Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ»

Институт информационных и вычислительных технологий

Кафедра Управления и интеллектуальных технологий

**Отчёт по лабораторной работе № 3**

# По курсу «Разработка ПО систем управления»

# «Декомпозиция программы»

# Выполнил студент группы А-03-19

# Шепелин В.А

# Проверили

# Мохов А. С

# Козлюк Д. А

Москва 2020

**Задание.**

# 1) Написать программу для построения гистограммы массива чисел как изображения в формате SVG

2) Доработать программу в соответствии с вариантом.

#### Вариант 3

Измените высоту изображения IMAGE\_HEIGHT = 700. Дайте пользователю возможность задавать высоту столбца гистограммы. Если итоговая высота гистограммы больше IMAGE\_HEIGHT, рассчитывать высоту столбца как (IMAGE\_HEIGHT / bins\_count).

**Пояснение к программе.**

Изменил константу const auto IMAGE\_HEIGHT = 700;. Далее добавил новую переменную, с помощью которой пользователь может менять высоту. Потом добавил If и если пользователь вводил height больше 700, то высота высчитывалась по формуле.

**Текст программы.**

1. Main.cpp

#include <iostream>

#include <vector>

#include "histogram.h"

#include "svg.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(const vector<double>& numbers,size\_t bin\_count)

{

double min;

double max;

find\_minmax(numbers,min,max);

vector<size\_t> bins(bin\_count,0);

for (double number : numbers)

{

size\_t bin;

bin = (number - min) / (max - min) \* bin\_count;

if (bin == bin\_count)

{

bin--;

}

bins[bin]++;

}

return bins;

}

int main()

{

const auto IMAGE\_HEIGHT = 700;

size\_t height;

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;

cerr << "enter height";

cin >> height;

double min, max;

find\_minmax(numbers, min, max);

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

if(height > IMAGE\_HEIGHT)

{

height = IMAGE\_HEIGHT / bin\_count;

}

show\_histogram\_svg(bins, height);

return 0;

}

2)histogram.cpp

#include "histogram.h"

#include <vector>

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

{

if(numbers.size()==0)

{

return;

}

else{

min = numbers[0];

max = numbers[0];

for (double number : numbers)

{

if (number < min)

{

min = number;

}

if (number > max)

{

max = number;

}

}

}

}

3)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

4) svg.cpp

#include <iostream>

#include <vector>

#include "svg.h"

using namespace std;

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

{

cout << "<rect x='"<<x<< "' y='" <<y<<"' width='" <<width <<"' height='" <<height <<"' stroke='"<<stroke<<"' fill='"<<fill<<"'/>";

}

void

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

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

}

void

svg\_begin(double width, double height) {

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

cout << "<svg ";

cout << "width='" << width << "' ";

cout << "height='" << height << "' ";

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

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

}

void

svg\_end() {

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

}

void

show\_histogram\_svg(const vector<size\_t>& bins, size\_t height)

{

const auto IMAGE\_WIDTH = 400;

const auto IMAGE\_HEIGHT = 700;

const auto TEXT\_LEFT = 20;

const auto TEXT\_BASELINE = 20;

const auto TEXT\_WIDTH = 50;

const auto BIN\_HEIGHT = 30;

const auto BLOCK\_WIDTH = 10;

svg\_begin(IMAGE\_WIDTH,height);

double top = 0;

for (size\_t bin : bins)

{

const double bin\_width = BLOCK\_WIDTH \* bin;

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

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT,"slateblue","#6A5ACD");

top += BIN\_HEIGHT;

}

}

5) svg.h

#ifndef SVG\_H\_INCLUDED

#define SVG\_H\_INCLUDED

#include <vector>

using namespace std;

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

void

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

void

svg\_begin(double width, double height);

void

svg\_end();

void

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

#endif // SVG\_H\_INCLUDED

**Ссылка на репозитарий.**

https://github.com/vvvvssss/lab03