|  |  |
| --- | --- |
| Міністерство освіти і науки України  НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  “ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ” | |
| Кафедра управління проєктами в інформаційних технологіях | |
| Звіт з лабораторної № 2  «Колекції C#, робота з даними з використанням технології LINQ to Objects використання методів розширень та узагальнень у C#.»  з дисципліни «Стек технологій»  Варіант | |
|  | *Виконали*:  студенти  групи КН-322а  Данилова Ангеліна Вадимівна  Непоменко Дмитро Григорович  Лобач Даніїл Артемович |
| Харків, НТУ “ХПІ”  2024 | |

**Мета:**

Навчитися використовувати методи розширення та узагальнення у мові програмування C#, вивчити можливості колекцій, отримати практичні навички роботи з колекціями .NET, освоїти методи обробки даних за допомогою технології LINQ to Objects. Ознайомитись підходом до розробки Test Driven Development та розробкою unit-тестів.

**Теоретичні відомості**

**Завдання**

1. Опрацювати теорію (див. додаток А)

2. Реалізувати методи розширення:

для класу String:

* інвертування рядка;
* підрахунок кількості входжень заданого у параметрі символа у рядок.

для одновимірних масивів:

* метод, що визначає скільки разів зустрічається задане значення у масиві (метод має працювати для одновимірних масивів усіх типів, для реалізації даного методу розширення використайте узагальнення та їх обмеження за допомогою “where”);
* метод, що повертає новий масив такого ж типу і формує його з унікальних елементів (видаляє повтори);

Реалізувати *unit-тести* для демонстрації роботи реалізованих методів розширення.

3. Реалізувати узагальнений клас для зберігання «розширеного словника» де для ключа передбачається зв'язок з двома значеннями:

ExtendedDictionary<T, U, V>

де T - тип даних ключа,

U - тип даних першого значення,

V - тип даних другого значення.

Подання елемента словника реалізувати у вигляді окремого класу:

ExtendedDictionaryElement<T, U, V>

передбачивши властивості для доступу до ключа, першого та другого значення.

Словник повинен мати можливість використання у циклах foreach:

foreach (var elem in array) { … }

Необхідно передбачити наступні операції:

* додавання елемента у словник;
* видалення елемента зі словника за заданим ключем;
* перевірка наявності елемента із заданим ключем;
* перевірка наявності елемента із заданим значенням (значення1 та значення2);
* повернення елемента за заданим ключем (реалізувати операцію індексування);
* властивість, що повертає кількість елементів;

Реалізувати *unit-тести* для демонстрації роботи з реалізованим словником.

4. За допомогою синтаксису запитів [LINQ to Objects](http://habrahabr.ru/post/81592/) реалізувати обробку послідовностей згідно одного з варіантів, наведених нижче. **Номер варіанта необхідно обирати за останньою цифрою залікової книжки.**

Дани масиви наступного змісту:

|  |
| --- |
| String[] A = {  "Falluot 3", "Daxter 2", "System Shok 2", "Morrowind", "Pes 2013"  };  int[] B = { 2, -7, -10, 6, 7, 9, 3 };  String[] C = {  "Light Green", "Red", "Green", "Yellow", "Purple", "Dark Green",  "Light Red", "Dark Red", "Dark Yellow", "Light Yellow"  }; |

Класи:

|  |
| --- |
| Car, содержащий максимальную скорость, цвет и производителя, число пассажиров.  Product, содержащий название продукта, количество упаковок на складе, описание. |

Списки (контейнери):

|  |
| --- |
| List<string> myCars = new List<String> { "Yugo", "Aztec", "BMW" };  List<string> yourCars = new List<String> { "BMW", "Saab", "Aztec" }; |

**Варіант 4**

1. Написати запит для знаходження елементів масиву А, що включають пробіл.
2. Написати запит для негайного знаходження негативних елементів масиву В як List[].
3. Написати запит для знаходження всіх відтінків зеленого кольору в масиві С.
4. Створити перелік автомобілів класу Car. Вибрати зі списку автомобілі Audi та вивести всю інформацію про них.
5. Створити перелік продуктів класу Product. Вибрати продукти, які закінчилися і вивести всю інформацію про них.
6. Вивести результуючий набір списків myCars та yourCars, що містить спільні елементи.

**Хід роботи**

**Висновки**