

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 користувацькі функції”

|  |  |  |
| --- | --- | --- |
| Виконав  студент I курсу  групи КП-01  Беліцький Олександр Сергійович  (*прізвище, ім’я, по батькові*)  Варіант № 3 |  | 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**

**Частина 1.** Точки графіка функції.

Дано кусково-задану функцію:

⎧ g(x), x є (-5, 3]

y = f(x) = ⎨

⎩ h(x), else

g(x) = - 2 / (4 \* x - 1) - 1

h(x) = tg(x^2) + sin(2 \* x)^2

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

#### **Частина 2.**  Чисельне інтегрування.

Використати математичну формулу з попередньої частини даного завдання.

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

Вивести результат обчислення у термінал.

Користувач може задавати проміжок обчислення інтегралу та кількість кроків інтегрування у терміналі.

**Аналіз вимог і проектування**

**Частина 1.** Точки графіка функції.

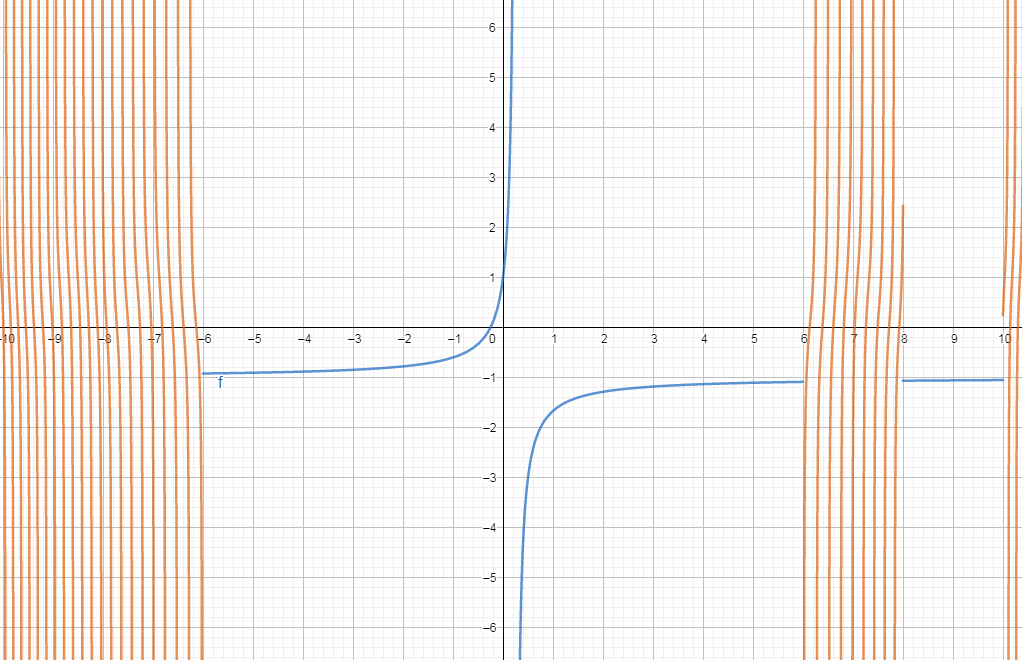
**Визначення ОДЗ:**

**1)**

*y* не існує при *х* = 0.25

**2)**

*y* не існує при або при



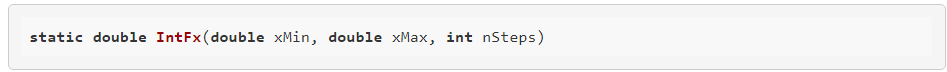
Графік *у(х)* на проміжку *х* ∈ [-10,10]

Було винесено обчислення значень g(x), h(x) та f(x) у нові програмні функції:

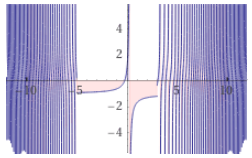


**Частина 2.** Чисельне інтегрування.

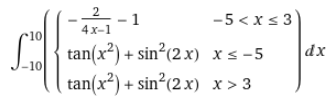
Було винесено код обчислення інтегралу у окрему функцію.

Забезпечити:

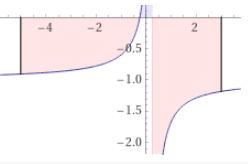
* Якщо xMin більше або рівне xMax - виводити користувачу помилку невірного вводу проміжку.
* Перевірку чи nSteps додатній і виводити помилку
* Якщо проміжок містить значення, що не входять у ОДЗ - виводити користувачу відповідну помилку.



Графік інтегрування *f(x)*



Повний інтеграл *f(x)*



Графік інтегрування *g(x)* на проміжку x є (-5, 3]



Результат обчислення *g(x)* на проміжку x є (-5, 3], але у мене буде виводити помилку, бо інший метод обчислення.

**Тексти коду програм**

**Частина 1**

|  |
| --- |
| **Program.cs** |
| using System;  using static System.Math;  using static System.Console;  namespace part1  {  class Program  {  static void Main(string[] args)  {  const double xMax = 10;  const double xMin = -10;  const double xStep = 0.5;  double x = xMin;  while (x <= xMax)  {  double y = Fx(x);  Console.WriteLine("x = {0}, y = {1}", x, y);  x = x + xStep;  }  }    static double Gx(double x)  {  double gx = -2 / (4 \* x - 1) - 1;  return gx;  }  static double Hx(double x)  {  double hx = Tan(Pow(x, 2)) + Pow(Sin(2 \* x), 2);  return hx;  }  static double Fx(double x)  {  double fx;  if ((x <= 3) && (x > -5))  {  fx = Gx(x);  }  else  {  fx = Hx(x);  }  return fx;  }  }  } |

**Частина 2**

|  |
| --- |
| **Program.cs** |
| using System;  using static System.Math;  using static System.Console;  namespace part1  {  class Program  {  static void Main(string[] args)  {  WriteLine("Enter Min: ");  double xMinn = double.Parse(ReadLine());  WriteLine("Enter Max: ");  double xMaxx = double.Parse(ReadLine());  WriteLine("Enter Steps: ");  int nSteps = int.Parse(ReadLine());  if (xMinn > xMaxx)  {  WriteLine("Change limits: Min > Max");  }  else  {  if (xMinn == xMaxx)  {  WriteLine("If Min = Max, integral will always be 0");  }  else  {  if (nSteps <= 0)  {  WriteLine("Step cannot be less than 0 or 0 ");  }  else  {  double Int = IntFx(xMinn, xMaxx, nSteps);  if (Int == 0)  {  if ((xMinn < xMaxx) && (nSteps > 0))  {  WriteLine("The space contains values that aren't included in valid range");  }  }  else  {  WriteLine("Integral by Left Rectangles : {0}", Int);  }  }  }  }  }  static double Gx(double x)  {  double gx = -2 / (4 \* x - 1) - 1;  return gx;  }  static double Hx(double x)  {  double hx = Tan(Pow(x, 2)) + Pow(Sin(2 \* x), 2);  return hx;  }    static double Fx(double x)  {  double fx;  if ((x <= 3) && (x > -5))  {  fx = Gx(x);  }  else  {  fx = Hx(x);  }  return fx;  }  static double IntFx(double xMin, double xMax, int steps)  {  double sum = 0;  double step = (xMax - xMin) / steps;  for (int i = 0; i <= steps - 1; i++)  {  double x = xMin + i \* step;  double n = (Pow(x, 2) - PI / 2) / PI;  if ((n % Floor(n) == 0) || ((x < 1) && (x > 0)))  {  return 0;  }  else  {  sum += Fx(x);  }  }  double result = step \* sum;  return result;  }  }  } |

**Приклади результатів**

**Частина 1.**

Програмне обчислення ***у*** кусково-заданої функції *f(x)*  з кроком ***х = 0.5***

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab2/part1$ dotnet run x = -10, y = 0,24625511566920188 x = -9,5, y = -1,1301744825173055 x = -9, y = -0,24701257101819196 x = -8,5, y = 0,9176540076352041 x = -8, y = 2,4307486289421814 x = -7,5, y = 0,11499714053834698 x = -7, y = -2,1916056190024076 x = -6,5, y = 6,314531782961436 x = -6, y = 8,03838140203065 x = -5,5, y = -1,3334946754680543 x = -5, y = 0,16243256207176812 x = -4,5, y = -0,8947368421052632 x = -4, y = -0,8823529411764706 x = -3,5, y = -0,8666666666666667 x = -3, y = -0,8461538461538461 x = -2,5, y = -0,8181818181818181 x = -2, y = -0,7777777777777778 x = -1,5, y = -0,7142857142857143 x = -1, y = -0,6 x = -0,5, y = -0,33333333333333337 x = 0, y = 1 x = 0,5, y = -3 x = 1, y = -1,6666666666666665 x = 1,5, y = -1,4 x = 2, y = -1,2857142857142856 x = 2,5, y = -1,2222222222222223 x = 3, y = -1,1818181818181819 x = 3,5, y = 0,10426512546288719 x = 4, y = 1,2794619821855957 x = 4,5, y = 5,983137021258485 x = 5, y = 0,16243256207176812 x = 5,5, y = -1,3334946754680543 x = 6, y = 8,03838140203065 x = 6,5, y = 6,314531782961436 x = 7, y = -2,1916056190024076 x = 7,5, y = 0,11499714053834698 x = 8, y = 2,4307486289421814 x = 8,5, y = 0,9176540076352041 x = 9, y = -0,24701257101819196 x = 9,5, y = -1,1301744825173055 x = 10, y = 0,24625511566920188 |

**Частина 2.**

Ввід значень, що входять в ОДЗ:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab2/part1$ dotnet run  Enter Min:  -8 Enter Max:  -5 Enter Steps:  10 Integral by Left Rectangles : 5,238552916899123 |

Мінімальне значення дорівнює максимальному:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab2/part1$ dotnet run  Enter Min:  5 Enter Max:  5 Enter Steps:  5 If Min = Max, integral will always be 0 |

Мінімальне значення більше максимального:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab2/part1$ dotnet run  Enter Min:  6 Enter Max:  5 Enter Steps:  5 Change limits: Min > Max |

Кількість кроків менше або дорівнює 0:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab2/part1$ dotnet run  Enter Min:  5 Enter Max:  6 Enter Steps:  0 Step cannot be less than 0 or 0 |

Значення, що не задовільняє ОДЗ:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab2/part1$ dotnet run  Enter Min:  -5 Enter Max:  3 Enter Steps:  10 The space contains values that aren't included in valid range |

**Висновки**

В даній лабораторній роботі були закріплені навички створення користувацьких функцій із параметрами числових типів даних. Виконавши дану лабораторну роботу, були покращені навички застосування циклічних конструкцій.   
 Також, за допомогою циклічних конструкцій було вивчено метод інтегрування математичних функцій, що було показано у завданні частини 2. Мова С# надає засоби достатньо точної обробки числових типів даних для широкого застосування.

В результаті виконання завдань даної лабораторної було застосовано і закріплено вміння програмно перевіряти ввід користувача і сповіщати його про можливі помилки.

Компіляція всього коду відбувалася за допомогою утиліти dotnet.