МИНЕСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное учреждение высшего образования

«Воронежский государственный технический университет»

**Факультет** информационных технологий и компьютерной безопасности

**Кафедра** информационных систем и технологий цифровизации

Курсовой проект

По дисциплине: Основы программирования и алгоритмизации

Тема: «»

Выполнил студент: Колесников В. А.

Группа: ИСТ-201

Руководитель: доцент, к. т. н. Минакова О.В.

Допущен к защите « » \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Работа защищена « » \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

С оценкой \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(подпись)

Воронеж 2020

Содержание

[Содержание 2](#_Toc59519981)

[**Введение** 3](#_Toc59519982)

[**1. Постановка задачи** 4](#_Toc59519983)

[**2. Конструирование программы** 5](#_Toc59519984)

[**3. Тестирование программы** 8](#_Toc59519985)

[**Заключение** 12](#_Toc59519986)

[**Список литературы** 13](#_Toc59519987)

[**Приложения** 14](#_Toc59519988)

# **Введение**

Целью курсового проекта является закрепление изученного материала в ходе разработки программы – табулятора значений функций с параметрами, задаваемыми пользователем с интерактивного консольного интерфейса.

Основные задачи, которые необходимо решить в рамках проектирования:

1. Реализовать два элементарных программных модуля для вычисления двух функций.
2. Реализовать модуль табуляции значений выбранной пользователем функции для заданного шага d(x) и интревала [xmin - xmax].
3. Создать интерфейс программы (реализованный функции main), который должен обеспечивать выбор одной из трёх функций, задание шага и интервала табуляции, возможность выбора и указание потока вывода результатов вычислений (консоль или файл).

Функции для вычисления:

## **1. Постановка задачи**

Для решения поставленной задачи необходимо реализовать следующие действия:

1. Выбор одной из двух функций
2. Выбор потока вывода (на экран или в файл)
3. Ввод табличных параметров (диапазон и шаг изменения аргумента)
4. Вычисление значений и вывод результатов

Интерфейс всей программы будет реализовываться в виде меню, в форме диалога с пользователем (рисунок 1).

Выбор функции (1 или 2)

Выберите устройство вывода:

1) экран 2) файл

Ввод имени файла

Ввод параметров таблицы

(интервал и шаг аргумента)

Вывод таблицы значений

аргумента и функции

Рисунок 1 – Структурная схема диалога с пользователем

## **2. Конструирование программы**

Прежде чем приступить к написанию программы, необходимо разработать её архитектуру. Для этого необходимо разбить её на неделимые логические части и продумать их взаимосвязи. Это поможет сократить время на собственно написание программы, а также избежать излишнего дублирования кода.

По условию необходимо запросить у пользователя выбор одной из функций, устройства вывода и параметров таблицы. Очевидно, что за взаимодействие с пользователем будет отвечать один модуль, реализуем это в основной функции программы.

Далее, понадобятся сами функции, вычисляющие значения переданного им аргумента - f(x) и y(x), они будут помещены в отдельные программные модули.

И, наконец, вывод таблицы будет одинаковым для любой выбранной функции. Этот модуль должен перебрать в цикле все значения аргумента в заданном интервале, для каждого из них посчитать значение функции и вывести все значения в виде таблицы. Конкретная математическая функция для вычисления будет передаваться извне, в качестве параметра.

Все программные модули будут представлены в виде отдельных функций, описание которых представлено в таблице 1.

Таблица 1 – Описание собственных функций

|  |  |  |
| --- | --- | --- |
| Описание собственных функция | | |
| Наименование | Назначение | Возвращаемое значение |
| input | Печать запроса и ввод пользовательских данных с консоли. | Введённое пользователем значение. |
| tab | Запрашивает у пользователя диапазон значений аргумента, шаг его изменения и выводит таблицу значений переданной в качестве аргумента функции. | - |
| fx | Вычисляет значение функции f(x) = x + sin(x) | Значение функции для переданного аргумента |
| yx | Вычисляет значение функции y(x) | Значение функции для переданного аргумента |
| main | Взаимодействие с пользователем — запрос действий и вызов соответствующих функций, с выводом результатов на экран либо в файл. | - |

Универсальная функция input принимает в качестве параметров потоки ввода и вывода, с которыми она будет работать, значение по умолчанию, а также необязательный параметр запроса на ввод.

Если этот параметр не пустой, функция выводит его на экран и ждёт пользовательского ввода, затем преобразует введённую строку в заданный тип и возвращает это значение.

Функция tab принимает поток вывода, и функцию, результаты которой нужно вывести в таблицу. Запрашивает у пользователя параметры — диапазон и шаг изменения аргумента, после этого выводит таблицу значений аргумента и функции в поток вывода (экран либо файл).

Функция fx вычисляет для переданного ей аргумента значение первой функции задания f(x).

Функция yx вычисляет для переданного ей аргумента значение первой функции задания y(x).

Функция main реализует взаимодействие с пользователем — запрашивает нудные параметры и, в зависимости от введённых значений, вызывает нужные функции вычислений и вывода результатов.

## **3. Тестирование программы**

Для проверки работоспособности программы поочерёдно вводим тестовые значения для проверки возможных ветвлений пользовательского меню.

В задании требуется табулировать значения двух функций, и при запуске программа просит ввести номер функции для вычислений, затем поток вывода и параметры таблицы значений:

* выбираем функцию №1;
* выбираем вывод на экран;
* вводим диапазон [-5.5, 6] и шаг 0.5.

Результат работы программы для этого случая показан на рисунке 2.

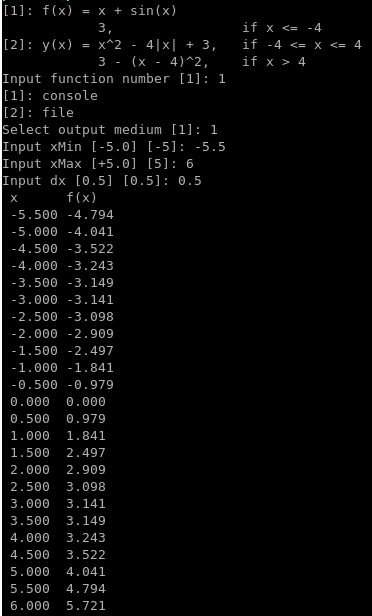


Рисунок 2 – Таблица значений функции f(x), вывод на экран

В угловых точках (0 и примерные значения -π, -π/2, π/2 π, ) тригонометрическая функции синус принимает верные значения (не забываем учесть слагаемое в виде самого аргумента).

Проверяем вторую функцию:

* выбираем функцию №2;
* выбираем вывод на экран;
* вводим диапазон [-4.5, 5] и шаг 0.25.

Результат работы программы для этого случая показан на рисунке 3.

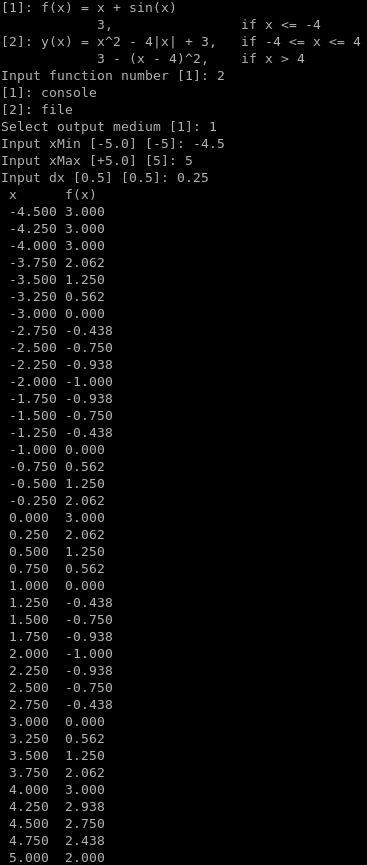


Рисунок 3 – Таблица значений функции y(x), вывод на экран

По таблице значений видно, что заданная функция y(x) на тестовом интервале вычисляется верно.

Далее проверим вывод в файл на любой из функций. Выберем первую из них:

* выбираем функцию №1;
* выбираем вывод в файл;
* вводим имя файла: fx.txt;
* вводим диапазон [-3.14, 3.14] и шаг 0.5;
* выводим на экран содержимое файла.

Результат работы программы для этого случая показан на рисунке 4.

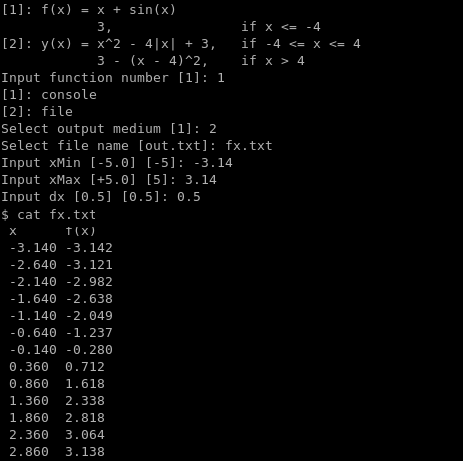


Рисунок 4 – Таблица значений функции f(x), вывод в файл

Как видно из рисунка, вывод работает точно также, как и вывод на экран.

# **Заключение**

В ходе курсового проектирования на языке С++ с использованием среды разработки Visual Studio 2017 была разработана программа для табуляции значений выбранной пользователем функции для заданного шага и интервала значений.

Все требования к программе реализованы полностью. разработан интерфейс программы взаимодействия с пользователем, а также реализован вывод результатов вычислений на экран либо в заданный пользователем файл.

# **Список литературы**

1. Курипта О.В. Основы программирования и алгоритмизации: практикум / О.В.Курипта, О.В. Минакова, Д.К. Проскурин; Воронежский ГАСУ. – Воронеж, 2015. – 132 с.
2. Неземский В.И. Процедуры и функции [Электронный ресурс]: методические указания/ Неземский В.И., Орешкина О.А.— Электрон. текстовые данные. — М.: Московский государственный технический университет имени Н.Э. Баумана, 2009.— 28 c.
3. Алексеев Е. Р. Программирование на языке С++ в среде Qt Creator:/ Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк, О. В. Чеснокова А.С. Чмыхало – М. : ALT Linux, 2015. – 448 с.
4. Конова Е. А. Алгоритмы и программы. Язык С++: учебное пособие 2-е изд./ Поллак Г. А — СПб.: Издательство «Лань», 2017. —384 с.
5. Шишкин А.Д. Программирование на языке Си [Электронный ресурс]: учебное пособие/ Шишкин А.Д.— Электрон. текстовые данные.— СПб.: Российский государственный гидрометеорологический университет, 2003.— 104 c.
6. Кузнецов М.В. С++ мастер-класс в задачах и примерах/ М.В. Кузнецов, И.В. Симдянов. – СПб.: БХВ-Петербург, 2007.—480с.

# **Приложения**

**Модуль вычисления f(x)**

**fx.h:**

#pragma once

// модуль для вычисления функции f(x)

double fx(double x);

**fx.cpp:**

#include "fx.h"

#include <cmath>

double fx(double x) {

return x + sin(x);

}

**Модуль вычисления y(x)**

**yx.h:**

#pragma once

// модуль для вычисления функции y(x)

double yx(double x);

**yx.cpp:**

#include "yx.h"

#include <cmath>

double yx(double x) {

double y = 0.0;

if(x <= -4.0)

return 3;

else if(x <= 4)

return x \* x - 4 \* std::abs(x) + 3;

double t = x - 4;

return 3 - t \* t;

}

**Модуль табуляции значений функции**

**tab.h:**

#pragma once

#include <functional>

// модуль для вычисления функции f(x)

void tab(std::ostream& os, const std::function<double(double)>& fn);

**tab.cpp:**

#include "tab.h"

#include "input.h"

#include <iomanip>

#include <iostream>

void tab(std::ostream& os, const std::function<double(double)>& fn) {

auto xMin = input(std::cin, std::cout, -5.0, "Input xMin [-5.0]");

auto xMax = input(std::cin, std::cout, +5.0, "Input xMax [+5.0]");

auto dx = input(std::cin, std::cout, 0.5, "Input dx [0.5]");

os << " x\tf(x)\n";

for(auto x = xMin; x <= xMax; x += dx) {

os << " " << std::fixed << std::setprecision(3) << x << '\t' << fn(x) << std::endl;

}

os.flush();

}

**Вспомогательный модуль универсального ввода**

**input.h**

#pragma once

#include <ostream>

#include <istream>

#include <sstream>

template<typename T>

T input\_impl(std::istream& is, std::ostream& os, const T\* def\_val, const char\* prompt) {

if(prompt) {

os << prompt;

if(def\_val)

os << " [" << \*def\_val << "]";

os << ": ";

}

T value;

std::string s;

if((is >> s) && s.size() > 0) {

std::istringstream iss(s);

iss >> value;

return value;

}

if(def\_val) {

return \*def\_val;

}

return T();

}

template<typename T>

T input(std::istream& is, std::ostream& os, const char\* prompt = nullptr) {

return input\_impl(is, os, static\_cast<T\*>(nullptr), prompt);

}

template<typename T>

T input(std::istream& is, std::ostream& os, const T& def\_val, const char\* prompt = nullptr) {

return input\_impl(is, os, &def\_val, prompt);

}

**Основной модуль с интерфейсом программы**

**functions.cpp:**

#include "fx.h" // модуль для вычисления функции f(x)

#include "yx.h" // модуль для вычисления функции y(x)

#include "tab.h" // модуль абуляции значений

#include "input.h"

#include <iostream>

#include <fstream>

#include <cmath>

std::ostream& get\_ostream() {

std::cout << "[1]: console\n";

std::cout << "[2]: file\n";

auto choice = input(std::cin, std::cout, 1, "Select output medium");

if(choice == 1)

return std::cout;

static std::ofstream file;

if(!file.is\_open()) {

auto name = input(std::cin, std::cout, std::string("out.txt"), "Select file name");

file.open(name);

}

return file;

}

int main() {

// интерфейс программы

std::cout << "[1]: f(x) = x + sin(x)\n";

std::cout << " 3, if x <= -4\n";

std::cout << "[2]: y(x) = x^2 - 4|x| + 3, if -4 <= x <= 4\n";

std::cout << " 3 - (x - 4)^2, if x > 4\n";

auto fn = input(std::cin, std::cout, 1, "Input function number");

auto& output = get\_ostream();

if(!output) {

std::cerr << "invalid stream" << std::endl;

return -1;

}

switch(fn) {

case 1:

tab(output, fx);

break;

case 2:

tab(output, yx);

break;

}

return 0;

}