Міністерство освіти і науки України

Національний Технічний Університет

«Харківський Політехнічний Інститут»

Кафедра «Стратегічного Керування»

Лабораторна робота № 6

«Використання вказівників на функції та заголовних файлів»

Перевірила: ас. кафедри СК

Вільхівская О. В.

Виконав:

Харків, 2017

Лабораторна робота № 6

Тема: Використання вказівників на функції та заголовних файлів

Завдання

1. Метод ділення відрізку навпіл

Реалізувати програму знаходження кореня рівняння методом ділення відрізку навпіл (дихотомії) у різних одиницях трансляції. Описати та використати вказівник на функцію, тип якого, а також прототип функції знаходження кореня слід розташувати у заголовному файлі.

1. Індивідуальне завдання

Написати програму, яка реалізує перебір значень з метою пошуку деякого значення відповідно до індивідуального завдання, наведеного в таблиці. Необхідне значення може бути знайдено шляхом перевірки проміжних значень функції (або першої / другої похідної). Слід використати вказівник на функцію, для якого визначити typedef.

Сирцевий код повинен бути розділений на дві одиниці трансляції. Перша одиниця трансляції буде представлена заголовним файлом і файлом реалізації. Визначення typedef, а також прототип функції пошуку потрібного значення, повинні бути розташовані в заголовному файлі. Визначення цієї функції слід здійснити у файлі реалізації. Функція для перевірки працездатності програми, а також функція main(), повинні бути розташовані в іншій одиниці трансляції.

|  |  |
| --- | --- |
| **Номер варіанту  (номер студента у списку)** | **Правило пошуку:** |
| 2 | Мінімальне значення першої похідної |

Хід роботи

1. Метод ділення відрізку навпіл

Вміст 1.cpp:

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  #include <cmath>  #include "Functions.cpp"  using std::cout;  using std::endl;  void main()  {  cout << root(g, 0, 6) << endl;  cout << root(tan, 2, 4, 0.00001) << endl;  cout << root(my, 1, 5, 0.1) << endl;  system("pause");  } |

Вміст Functions.cpp:

|  |
| --- |
| #include "stdafx.h"  #include "Header.h"  inline double g(double x)  {  return x \* x - 2;  }  inline double my(double x) {  return x - 2;  }  inline double root(FuncType f, double a, double b, double eps)  {  double x;  do  {  x = (a + b) / 2;  if (f(a) \* f(x) > 0)  {  a = x;  }  else  {  if (f(a) \* f(x) < 0)  {  b = x;  }  else  {  return x;  }  }  } while (b - a > eps);  return x;  } |

Вміст Header.h:

|  |
| --- |
| //#pragma once  #ifndef Header\_h  #define Header\_h  typedef double(\*FuncType)(double);  double root(FuncType f, double a, double b, double eps = 0.001);  #endif |

Приклад виконання програми:

|  |
| --- |
| 1.41431  3.14159  2  Для продолжения нажмите любую клавишу . . . |

1. Індивідуальне завдання

Вміст First.cpp:

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  #include "Second.cpp"  using std::cout;  using std::endl;  void main() {  cout << "The smallest derivative = " << der\_min(parabola, -5, 5, 1) << endl;  system("pause");  } |

Вміст Second.cpp:

|  |
| --- |
| #include "stdafx.h"  #include <cmath>  #include <iostream>  #include "Header.h"  using namespace std;  inline double parabola(double x) {  return x\*x;  }  double der\_min(func f, double left, double right, double step) {  double y;  double temp;  bool mustFind = false;  double derivative;  for (double x = left; x <= right; x += step) {  y = f(x);  derivative = (f(x + 0.0000001) - f(x)) / 0.0000001;  if (!mustFind) {  temp = derivative;  mustFind = true;  }  if (derivative < temp) {  temp = derivative;  }  cout << "X = " << x << "\t Y = " << y << "\t\t Derivative = " << derivative << endl;  }  cout << endl;  return temp;  } |

Вміст Header.h:

|  |
| --- |
| //#pragma once  #ifndef Header\_h  #define Header\_h  typedef double(\*func)(double);  inline double der\_min(func f, double left, double right, double step);  #endif |

Приклад виконання програми:

|  |
| --- |
| X = -5 Y = 25 Derivative = -10  X = -4 Y = 16 Derivative = -8  X = -3 Y = 9 Derivative = -6  X = -2 Y = 4 Derivative = -4  X = -1 Y = 1 Derivative = -2  X = 0 Y = 0 Derivative = 1e-07  X = 1 Y = 1 Derivative = 2  X = 2 Y = 4 Derivative = 4  X = 3 Y = 9 Derivative = 6  X = 4 Y = 16 Derivative = 8  X = 5 Y = 25 Derivative = 10  The smallest derivative = -10  Для продолжения нажмите любую клавишу . . . |

Висновок

У C++ існує два види написання програм – з використанням однієї одиниці трансляції та з використанням багатьох. Одиниця трансляції у С++ – файл з сирцевим кодом формату \*.срр, до якого включено заголовні файли формату \*.h. Щоб не підключати заголовні файли повторно слід використовувати «стражі включення». Інша частина лабораторної – використання вказівників на функції – особливість, що стає у нагоді коли ми наприклад хочемо обчислити складну функцію (функцію від функції).