

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 2

з дисципліни “Основи програмування”

тема “Цикли та функції”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП-02  Жученко Андрій Сергійович  (*прізвище, ім’я, по батькові*)  варіант №5 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

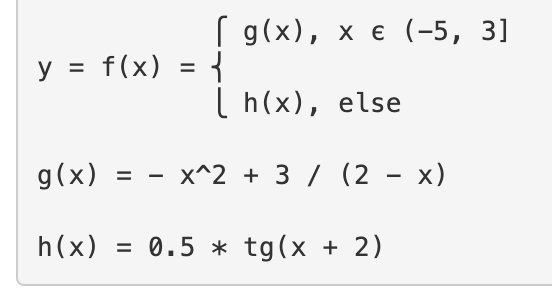
**Мета роботи**

Навчитися створювати користувацькі функції із параметрами числових типів даних.  
Застосувати циклічні конструкції для інтрегрування математичних функцій.  
Навчитись програмно перевіряти ввід користувача і сповіщати його про можливі помилки вводу.

**Постановка завдання**

#### Завдання 1

Дано кусково-задану функцію:



Програмно обчислити всі значення y при x є [-10, 10] з кроком 0.5.

#### Завдання 2. Чисельне інтегрування

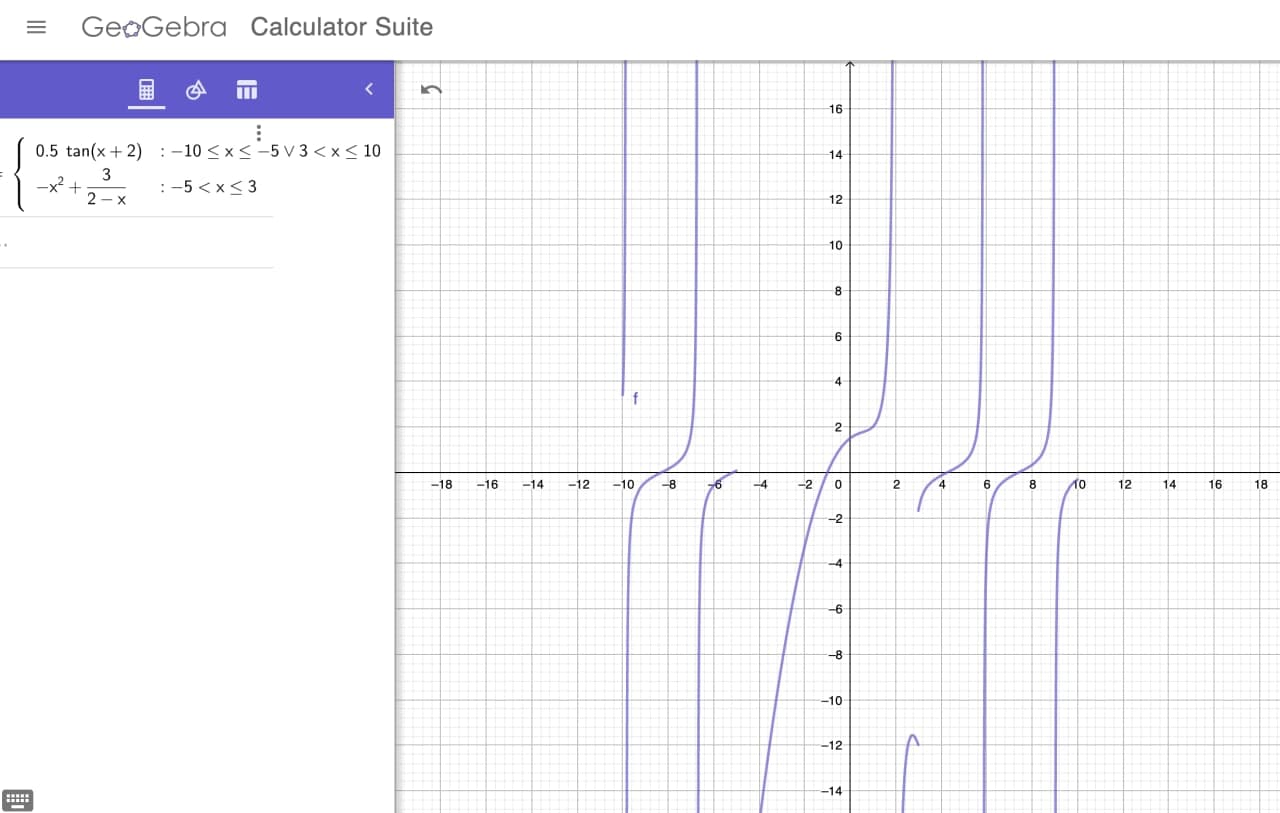
Використати математичну формулу з попередньої частини даного завдання.

Програмно обчислити інтеграл функції методом **трапецій** на проміжку від деякого xMin до xMax із кількістю кроків інтегрування nSteps.

Вивести результат обчислення у термінал.

**Графік, ОДЗ**

ОДЗ: х != 2, cos(x + 2) != 0

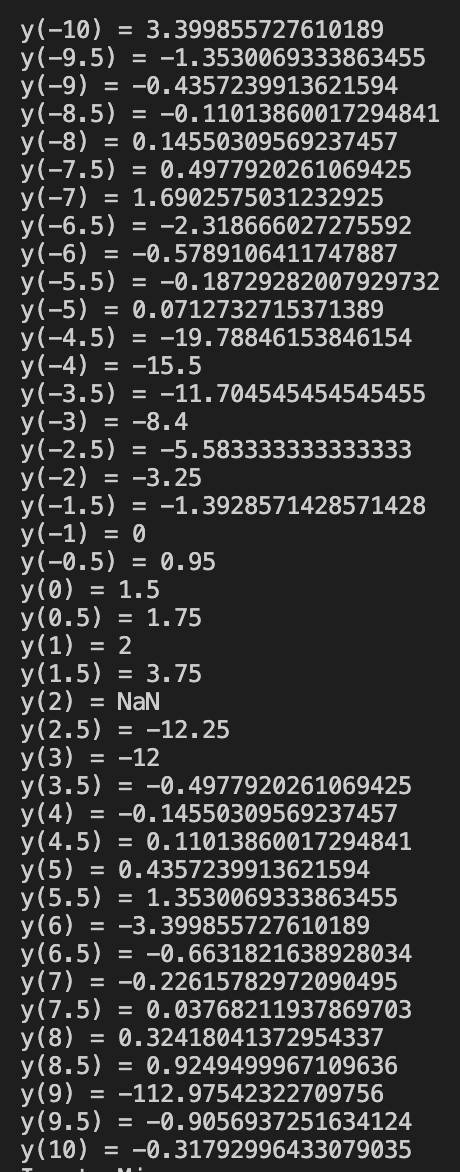
****

**Тексти коду программ**

|  |
| --- |
| **Program.cs** |
| using System;  using static System.Console;  using static System.Math;  namespace lab2  {  class Program  {  static void Main(string[] args)  {  double minPoint = -10.0;  double maxPoint = 10.0;  while (minPoint <= maxPoint)  {  WriteLine("y({0}) = {1}",minPoint, Fx(minPoint));  minPoint += 0.5;  }  WriteLine("Input xMin");  double xMin = double.Parse(ReadLine());  WriteLine("Input xMax");  double xMax = double.Parse(ReadLine());  WriteLine("Input nSteps");  int nSteps = int.Parse(ReadLine());  if (nSteps <= 0)  {  WriteLine("Wrong input for nSteps");  }  else if (xMin >= xMax)  {  WriteLine("Wrong input for xMin or/and for xMax");  }  else if((xMin <= 2 && xMax > 2) || (xMin < 2 && xMax >=2))  {  WriteLine("Impossible to count integral");  }  else // looking for the closest breakpoint to xMin for the Hx function and comparing it with xMax  {  double i = (xMin + 2 - PI / 2) / PI;  i = Ceiling(i);  double xi = PI / 2 + PI \* i - 2;  if (xi <= xMax)  {  WriteLine("Impossible to count integral");  }  else  {  WriteLine(intFx(xMin, xMax, nSteps));  }  }  }  static double Gx(double x)  {  double y = 0;  if (x == 2)  {  y = double.NaN;  }  else  {  y = -Pow(x,2) + 3 / (2 - x);  }  return y;  }  static double Hx(double x)  {  double y = 0;  if (Cos(x + 2) == 0)  {  y = double.NaN;  }  else  {  y = 0.5 \* Tan(x + 2);  }  return y;  }  static double Fx(double x)  {  double y = 0;  if (x > -5 && x <= 3)  {  y = Gx(x);  }  else  {  y = Hx(x);  }  return y;  }  static double intFx(double xMin, double xMax, int nSteps)  {  double integral = 0;  int counter = 0;  double step = (xMax - xMin) / nSteps;  while (counter <= nSteps - 1)  {  integral += Fx(xMin + counter \* step) + Fx(xMin + (counter + 1) \* step);  counter += 1;  }  integral = integral \* (step / 2);  return integral;  }  }  } |

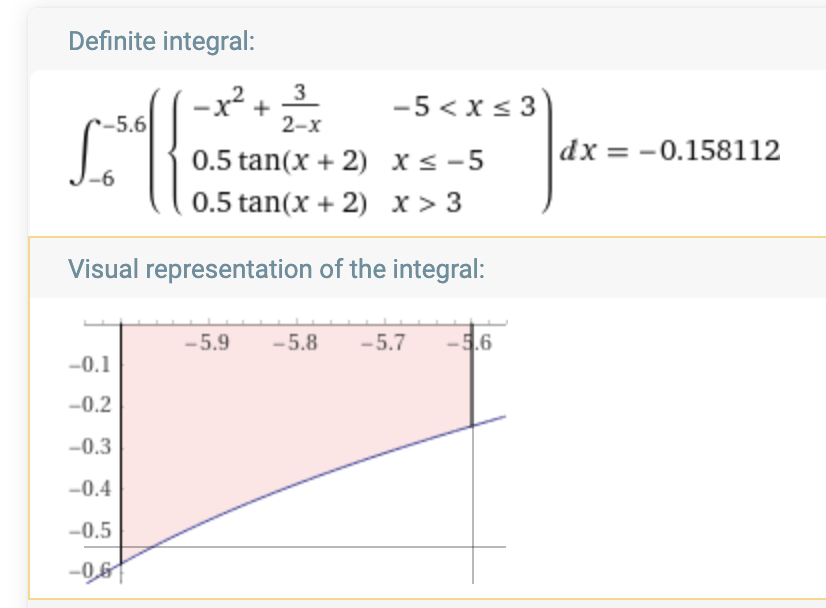
**Приклади результатів**

**Завдання 1**

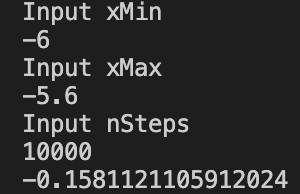
****

**Завдання 2**

**Обчислення через Вольфрам**

****

**Обчислення у програмі**

****

**Висновки**

Виконавши дану лабораторну роботу, ми навчились створювати користувацькі функції із параметрами числових типів даних, застосувавши циклічні конструкції для інтегрування математичних функції, і навчилися програмно перевіряти ввід користувача і сповіщати його про можливі помилки вводу.