Федеральное государственное бюджетное образовательное учреждение высшего образования

“Национальный исследовательский университет “МЭИ” ”

**ОТЧЕТ**

По лабораторной работе №3

“Д[екомпозиция и контроль корректности программ](http://uii.mpei.ru/study/courses/cs/lab03/index.html)”

ФИО студента Машкова Ирина Денисовна

Номер группы А-02-18

Имя преподавателя Мохов Андрей Сергеевич

Козлюк Дмитрий Александрович

**Цель работы**

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

**Задание**

Выделить в модуль все функции для работы с SVG и опубликовать коммит.

В ходе выполнения этой работы я следовала такому же алгоритму , как для выделения в модуль функции поиска минимума и максимума .

**Файл main.cpp**:

#include<iostream>

#include<vector>

#include "histogram.h"

#include "histogramSVG.h"

using namespace std;

vector<double> input\_numbers(size\_t number\_count)

{

vector<double> result(number\_count);

for (size\_t i=0; i<number\_count; i++)

{cin>>result[i];}

return result;

}

int main()

{

size\_t number\_count;

cerr<<" Enter number count\n";

cin>>number\_count;

cerr<<" Enter numbers\n";

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

size\_t bin\_count;

cerr<<" Enter bin count\n";

cin>>bin\_count;

const auto bins=make\_histogram(numbers, bin\_count);

show\_histogram\_svg(bins);

}

**Файл histogramSVG.h:**

#pragma once

#include <string>

#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, string fill);

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

**Файл histogramSVG.cpp:**

#include "histogramSVG.h"

#include<iostream>

#include<vector>

#include<string>

using namespace std;

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)

{

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;

string stroke = "red";

string fill = "blue";

const size\_t MAX=IMAGE\_WIDTH-TEXT\_WIDTH;

size\_t max\_lenght=0;

for(size\_t bin : bins)

{

if(bin>max\_lenght)

{

max\_lenght=bin\*10;

}

}

const bool scaling\_needed=max\_lenght>MAX;

for(size\_t bin:bins)

{

size\_t height=bin;

if(scaling\_needed)

{

const double scaling\_factor=(double)MAX/max\_lenght;

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, stroke, fill);

top += BIN\_HEIGHT;

}

svg\_end();

}

**Ссылка на репозитарий: https://github.com/MashkovaID/lab03.1**