1. **Цель работы**

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

1. **Задание**

Согласно варианту №3 необходимо вычислить математические выражения:

1. **Описание созданных функций**

**Имя:** degree2radian

**Назначение:** Перевод градусов в радианы

**Входные данные:**

* double angle – Значение угла в градусах

**Выходные данные:**

* double angle\*M\_PI/180 – Значение угла в радианах

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
| angle | angle\*M\_PI/180 |
| 90 | 1.57 |
| 0 | 0 |

**Прототип:** double degree2radian(const double angle)

**Псевдокод**

Получить угол в градусах

Умножить его на pi и разделить на 180

**Блок-схема**

angle

angle\*M\_PI/180

**Имя:** floor\_2

**Назначение:** Округление до двух знаков после запятой в меньшую сторону

**Входные данные:**

* double number – Число для округления

**Выходные данные:**

* double floor(number\*100)/100 – Округлённое число

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
| number | floor(number\*100)/100 |
| 7.777 | 7.77 |
| 0.159 | 0.15 |

**Прототип:** double floor\_2(const double number)

**Псевдокод**

Получить число для округление

Умножить его на 100

Округлить стандартной функцией floor

Разделить на 100

**Блок-схема**

number

floor(number\*100)/100

**Имя:** equation1

**Назначение:** Cчитает первое выражение согласно варианту

**Входные данные:**

* double angle – Угол в радианах

**Выходные данные:**

* double – Значение выражение   
  \*если знаменатель 0, вернёт 0.

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
| angle | Выход |
| pi/2 | -2 |
| 0 | 0 |

**Прототип:** double equation1(const double angle)

**Псевдокод**

Получить угол в радианах

Посчитать числитель с использованием floor\_2

Посчитать знаменатель с использованием floor\_2

Если знаменатель = 0, вернуть 0  
 Иначе вернёт числитель/знаменатель

**Блок-схема**

angle

a = floor\_2(sin(2\*angle) + sin(5\*angle) - sin(3\*angle));

b = floor\_2(cos(angle) - 1 + 2\*pow(sin(2\*angle),2));

b == 0

0

a/b

**Имя:** request

**Назначение:** Запрашивает угол в градусах и проверяет его на корректность

**Входные данные:**

* **-**

**Выходные данные:**

* double – Значение угла в градусах

**Побочный эффект:** Отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
| Вход | Выход |
| 90 | 90 |
| 5g | Неверное значение |

**Прототип:** double request()

**Псевдокод**

Запрос угла

Проверка корректности

Если неверно, то начать заново

Вернуть значение угла

**Блок-схема**

Привет! Введи угол в градусах!

а =

a

cin.fail() || (cin.peek() != '\n')

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(),'\n');

Неверное значение! Введи заново

a =

a

1. **Текст программы**

#define \_USE\_MATH\_DEFINES

#include<iostream>

#include<cmath>

#include <limits>

using namespace std;

double degree2radian(const double angle) // Функция перевода градусов в радианы

{

return angle \* M\_PI / 180;

}

double floor\_2(const double number) // Округление числа до двух знаков после запятой

{

return floor(number \* 100) / 100;

}

double equation1(const double angle) // Функция подсчёта первого выражения

{

double a; // Инциализация переменных

double b;

a = floor\_2(sin(2 \* angle) + sin(5 \* angle) - sin(3 \* angle)); // Присвоение значение числителя

b = floor\_2(cos(angle) + 1 - 2 \* pow(sin(2 \* angle), 2)); // Присвоение значения знаменателя

if (b == 0)

{

return 0; // Если знаметатель ноль, вернёт ноль

}

else

{

return a / b; // Иначе посчитает выражение

}

}

double request() // Функция запроса угла

{

double a;

cout << "Привет! Введи угол в градусах!" << endl;

cout << "a = ";

cin >> a;

while (cin.fail() || (cin.peek() != '\n')) // Проверка на корректный ввод

{

cin.clear(); // Очищение флага ошибки

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Очистка буфера запроса

cout << "Неверное значение! Введи заново" << endl;

cout << "a = ";

cin >> a;

}

return a;

}

int main()

{

setlocale(LC\_ALL, "rus");

double a;

a = request(); // Запрос данного

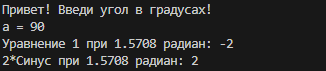
a = degree2radian(a); // Перевод в радианы

cout << "Уравнение 1 при " << a << " радиан: " << equation1(a) << endl; // Вывод значений выражений

cout << "2\*Синус при " << a << " радиан: " << 2 \* sin(a);

}

1. **Пример работы программы**

****Рисунок 1 – Пример работы программы

Полученные данные совпадают с тестовыми

1. **Анализ результатов и выводы**

В процессе лабораторной работы были изучены базовые математические функции C++, были получены начальные знания работы с функциями.

Достоинства программы:

* Некоторые пользовательские функции можно использовать в других работах
* Данные проверяются на ввод во избежание ошибок

Недостатки:

* Программа считает только те выражения, что изначально заданы.