### Кам’янець-Подільський національний університет імені Івана Огієнка

### КАФЕДРА КОМП’ЮТЕРНИХ НАУК

### Навчальна дисципліна «Об’єктно-орієнтоване програмування»

### ЛАБОРАТОРНА РОБОТА #0109

## Тема:

Делегати і події

Варіант №2

**Виконав**:  
студент 1-го курсу  
групи KNms1-B23  
Веретко С. І.

**Прийняла**:  
старший викладач,  
Мястковська М.О.

Кам’янець-Подільський – 2023

1. Теоретичні відомості

Почнемо з визначення терміну *делегат*(delegate). Делегат – це об'єкт, який може посилатися на метод. Таким чином, створюючи делегата, ви, по суті, створюєте об'єкт, який може містити посилання на метод. Більш того, цей метод можна викликати за допомогою відповідного посилання. Таким чином, делегат може викликати метод, на який він посилається.

Делегат в C# в деякому роді аналогічний покажчику на функцію в C/C++.

***Делегат***задає визначення функціонального типу (класу) даних. Екземплярами класу є функції. Опис *делегата*в мові C# є описом ще одного окремого випадку класу. Кожен *делегат*описує множину функцій із заданою сигнатурою. Кожна функція (метод), сигнатура якої співпадає з сигнатурою *делегата*, може розглядатися як екземпляр класу, заданого *делегатом*. Синтаксис оголошення *делегата*має наступний вигляд:

[<специфікатор доступу>] delegate <тип результату > <ім'я класу> (<список аргументів>);

Цим оголошенням класу задається функціональний тип - множина функцій із заданою сигнатурою, у яких аргументи визначаються списком, заданим в оголошенні *делегата*, і тип значення, що повертається, визначається типом результату *делегата*.

Специфікатор доступу може бути, як завжди, опущений. Де слід розміщувати оголошення *делегата*? Як і у всякого класу, є дві можливості:

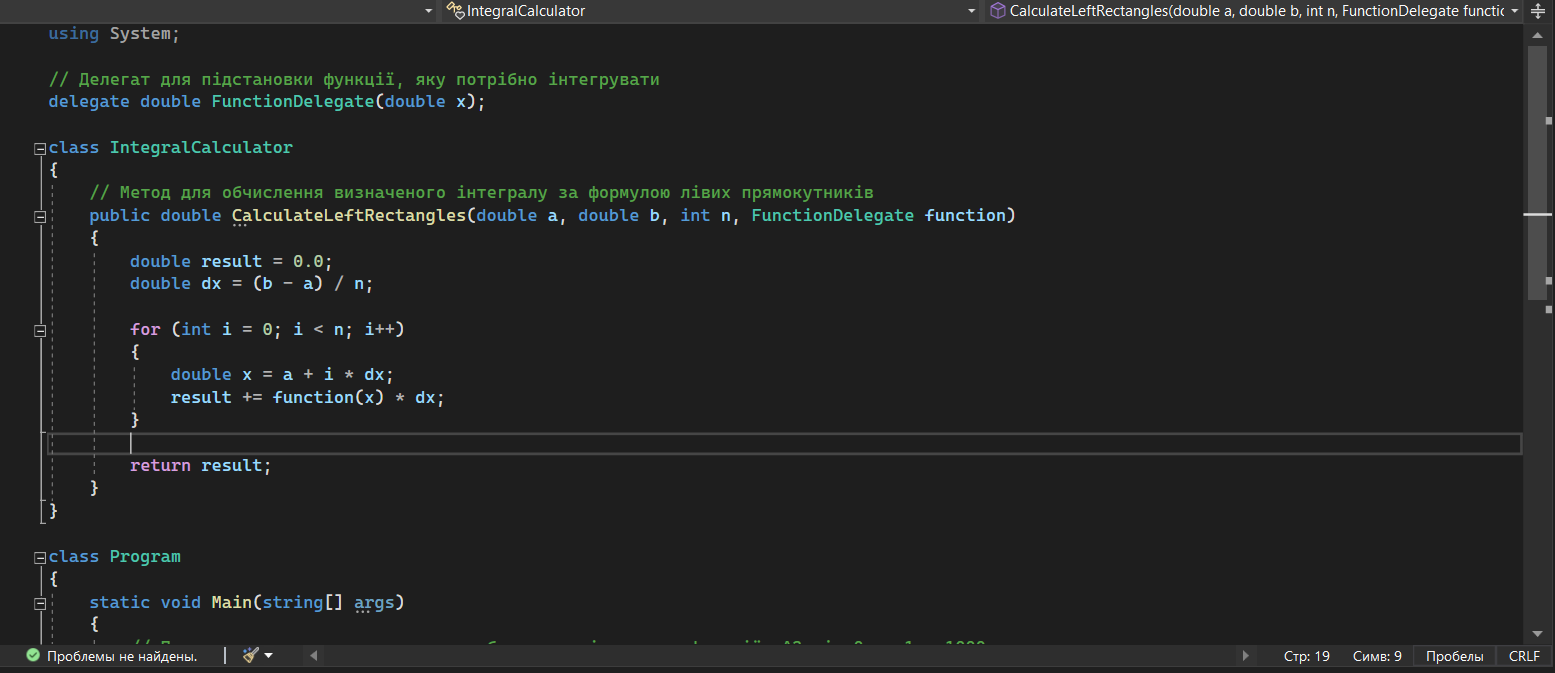
* безпосередньо у просторі імен, разом з оголошеннями інших класів, структур, інтерфейсів;
* усередині іншого класу, разом з оголошеннями методів і властивостей. Таке оголошення розглядається як оголошення вкладеного класу.

Так само, як і інтерфейси C#, делегати не задають реалізації. Фактично між деякими класами і делегатом полягає контракт на реалізацію делегата. Класи, згодні, з контрактом повинні оголосити у себе статичні або динамічні функції, сигнатура яких співпадає з сигнатурою делегата. Якщо контракт виконується, то можна створити екземпляри делегата, привласнивши їм як значення функції, що задовольняють контракту. Відмітьте, контракт є жорстким: не допускається ситуація, при якій у делегата тип параметра, - object, а у екземпляра відповідний параметр має тип, узгоджений з object, наприклад, int.

Одне з найбільш важливих застосувань *делегатів*пов'язане з *функціями вищих порядків*. ***Функцією вищого порядку***називається така функція (метод) класу, у якої один або декілька аргументів належать до функціонального типу. Без цих функцій в програмуванні обійтися досить важко. Класичним прикладом є функція обчислення інтеграла, у якої один з аргументів задає підінтегральну функцію. Іншим прикладом може служити функція, що сортує об'єкти. Аргументом її є функція Compare, що порівнює два об'єкти. Залежно від того, яка функція порівняння буде передана на вхід функції сортування, об'єкти сортуватимуться по-різному, наприклад, по імені, або по ключу, або по декількох полях.

2. Завдання для виконання

2. Розробити клас для обчислення визначених інтегралів за формулою лівих прямокутників з використанням делегатів.

3. Послідовність виконання завданнь  


4. Лістинг програми

using System;

// Делегат для підстановки функції, яку потрібно інтегрувати

delegate double FunctionDelegate(double x);

class IntegralCalculator

{

// Метод для обчислення визначеного інтегралу за формулою лівих прямокутників

public double CalculateLeftRectangles(double a, double b, int n, FunctionDelegate function)

{

double result = 0.0;

double dx = (b - a) / n;

for (int i = 0; i < n; i++)

{

double x = a + i \* dx;

result += function(x) \* dx;

}

return result;

}

}

class Program

{

static void Main(string[] args)

{

// Приклад використання класу для обчислення інтегралу функції x^2 від 0 до 1 з 1000 прямокутниками

IntegralCalculator calculator = new IntegralCalculator();

double result = calculator.CalculateLeftRectangles(0, 1, 1000, x => x \* x);

Console.WriteLine("Результат обчислення інтегралу x^2 від 0 до 1: " + result);

}

}

5. Результат роботи програми



6. [xanax4rl/lab (github.com)](https://github.com/xanax4rl/lab)