Федеральное государственное бюджетное образовательное учреждение высшего образования "Национальный исследовательский университет "МЭИ"

ОТЧЕТ

По лабораторной работе № 3

«Функции и структуры»

ФИО студента Абдурахманова Регина Абдурауфовна

Номер группы А-01-18

Имя преподавателя Мохов Андрей Сергеевич

Василькова Полина Денисовна

*Цель работы*

1. Уметь структурировать программу при помощи функций и структур.
2. Уметь писать модульные тесты.

Необходимо выделить части программы в функции:

1. Ввод чисел:

* принимает количество чисел, которое необходимо ввести;
* возвращает вектор чисел.

1. Поиск наибольшего и наименьшего значения:

* принимает вектор чисел;
* возвращает два результата — min и max.

1. Расчет гистограммы:

* принимает вектор чисел и количество корзин;
* возвращает вектор количеств чисел в каждой корзине;
* вызывает в процессе работы функцию поиска min и max.

Вариант 1

Дайте пользователю возможность задавать произвольную ширину гистограммы вместо 80 символов. Ширину менее 7, более 80 или менее трети количества чисел считайте некорректной — предлагайте пользователю ввести ее заново в этом случае с указанием причины.

*Код файла main.cpp:*

#include <iostream>

#include <vector>

#include "histogram.h"

#include "svg\_func.h"

using namespace std;

vector<double> input\_numbers(size\_t count) {

vector<double> result(count);

for (size\_t i = 0; i < count; i++) {

cin >> result[i];

}

return result;

}

vector<size\_t> make\_histogram(vector<double> numbers,size\_t count){

double min = numbers[0];

double max = numbers[0];

find\_minmax(numbers, min, max);

vector<size\_t> bins(count);

for (double number : numbers) {

size\_t bin = (size\_t)((number - min) / (max - min) \* count);

if (bin == count) {

bin--;

}

bins[bin]++;

}

return bins;

}

void show\_histogram\_txt(vector<size\_t> bins, size\_t width){

const size\_t SCREEN\_WIDTH = 80;

const size\_t MAX\_ASTERISK = SCREEN\_WIDTH - 4 - 1;

size\_t max\_count = 0;

for (size\_t count : bins) {

if (count > max\_count) {

max\_count = count;

}

}

const bool scaling\_needed = max\_count > width;

for (size\_t bin : bins) {

if (bin < 100) {

cout << ' ';

}

if (bin < 10) {

cout << ' ';

}

cout << bin << "|";

size\_t height = bin;

if (scaling\_needed) {

size\_t max\_count = bins[0];

for (size\_t bin : bins)

{

if (bin > max\_count)

{

max\_count = bin;

}

}

for (size\_t bin : bins)

{

height = width \* (static\_cast<double>(bin) / max\_count); //max\_asterisk

}

}

for (size\_t i = 0; i < height; i++) {

cout << '\*';

}

cout << '\n';

}

}

int main() {

// Ввод данных

size\_t number\_count;

cerr << "Enter number count: ";

cin >> number\_count;

cerr << "Enter numbers: ";

const auto numbers = input\_numbers(number\_count);

size\_t bin\_count;

cerr << "Enter column count: ";

cin >> bin\_count;

// Обработка данных

vector<size\_t> bins = make\_histogram(numbers, bin\_count);

size\_t H\_Width; //выбранная пользователем ширина гистограммы

cerr << "Enter histogram width: \n";

cin >> H\_Width;

while ((H\_Width < 7) || (H\_Width > 80) || (H\_Width < number\_count/3 )) {

if (H\_Width < 7) {

cerr << "histogram's width is less than 7. Enter again. \n";

}

if (H\_Width > 80) {

cerr << "histogram's width is more than 80. Enter again. \n";

}

if (H\_Width < number\_count / 3) {

cerr << "histogram's width is less than third of number count. Enter again. \n";

}

cerr << "Enter histogram width: \n";

cin >> H\_Width;

}

// Вывод данных

//show\_histogram\_txt(bins);

show\_histogram\_svg(bins, H\_Width);

return 0;

}

*Код файла histogram.h:*

#ifndef HISTOGRAM\_H\_INCLUDED

#define HISTOGRAM\_H\_INCLUDED

#include <vector>

using namespace std;

void find\_minmax(const vector<double> numbers, double& min, double& max) ;

#endif // HISTOGRAM\_H\_INCLUDED

*Код файла histogram.cpp:*

#include "histogram.h"

void find\_minmax(const vector<double> numbers, double& min, double& max) {

if (numbers.size() != 0)

{

min = numbers[0];

max = numbers[0];

for (double number : numbers) {

if (number < min) {

min = number;

}

if (number > max) {

max = number;

}

}

}

}

*Код файла svg\_func.h:*

#ifndef SVG\_FUNC\_H\_INCLUDED

#define SVG\_FUNC\_H\_INCLUDED

#include <iostream>

#include <vector>

using namespace std;

void svg\_begin(double width, double height);

void svg\_end();

void svg\_text(double left, double baseline, string text);

void svg\_rect(double x, double y, double width, double height, string stroke = "black", string fill = "black");

void show\_histogram\_svg(const vector<size\_t>& bins, size\_t width);

#endif // SVG\_FUNC\_H\_INCLUDED

*Svg\_func.cpp*

#include "svg\_func.h"

void svg\_begin(double width, double height){

cout << "<?xml version='1.0' encoding='UTF-8'?>\n";

cout << "<svg width='" << width << "' height='" << height << "' "

<< "viewBox='0 0 " << width << " " << height << "' "

<< "xmlns='http://www.w3.org/2000/svg'>\n";

}

void svg\_end(){

cout << "</svg>\n";

}

void svg\_text(double left, double baseline, string text){

cout << "<text x='"<< left << "' y='"<< baseline << "'>" << text << "</text>";

}

void svg\_rect(double x, double y, double width, double height, string stroke, string fill){

cout << "<rect x='"<< x << "' y='"<< y << "' width='" << width << "' height='" << height <<"' stroke='"

<< stroke << "' fill='" << fill <<"' />";

}

void show\_histogram\_svg(const vector<size\_t>& bins, size\_t width){

const auto IMAGE\_WIDTH = 400;

const auto IMAGE\_HEIGHT = 300;

const auto TEXT\_LEFT = 20;

const auto TEXT\_BASELINE = 20;

const auto TEXT\_WIDTH = 50;

const auto BIN\_HEIGHT = 30;

svg\_begin(IMAGE\_WIDTH, IMAGE\_HEIGHT);

double top = 0;

size\_t max\_count = 0;

for (size\_t bin : bins) {

if (bin > max\_count){

max\_count = bin;

}

}

const bool scaling\_needed = (max\_count > width);

for (size\_t bin : bins) {

size\_t height = bin;

if (scaling\_needed) {

const double scaling\_factor = (double)width / max\_count;

height = (size\_t)(bin \* scaling\_factor);

}

const double bin\_width = 10 \* height;

svg\_text(TEXT\_LEFT, top + TEXT\_BASELINE, to\_string(bin));

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT, "#800000", "#DC143C");

top += BIN\_HEIGHT;

}

svg\_end();

}

*Код файла test.cpp:*

#include "histogram.h"

#include <cassert>

void test\_positive() {

double min = 0;

double max = 0;

find\_minmax({1, 2, 3}, min, max);

assert(min == 1);

assert(max == 3);

}

void test\_negative(){

double min = 0;

double max = 0;

find\_minmax({-2, -1, -3}, min, max);

assert(min == -3);

assert(max == -1);

}

void test\_samenum(){

double min = 0;

double max = 0;

find\_minmax({3, 3, 3}, min, max);

assert(min == 3);

assert(max == 3);

}

void test\_onenum(){

double min = 0;

double max = 0;

find\_minmax({12}, min, max);

assert(min == 12);

assert(max == 12);

}

void test\_empty(){

double min = 0;

double max = 0;

find\_minmax({}, min, max);

assert(min == 0);

assert(max == 0);

}

int main() {

test\_positive();

test\_negative();

test\_samenum();

test\_onenum();

test\_empty();

}