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»**

**«Основы языка С++»**

Выполнил:

студент группы А-02-19

Долгачева А.С.

Проверили:

Мохов А. С

Козлюк Д. А

Москва 2020

**Задание**

Результат выполнения нужно оформить отдельным коммитом на GitHub. Доработку следует делать с использованием функции, для который нужно добавить unit-тесты из не менее двух существенно отличающихся случаев.

#### Вариант 4

Подсчитайте процент элементов, попавших в столбец, как целое двузначное число с % в конце и отображайте этот процент после столбца гистограммы с выравниванием:

3|\*\*\* 25%

5|\*\*\*\*\* 42%

4|\*\*\*\* 33%

**Описание логики решения задачи:**

Чтобы посчитать проценты, я взяла длину прямоугольника в строке, разделила на общее количество элементов в гистограмме и умножила на 100. Для вывода процентов в SVG-файле необходимо было добавить в исходную программу еще одну функцию, для вывода процентов.

**Код программы:**

#include <iostream>

#include <vector>

#include <conio.h>

#include "histogram.h"

using namespace std;

vector<double>

input\_numbers (size\_t count)

{

vector <double> result(count);

for (int i=0; i<count; i++)

{

cin>>result[i];

}

return result;

}

vector<size\_t> make\_histogram(const vector <double> &numbers,double &max, double &min, size\_t &bin\_count)

{

vector<size\_t>bins(bin\_count,0); /\* массив из индексов корзин \*/

for(double x : numbers) /\* мы каждому x присваиваем последовательно каждый элемент массива "numbers" \*/

{

size\_t bin\_index=(size\_t)((x-min)\*bin\_count/(max-min));

if (bin\_index==bin\_count) /\* индекс корзины "bin\_index"\*/

{

bin\_index=bin\_index-1; /\* нумирация идет с 0\*/

}

bins[bin\_index]++;

}

return bins;

}

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

{

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

}

void svg\_proc(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= "black", string fill= "black")

{

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

}

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 &number\_count)

{

const size\_t MAX\_ASTERISK=30;

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;

const auto BLOCK\_WIDTH = 10;

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

double top = 0;

unsigned max\_count=0;

for (size\_t b : bins) /\* мы присваиваем значение количества элементов \*/

{

if(max\_count<b)

max\_count=b; /\* здесь мы ищем максимальное количество элементов в массиве\*/

}

for (size\_t bin : bins)

{

size\_t height=30;

if(max\_count>MAX\_ASTERISK) /\* Если количество будет больше 35, то уменьшаем масштаб\*/

height=MAX\_ASTERISK\*((static\_cast<double>(bin))/max\_count); /\* это для того,чтобы было дробное число , получаем, используя "static\_cast<double>"\*/

else

{

height=bin;

}

height = BLOCK\_WIDTH \* height;

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

svg\_rect(TEXT\_WIDTH, top, height, BIN\_HEIGHT, "red", "#aaffaa");

size\_t pr=(size\_t)(bin\*100/number\_count);

svg\_proc(370, top + TEXT\_BASELINE, to\_string(pr));

top += BIN\_HEIGHT;

}

svg\_end();

}

int main()

{

double left;

double baseline;

string text;

size\_t number\_count, bin\_count;

double max=0, min=0;

cerr<<"number\_count=";

cin>>number\_count;

cerr<<"Enter number count: "<<"\n";

const auto numbers = input\_numbers(number\_count);/\* массив из количества элементов "number\_count"\*/

cerr<<"bin\_count=";

cin>> bin\_count;

find\_minmax (numbers,max,min);

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

show\_histogram\_svg(bins,number\_count);

return 0;

}

**Ссылка на репозитарий:**

**https://github.com/DolgachevaAS/cs-lab03.git**