ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Лабораторная работа 1. Вычисление математических выражений |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

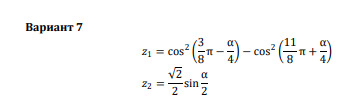
Целью работы является вычисление сложных математических выражений, а также отладка программы для поиска ошибок.

**Задание на лабораторную работу**

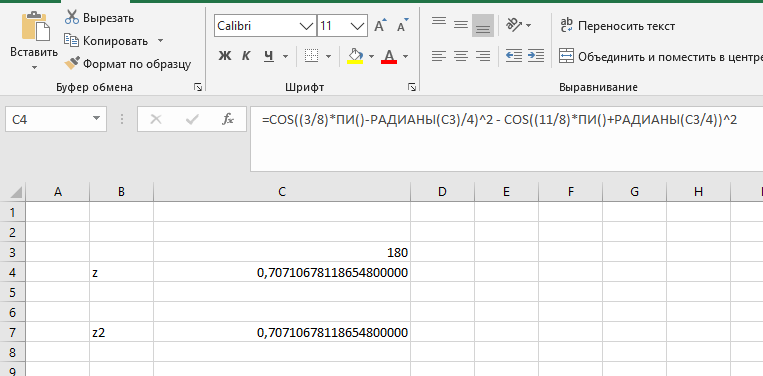
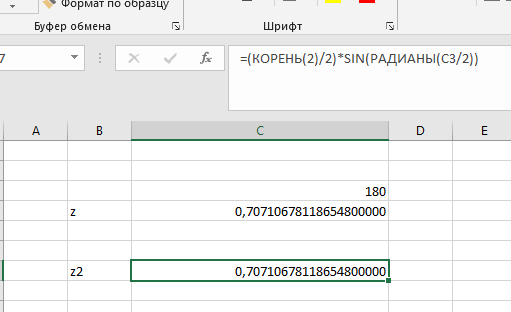
Напишите программу для расчѐта двух выражений. Предварительно подготовьте тестовые примеры по обеим формулам (в excel или с помощью калькулятора; результат вычисления по первой формуле должен совпадать со второй). Значение параметров тригонометрических функций должны задаваться пользователем в градусах.

Примечание 1: список математических функций библиотеки С++ приведѐн в «Приложение 7. Перечень функций стандартной библиотеки C++».

Примечание 2: сведения об отладке приводятся в «Приложение 9. Отладка кода проекта в среде Visual C++».



**Ход Выполнения**

1. **Тестовые примеры** 
2. 

**Используемые функции**

Printf() – форматированный вывод в консоль

Isdigit() – Фкнкция получает на вход массив чаров, после чего возвращает массив из 0 и 1. Если символ является цифрой 1, и 0 в обратном случае

Sqrt() – извлечение квадратного корня

Cin>> ввод с клавиатуры

**Имя** : z1

**Назначение** : вычислить уравнение по варианту

**Входные данные** –

angle – угол пользователя, Double

**Выходные данные** –

Результат уравнения по варианту

**Побочные эффекты отсутствуют**

|  |  |
| --- | --- |
| **Angle** | **res** |
| **180** | **0.707** |
| **30** | **0.183** |

**Псевдокод**

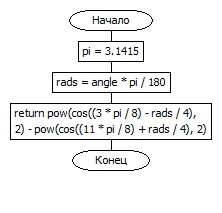
На вход дабл

Объявление констант

Перевод в радианы

Возврат значения по уравнению

**Схема**

****

**Имя** : z2

**Назначение** : вычислить уравнение по варианту

**Входные данные** –

angle – угол пользователя, Double

**Выходные данные** –

Результат уравнения по варианту

**Побочные эффекты отсутствуют**

|  |  |
| --- | --- |
| **Angle** | **res** |
| **180** | **0.707** |
| **45** | **0.27** |
| **30** | **0.183** |

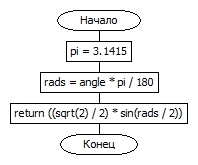
**Псевдокод**

На вход дабл

Объявление констант

Перевод в радианы

Возврат значения по уравнению

****

**Имя** : main

**Назначение** : вычислить уравнение по варианту и проверка входных данных

**Входные данные** –

input – угол пользователя, str

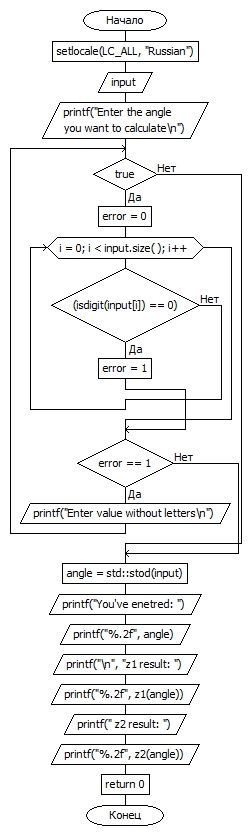
**Выходные данные** –

Результат уравнения по варианту и диалог

**Побочные эффекты отсутствуют**

|  |  |
| --- | --- |
| **input** | **res** |
| **180** | **0.707** |
| **45** | **0.27** |
| **30** | **0.183** |

**Схема**

****

**Псевдокод main**

Введите угол

Пока 1:

Ошибка = 0

Для I длинны введенного угла:

Если все символы это числа, выходим из цикла

Иначе ошибка = 1

Если ошибка = 1, введите угол заново

Если ошибка = 0, выход из цикла

Вывод диалога с пользователем и результатов вычислений

**Листинг программы**

#include <math.h>

#include <iostream>

#include <String>

double z1(const double angle) // формула 1

{

float pi = 3.1415;

double rads = angle \* pi / 180; //расчет радианной меры угла

return pow(cos((3 \* pi / 8) - rads / 4), 2) - pow(cos((11 \* pi / 8) + rads / 4), 2); //формула 1

}

double z2(double angle) //формула 2

{

float pi = 3.1415;

double rads = angle \* pi / 180; //расчет радианной меры угла

return ((sqrt(2) / 2) \* sin(rads / 2)); //формула 2

}

int main()

{

setlocale(LC\_ALL, "Russian");

std::string input;

double angle;

printf("Enter the angle you want to calculate\n"); //общение с пользователем

while (true) //проверка введенного числа

{

bool error = 0;

std::cin >> input;

for (int i = 0; i < input.size(); i++) {

if ((isdigit(input[i]) == 0)) {

error = 1;

break;

}

}

if (error == 1) {

printf("Enter value without letters\n");

}

else

{

break;

}

}

angle = std::stod(input);

printf("You've enetred: "); //вычисления

printf("%.2f", angle);

printf("\n", "z1 result: ");

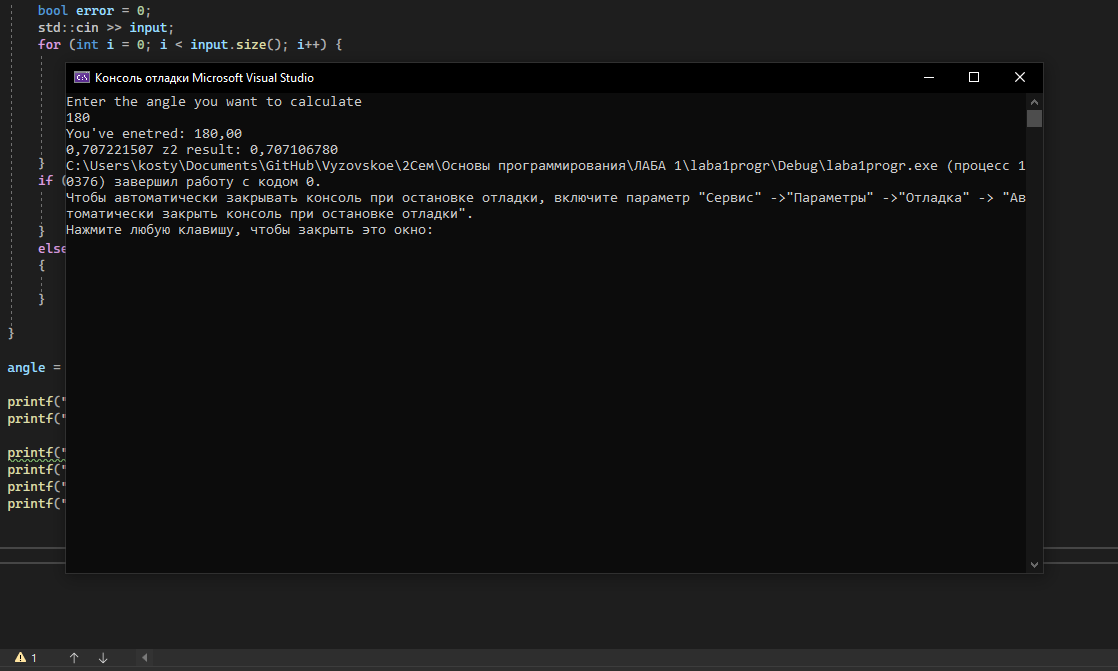
printf("%.9f", z1(angle));

printf(" z2 result: ");

printf("%.9f", z2(angle));

}

**Результат работы программы**



**Вывод:**  Я научился проводить вычисление сложных математических выражений а также производить отладку программы для поиска ошибок

Из достоинств программы можно выделить

- высокая скорость

- проверка введенных пользователем данных

Из минусов

- примерные вычисления из-за двоичного представления переменных типа double