniuk Okasana V.** – Ph.D., Associate Professor of Software, Vinnytsia National Technical University, Vinnytsia, email: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)