**Лабораторная работа № 1**

**Тема работы:** Разработка и отладка линейных алгоритмов и программ.

**Цель работы:** Сформировать умения разрабатывать и тестировать консольные приложения линейной структуры, выполнять их отладку.

**Оснащение рабочего места:** ПК, VS, методические указания для проведения лабораторных работ, индивидуальные задания на ЛР

**Краткие теоретические сведения**

Алгоритм – это последовательность действий, выполняемых по строго определенным правилам, однозначно определяющая процесс решения задачи и заведомо приводящая к её решению за некоторое количество шагов. алгоритмизация – разработка формального метода решения практической задачи с возможностью реализации в виде программы для эвм.

изображение алгоритма в виде схемы выполняется в соответствии с ГОСТ 19.701–90 единой системы программной документации «схемы алгоритмов, программ, данных и систем».

Структура программ для Microsoft Visual Studio.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // struct\_program.cpp: определяет точку входа для консольного приложения.  #include "stdafx.h"  //здесь подключаем все необходимые препроцессорные директивы  int main() { // начало главной функции с именем main  //здесь будет находится ваш программный код  } |

в строке 1 говорится о точке входа для консольного приложения, это значит, что данную программу можно запустить через командную строку windows указав имя программы, к примеру, такое struct\_program.cpp. строка 1 является однострочным комментарием, так как начинается с символов //. в строке 2 подключен заголовочный файл "stdafx.h". данный файл похож на контейнер, так как в нем подключены основные препроцессорные директивы (те, что подключил компилятор, при создании консольного приложения), и вспомогательные (подключенные программистом).

include — директива препроцессора, т. е. сообщение препроцессору. строки, начинающиеся с символа # обрабатываются препроцессором до компиляции программы.

заголовочные файлы:

библиотека cmath определяет набор функций для выполнения общих математических операций и преобразований. математические функции:

1) тригонометрические функции:

cos-вычисление косинуса угла, переведенного в радианы; sin-вычисление синуса угла, переведенного в радианы; tan-вычисление тангенса угла, переведенного в радианы;

acos-вычисление арккосинуса, результат будет в радианах; asin-вычисление арксинуса, результат будет в радианах;

atan-вычисление арктангенса, возвращаемый результат будет в радианах; atan2-вычисление арктангенса и квадранта по координатам x и

y, возвращаемый результат будет в радианах;

2) гиперболические функции:

cosh-вычисление гиперболического косинуса; sinh-вычисление гиперболического синуса; tanh-вычисление гиперболического тангенса;

3) экспоненциальные и логарифмические функции: exp-вычисление экспоненты;

frexp-получить мантиссу и показатель степени двойки;

modf-разделение вещественного значения на дробную и целую части;

4) функции степени:

pow-возведение числа в степень. пример использования функции; sqrt-корень квадратный;

5) округление, модуль и другие функции ceil-округление до наименьшего целого значения; fabs-вычислить модуль значения;

floor-округление до наибольшего целого значения; fmod-остаток от деления числителя на знаменатель.

**Порядок выполнения работы**

1. Ознакомиться с краткими теоретическими сведениями
2. Ознакомиться с поставленной задачей на ЛР

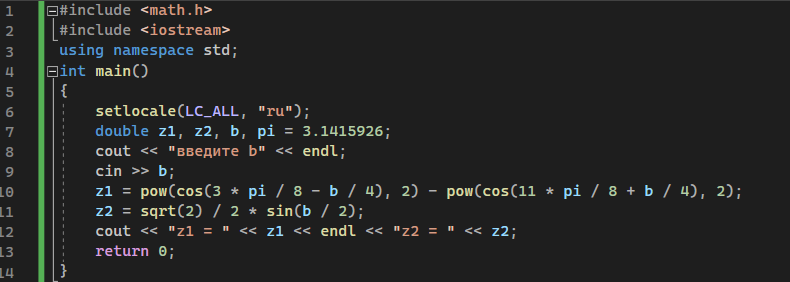
|  |  |  |  |
| --- | --- | --- | --- |
| Математическая функция | Имя функции в языке | Математическая функция | Имя функции в языке |
| √x | sqrt(x) | arcsin(x) | asin(x) |
| |x| | fabs(x) | arccos(x) | acos(x) |
| ex | exp(x) | arctg(x) | atan(x) |
| xy | pow(x,y) | arctg(x/y) | atan2(x,y) |
| ln(x) | log(x) | sh(x)=1/2 (ex-e-x) | sinh(x) |
| lg10(x) | log10(x) | ch(x)=1/2 (ex+e-x) | cosh(x) |
| sin(x) | sin(x) | tgh(x) | tanh(x) |
| cos(x) | cos(x) | Остаток от деления x на y | fmod(x,y) |
| tg(x) | tan(x) | Наименьшее целое >=x | ceil(x) |
| ctg(x) | 1/ tan(x) | Наибольшее целое <=x | floor(x) |

### Первое задание

Составить программу для расчета двух значений *z*1 и *z*2, результаты которых должны совпадать. Ввод исходных данных можно задавать при декларации или вводить с клавиатуры. Игнорировать возможность деления на ноль. Значение π = 3,1415926.

Пример:





Варианты:

Четные компьютеры: 2, 4, 6, 8

Нечетные компьютеры: 1, 3, 5, 7.

Можно выполнять в одной программе.

1. .

Вместо ¼ вводить 0.25

2. .

3. 

Вместо ¼ вводить 0.25

4. .

5. .

6.  .

7.  .

8.  .

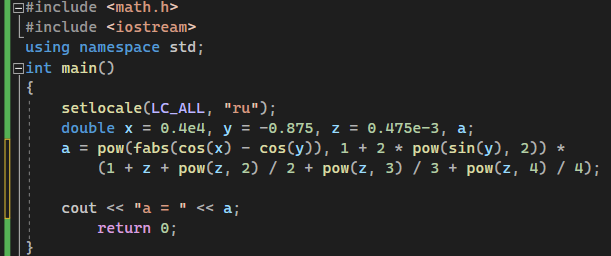
### Второе задание

Составить программу для расчета заданных выражений. Исходные данные инициализировать в программе. Значение π = 3,1415926 оформить в виде константы.

Пример:



При *x* = 0.4×104, *y* = –0.875, *z* = –0.475×10-3. Результат = 1.9873



Варианты:

Четные компьютеры:

1..

При *x* = 12.3×10-1, *y* = 15.4, *z* = 0.252×103. Результат = 82.8257

2. .

При *x* = 17.421, *y* = 10.365×10-3, *z* = 0.828×105. Результат = 0.33056

Нечетные компьютеры:

1.



При *x* = 2.444, *y* = 0.869×10-2, *z* = –0.13×103. Результат = -0.49871

2.

При *x* = 3.251, *y* = 0.325, *z* = 0.466×10-4. Результат = 4.25

1. Решить программным способом поставленную задачу
2. Составить отчёт по выполненной ЛР

**Содержание отчета:**

1. Титульный лист (см. образец отчёта)
2. Наименование и цель работы.
3. Задание на ЛР
4. Скопированный код из VS.
5. Скриншот выполненной программы
6. Ответы на контрольные вопросы.
7. Выводы по работе.