

## ЛАБОРАТОРНА РОБОТА №4

### Тема: ПЕРЕВАНТАЖЕННЯ УНАРНИХ, БІНАРНИХ, ЛОГІЧНИХ ОПЕРАТОРІВ, ОПЕРАТОРІВ ПОРІВНЯННЯ

**Мета:** придбати практичні навички роботи реалізації методів перевантаження унарних, бінарних, логічних операторів, операторів порівняння.

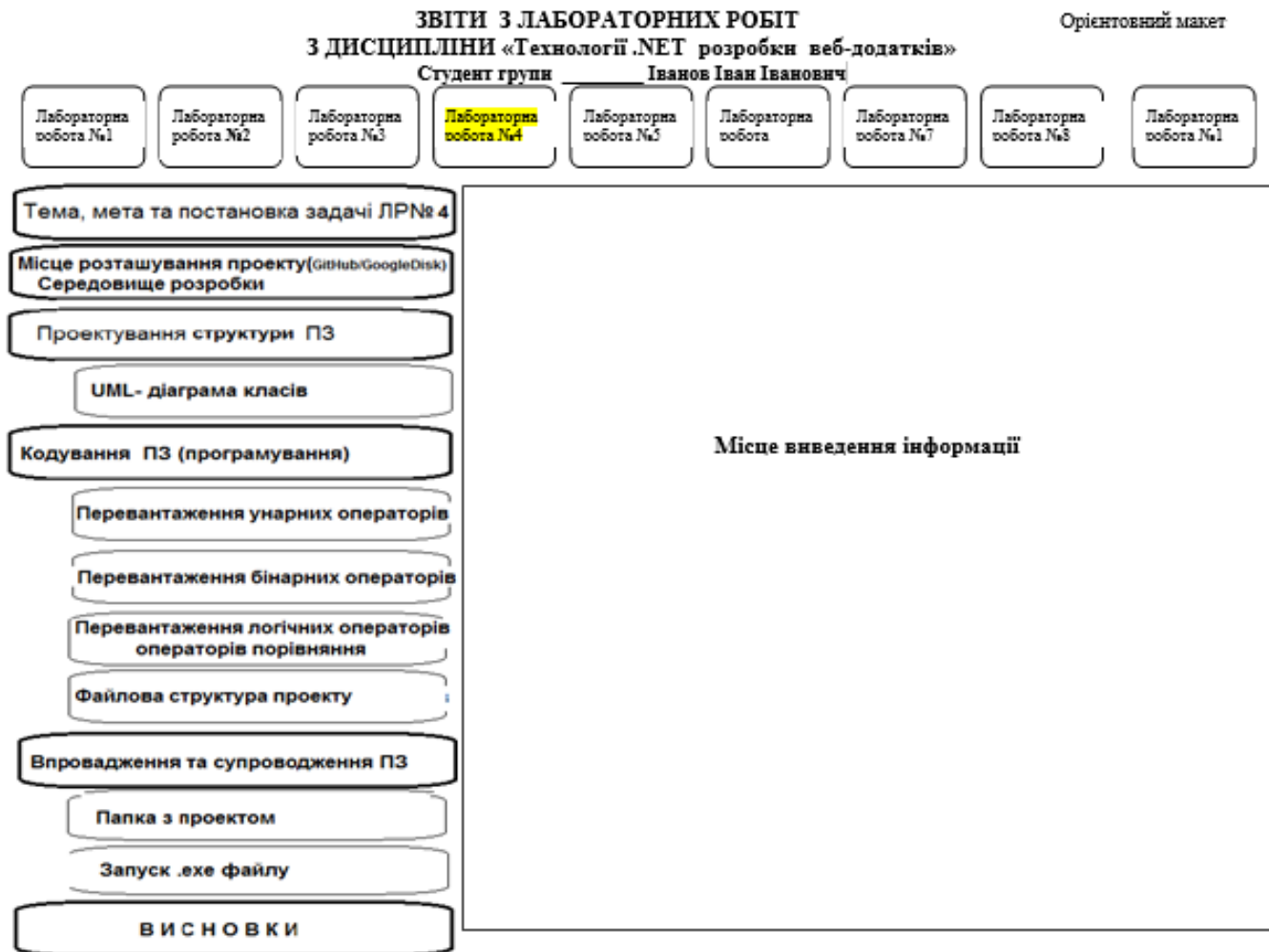
#### Постановка задачі:

1. Доповнити програмну реалізацію класів методами для перевантаження унарних, бінарних, логічних операторів, операторів порівняння.
2. Протестувати програму. Зробити висновки.

### ХІД ВИКОНАННЯ РОБОТИ

- 1 У пункті меню «Тема, мета та постановка задачі ЛР №4 » звітного HTML-документу розмістити :
  - 1.1 Тему лабораторної роботи №4
  - 1.2 Мету лабораторної роботи №4
  - 1.3 Постановку задачі лабораторної роботи №4
  - 1.4 Вказати класи та методи перевантаження унарних, бінарних, логічних операторів, операторів порівняння, які реалізуються у цих класах. Написати призначення цих методів перевантаження операторів.
- 2 У пункті меню «Місце розташування проекту. Середовище розробки» вказатим **місце розташування проекту** (шлях до папки з проектом). Це може бути GitHub або GoogleDisk. Також написати **середовище розробки програми**.
- 3 У пункті «Проектування структури ПЗ» розмістит Діаграму класів, попередньо добавивши методи перевантаження унарних, бінарних, логічних операторів
- 4 У пункті меню « Кодування ПЗ (програмування)» розмістити програмні коди та результати виконання :
  - 4.1 методів перевантаження унарних операторів.
  - 4.2 методів перевантаження бінарних операторів
  - 4.3 методів перевантаження логічних операторів та операторів порівняння
  - 4.4 показати файлову структуру проекту (скрін шот Solution explorer)..
- 5 У пункті меню «Впровадження та супроводження ПЗ» розмістити посилання на папку з проектом, усіма файлами проекту, виконувальним файлом.exe, з можливістю відкрити проекту зі звіту.
- 6 У пункті меню «ВИСНОВКИ »написати висновки по лабораторній роботі

- 7 Самостійно відпрацювати матеріал на стор.269 - 302 посібника Герберта Шилдта «С# 4.0», Полное руководство.
- 8 Відпрацювати матеріал «Перегрузка операторов» за посиланням <https://metanit.com/sharp/questions/2.15.php>
- 9 Для оформлення звітнього HTML-документу приведен орієнтовний макет.



### ПРИМІТКА:

Звіт з лабораторних робіт слід підготувати у вигляді **гіпертекстового документа у форматі html**. Документ має містити меню, яке включає команди, що подані нижче. Слід реалізувати запуск програм на виконання з гіпертекстового документа. Звіт та проекти лабораторних робіт слід записати на Гугл-Диск а. На диску має бути файл readme.txt, який містить відомості про автора звіту та проектів.

### Титульна html- сторінка.

Назва роботи  
Автор (ПІБ, група, курс, № заліковки)  
Фото  
Рік навчання

## ТЕОРЕТИЧНІ ВІДОМОСТІ

### *Перевантаження операторів*

#### питання 1

Оператор + (додавання) при застосуванні до двох об'єктах типу int повертає їх суму.

```
1  int x = 9;
2  int y = 10;
3  int z = x + y;  // 19
```

Як можна перевизначити оператор + для типу int, щоб цей оператор повертався не суму, а твір двох об'єктів int?

Відповідь

Ніяк. Ми не можемо перевизначити або перевантажити оператори для типів, які є в .NET за замовчуванням.

#### питання 2

Є клас Counter:

```
1  class Counter
2  {
3      public int Number { get; set; }
4
5      // определение оператора сложения
6  }
```

В цьому класі визначено оператор складання, який дозволяє складати об'єкт Counter з об'єктом типу int наступним чином:

```
1  Counter counter = new Counter { Number = 45 };
2  int x = counter + 6;
3  Console.WriteLine(x);  // 51
```

Який саме оператор складання визначено в класі Counter? Варіанти відповідей (правильних варіантів може бути декілька)

1.

```
1  public int operator + (Counter counter, int val)
2  {
3      return counter.Number + val;
4  }
```

2.

```
1  public static int operator + (Counter counter, int val)
```

```
2  {
3      return counter.Number + val;
4  }
```

3.

```
1  public static int operator + (int val, Counter counter)
2  {
3      return counter.Number + val;
4  }
```

4.

```
1  public static Counter operator + (Counter counter, int val)
2  {
3      return new Counter {Number = counter.Number + val};
4  }
```

5.

```
1  public static Counter operator + (int val, Counter counter)
2  {
3      return new Counter {Number = counter.Number + val};
4  }
```

Відповідь

Правильну відповідь:

```
1  public static int operator + (Counter counter, int val)
2  {
3      return counter.Number + val;
4  }
```

### питання 3

Чому не компілюється наступна програма?

```
1
2  class Counter
3  {
4      public int Number { get; set; }
5
6      public static int operator + (int val, Counter counter)
7      {
8          return counter.Number + val;
9      }
10 class Program
11 {
12     static void Main(string[] args)
13     {
14         Counter counter = new Counter { Number = 45 };
15         int x = counter + 6;
16         Console.WriteLine(x); // 51
17
18         Console.ReadKey();
19     }
20 }
```

