# Министерство образования и науки РФ

# Санкт-Петербургский политехнический университет Петра Великого Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

### ОТЧЕТ

по дисциплине «Практикум по программированию»

# Лабораторная работа № 4

#### Выполнил:

Студент гр. з5130902/20001

Д.Л. Рязанцев

## Проверил

Ст. преподаватель

А.М. Журавская

Санкт-Петербург 2024 г.

## Задание

В соответствии с вариантом 19:

Дана f - рядная функция, есть F - функция, взятая из стандартной математической библиотеки.

#### Необходимо:

- Написать программу для вычисления значения рядной и библиотечной функции
- Вычислить невязку (дельту) значений рядной функции и стандартной библиотечной функций по формуле, приложенной к заданию

# Код программы/Листинг программы

#### **CMakeLists.txt:**

```
cmake_minimum_required(VERSION 3.0.0)
project(lab4
    DESCRIPTION "Лабораторная работа 4"
    HOMEPAGE_URL "github.com/whs31/education"
    LANGUAGES CXX
)

set(CMAKE_CXX_STANDARD 20)

add_executable(${PROJECT_NAME})
target_sources(${PROJECT_NAME} PRIVATE main.cc)
target_include_directories(${PROJECT_NAME} PRIVATE ${CMAKE_CURRENT_SOURCE_DIR})
```

#### main.cc:

```
/*
 * Лабораторная работа 4
 * Студент: Рязанцев Дмитрий
#include <cmath>
#include <array>
#include <iostream>
#include <format>
#include <numbers>
using f64 = double;
namespace terminal::colors
  using namespace std::literals;
  auto constexpr bold = "\x1b[1m"sv;
  auto constexpr red = "\x1b[31m"sv;
  auto constexpr magenta = "\x1b[35m"sv;
  auto constexpr blue = \sqrt{x1b[34m"sv;}
  auto constexpr green = "\x1b[32m"sv;
  auto constexpr reset = "\x1b[0m"sv;
namespace
 using std::numbers::pi;
  [[nodiscard]] auto fn a(f64 const x) -> f64 { return pi * x; }
  [[nodiscard]] auto fn_b(f64 const x, f64 const epsilon) -> f64 {
    using std::numeric_limits;
    return 1 / (pi * x) + 2 * x / pi * [x, epsilon]() \rightarrow f64 {
      auto expr = [x](auto const i) -> f64 { return 1 / (x * x - i * i); };
      auto prev = 0.0;
      auto sum = 0.0;
      for(auto i = 1.0; true; i += 1.0) { // NOLINT(*-flp30-c)
        auto const current = expr(i);
        if(std::abs(prev - current) < epsilon)</pre>
          return sum;
        prev = current;
        sum += current;
      return 0.0;
    }();
  }
  [[nodiscard]] auto delta(f64 const a, f64 const b) -> f64 {
    using std::sqrt;
    using std::abs;
```

```
return sqrt(abs(a * a - b * b));
 }
}
auto main() -> int
 using std::cin;
 using std::cout;
 using std::cerr;
 using std::endl;
 using std::format;
 auto receive_f64 = [](std::string_view const name) -> f64 {
   f64 result:
   cout << format("Please, enter variable {}{}{}:\n",</pre>
     terminal::colors::blue,
     terminal::colors::bold,
     name,
     terminal::colors::reset
   if(cin >> result)
     return result;
     cerr << format("{}Wrong input: {}{}\n",</pre>
       terminal::colors::red,
       name,
       terminal::colors::reset
     );
     std::exit(1);
 };
 auto const x_start = receive_f64("X (start)");
 auto const x_end = receive_f64("X (end)");
 auto const x_delta = receive_f64("Delta");
 cout << format("Result table (1){}:\n", terminal::colors::blue);</pre>
 cout << format("| x | f(x) | F(x) | delta |\n");
 cout << format("|-----|\n");
 for(auto x = x_{start}; x <= x_{end}; x += x_{delta}) { // NOLINT(*-flp30-c)
   cout << format("|{:<9.2}|{:<12.5}|{:<12.5}|\n",
     Χ,
     ::fn_a(x),
     ::fn_b(x, std::numeric_limits<float>::epsilon() * 2.0),
     :::delta(::fn_a(x), ::fn_b(x, std::numeric_limits<float>::epsilon() * 2.0))
   );
  }
 cout << format("|------|{}\n\n",</pre>
   terminal::colors::reset
  );
```

```
auto constexpr epsilons = std::array{ 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7
};
 auto const x_ideal = receive_f64("X (ideal)");
 cout << format("Result table (2){}:\n", terminal::colors::magenta);</pre>
 cout << format(" | eps | f(x) | F(x) | delta |\n");
 cout << format("|-----|\n");</pre>
 for(auto const epsilon : epsilons) { // NOLINT(*-flp30-c)
   cout << format("|{:<9.2}|{:<12.5}|{:<12.5}|\n",</pre>
     epsilon,
    ::fn_a(x_ideal),
     ::fn_b(x_ideal, epsilon),
     ::delta(::fn_a(x_ideal), ::fn_b(x_ideal, epsilon))
   );
 cout << format("|-----|\n\n");</pre>
 cout << format("{}Done.{}\n", terminal::colors::green, terminal::colors::re-</pre>
set);
 return 0;
```

# Пример работы программы

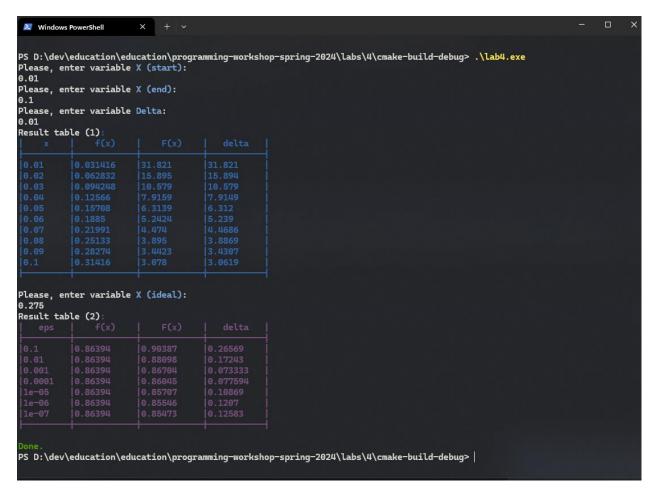


Рисунок 1 – Правильная работа программы

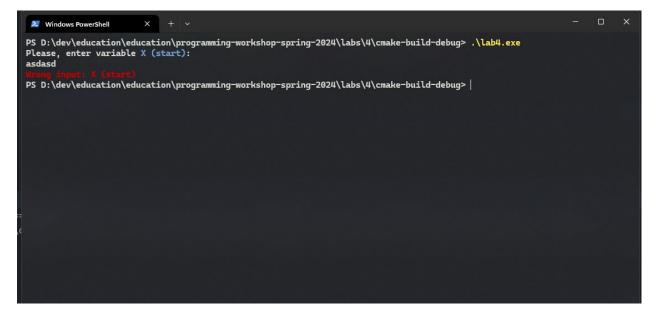


Рисунок 2 – Вывол программы при попытке ввести некорректное значение

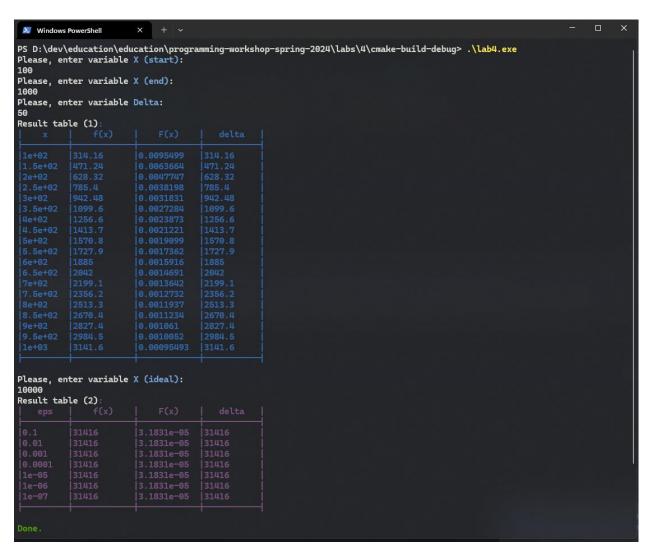


Рисунок 3 — Вывод программы при попытке ввести значения, на которых функция стремится к нулю

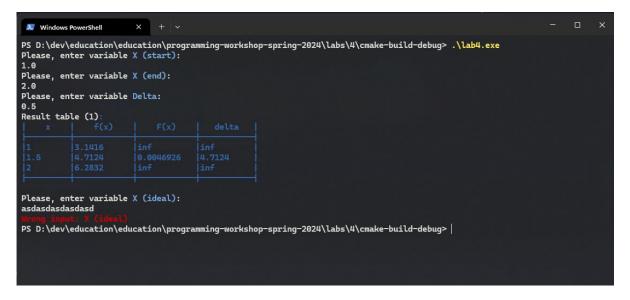


Рисунок 4 – Сообщение об ошибке при вводе некорректного x ideal

## Вывод

В ходе выполнения лабораторной работы я выполнил поставленную задачу по работе с файлами на языке С++. Получены навыки работы с математическими рядами в С++, простыми математическими функциями и циклами; навыки форматирования вывода с использованием std::format и универсальных цветовых кодов ANSI, ввод и вывод в окне терминала с использованием потоковых классов стандартной библиотеки.