Лаба 1

15 варик 1 задание

#include <iostream>

#include <cmath>

// Уравнение 1: x^3 + x - 2

double equation1(double x) {

return std::pow(x, 3) + x - 2;

}

// Уравнение 2: sin(x) + 0.3

double equation2(double x) {

return std::sin(x) + 0.3;

}

// Метод дихотомии

double bisectionMethod(double (\*equation)(double), double a, double b, double epsilon) {

double fa = equation(a);

double fb = equation(b);

if (fa \* fb > 0) {

std::cerr << "Ошибка: Начальные значения не обеспечивают сходимость метода дихотомии." << std::endl;

return std::numeric\_limits<double>::quiet\_NaN();

}

while ((b - a) / 2 > epsilon) {

double c = (a + b) / 2;

double fc = equation(c);

if (fc == 0.0) {

return c;

}

else if (fa \* fc < 0) {

b = c;

fb = fc;

}

else {

a = c;

fa = fc;

}

}

return (a + b) / 2;

}

int main() {

setlocale(LC\_ALL, "Russian");

double epsilon = 0.001;

// Уравнение 1: x^3 + x - 2

double root1 = bisectionMethod(equation1, -2, 2, epsilon);

// Уравнение 2: sin(x) + 0.3

double root2 = bisectionMethod(equation2, -2, 2, epsilon);

// Вывод результатов

std::cout << "Корень уравнения 1: " << root1 << std::endl;

std::cout << "Корень уравнения 2: " << root2 << std::endl;

return 0;

}

Объяснение

Конечно, давайте разберем код по шагам:

1. \*\*Подключение библиотек:\*\*

```cpp

#include <iostream>

#include <cmath>

```

Здесь мы подключаем необходимые библиотеки: `<iostream>` для ввода/вывода данных и `<cmath>` для математических операций.

2. \*\*Определение функций для уравнений:\*\*

```cpp

double equation1(double x) {

return std::pow(x, 3) + x - 2;

}

double equation2(double x) {

return std::sin(x) + 0.3;

}

```

Здесь мы определяем две функции, представляющие уравнения. `equation1` представляет уравнение `x^3 + x - 2`, а `equation2` представляет уравнение `sin(x) + 0.3`.

3. \*\*Определение метода дихотомии:\*\*

```cpp

double bisectionMethod(double (\*equation)(double), double a, double b, double epsilon) {

// ...

}

```

Здесь мы определяем функцию `bisectionMethod`, которая принимает указатель на функцию `equation`, а также интервал `[a, b]` и желаемую точность `epsilon`.

5. \*\*Применение метода дихотомии к уравнениям:\*\*

```cpp

double root1 = bisectionMethod(equation1, -2, 2, epsilon);

double root2 = bisectionMethod(equation2, -2, 2, epsilon);

```

Здесь мы вызываем метод дихотомии для каждого уравнения Результаты сохраняются в переменных `root1` и `root2`.

6. \*\*Вывод результатов:\*\*

```cpp

std::cout << "Корень уравнения 1: " << root1 << std::endl;

std::cout << "Корень уравнения 2: " << root2 << std::endl;

```

Здесь мы выводим найденные корни уравнений на экран.

7. \*\*Завершение программы:\*\*

```cpp

return 0;

```

Возвращаем 0, что означает успешное завершение программы.

2 задание

#include <iostream>

#include <bitset>

#include <vector>

// Функция для перевода чисел из десятичной системы счисления в двоичную

void decimalToBinary(const std::vector<int>& numbers) {

setlocale(LC\_ALL, "Russian");

for (int value : numbers) {

std::bitset<32> binary(value);

std::cout << "Десятичное число " << value << " в двоичной системе: " << binary << std::endl;

}

}

int main() {

// Передача вектора чисел для перевода в двоичную систему

std::vector<int> numbers = { 9, 158, 2300 };

decimalToBinary(numbers);

return 0;

}

Объяснение

Потом сама как-нибудь))