Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

**Звіт про виконану роботу до лабораторного заняття № 4**

З курсу **Архітектура програмної платформи .NET**

***Виконала:*** студентка групи41ІПЗ

спеціальності «Інженерія програмного

забезпечення»

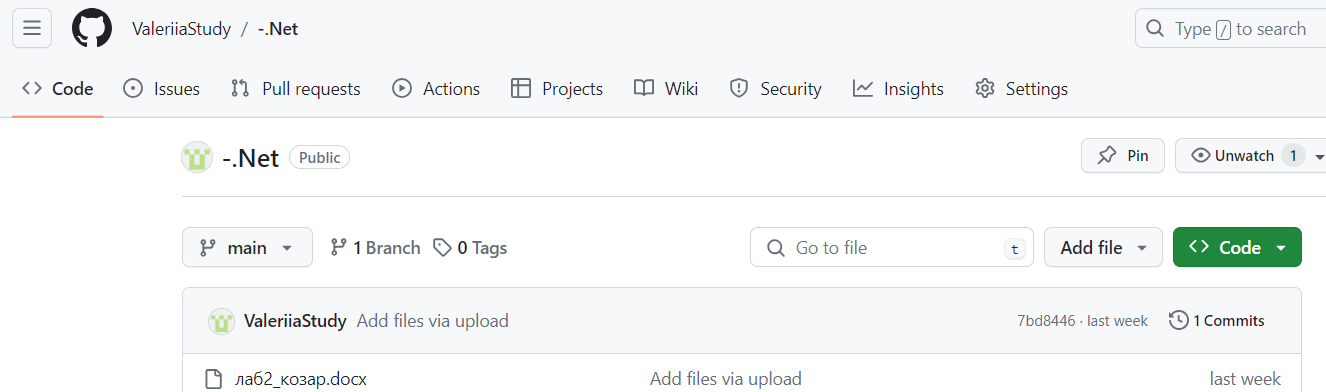
***Козар В.М.***

***Викладач:  [Січкаренко](https://moodle.fmif.udu.edu.ua/user/view.php?id=21135&course=1)***

***[Володимир Олександрович](https://moodle.fmif.udu.edu.ua/user/view.php?id=21135&course=1)***

1. Ознайомилась з теоретичною частиною.

2. Обрала тему - Текстовий редактор

3. Перейти в локальний репозиторій з лабораторними роботами;

4. Обрати тему з переліку запропонованих або визначити власну;

5. Створити новий консольний проект на мові програмування C# в локальному репозиторії;

6. Визначити необхідні колекції, які будуть використовуватись в обраній темі;

7. Під час реалізації програми використовувати роботу з елементами (додавання, видалення,

оновлення);

8. Продемонструвати роботу програми;

9. Результат роботи програми зберегти у вигляді скріншоту (png або jpeg) в репозиторії з

проектом;

|  |
| --- |
| using System;  using System.Collections.Generic;  class TextEditor  {  static List<string> textLines = new List<string>();  static void Main()  {  Console.WriteLine("Простий текстовий редактор. Введіть команду або 'exit' для виходу.");  while (true)  {  Console.Write("Команда: ");  string command = Console.ReadLine().ToLower();  switch (command)  {  case "add":  AddLine();  break;  case "delete":  DeleteLine();  break;  case "update":  UpdateText();  break;  case "show":  ShowText();  break;  case "exit":  Environment.Exit(0);  break;  default:  Console.WriteLine("Невідома команда. Спробуйте ще раз.");  break;  }  }  }  static void AddLine()  {  Console.Write("Введіть рядок для додавання: ");  string newLine = Console.ReadLine();  textLines.Add(newLine);  Console.WriteLine("Рядок успішно додано.");  }  static void DeleteLine()  {  Console.Write("Введіть номер рядка для видалення: ");  if (int.TryParse(Console.ReadLine(), out int lineNumber) && lineNumber > 0 && lineNumber <= textLines.Count)  {  textLines.RemoveAt(lineNumber - 1);  Console.WriteLine("Рядок успішно видалено.");  }  else  {  Console.WriteLine("Невірний номер рядка. Спробуйте ще раз.");  }  }  static void UpdateText()  {  Console.Write("Введіть номер рядка для оновлення: ");  if (int.TryParse(Console.ReadLine(), out int lineNumber) && lineNumber > 0 && lineNumber <= textLines.Count)  {  Console.Write("Введіть новий текст для рядка: ");  textLines[lineNumber - 1] = Console.ReadLine();  Console.WriteLine("Рядок успішно оновлено.");  }  else  {  Console.WriteLine("Невірний номер рядка. Спробуйте ще раз.");  }  }  static void ShowText()  {  Console.WriteLine("Текстовий редактор:");  for (int i = 0; i < textLines.Count; i++)  {  Console.WriteLine($"{i + 1}. {textLines[i]}");  }  }  } |
| 3 |

1. Зафіксувати зміни (git commit);

4

11. Надіслати зміни у віддалений репозиторій (git push);

5

**Відповіді на запитання**

1 Масив даних:

Масив даних - це структура даних, яка зберігає фіксовану кількість елементів одного типу в послідовності. Кожен елемент масиву може бути доступний за допомогою індексу.

2 Індекс та індексатор:

Індекс: Це ціле число, яке вказує на позицію елемента в колекції чи масиві.

Індексатор: Це механізм, який дозволяє використовувати екземпляр класу, як масив. Визначається в класі за допомогою ключового слова this та індексу.

3 Ініціалізація колекції:

Це процес надання початкових значень елементам колекції. В C# існують різні способи ініціалізації, такі як визначення значень при оголошенні або використання конструкторів.

4 Створення масиву стандартним способом:

csharpCopy code

int[] myArray = new int[5];

5 Створення масиву з використанням класу System.Array:

csharpCopy code

Array myArray = Array.CreateInstance(typeof(int), 5);

6 Інтерфейси, реалізовані класом System.Array:

IEnumerable, ICloneable, IList, ICollection, IComparer, IStructuralComparable, IStructuralEquatable.

7 Ітерація по елементах масиву:

Зазвичай використовують цикли, такі як for чи foreach. Можна також використовувати LINQ-вирази для обробки елементів колекції.

8 Простір імен System.Collections:

Призначений для забезпечення основних інтерфейсів та класів колекцій в .NET.

Включає такі простори імен як System.Collections, System.Collections.Generic, System.Collections.Specialized, System.Collections.ObjectModel та інші.

9 IComparer, Comparer, IComparable:

IComparer: Інтерфейс, який дозволяє визначити метод порівняння для двох об'єктів.

Comparer: Клас, який надає стандартні реалізації для інтерфейсу IComparer.

IComparable: Інтерфейс, який визначає метод порівняння для одного об'єкта.