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 № 4**

# По курсу «Разработка ПО систем управления»

# «Библиотеки и низкоуровневые операции»

# Выполнил студент группы:

# А-02-19

# Ковалев А.В.

# Проверили:

# Мохов А. С

# Козлюк Д. А

Москва 2020

# Задание 4.2 :

Код, который будет написан в этой части работы, должен в итоге формировать строку, которую нужно отобразить снизу итогового SVG в формате:

Windows v5.1 (build 1234)

Computer name: My-Comp

#### **Вариант 9**

С помощью функции curl\_version\_info() печатайте на стандартный вывод ошибок версию cURL и версию OpenSSL в строковом виде.

**Описание логики решения индивидуального задания:**

Нужно вызвать указатель на структуру где содержатся cURL и OpenSSL.Понадобится функцией curl\_version\_info(). Чтобы получить соответствующую версию нужно обратится к ней в структуре с помощью ( -> ).

**Код программы:**

***Main.cpp***

*#include <iostream>*

*#include <vector>*

*#include <conio.h>*

*#include<curl/curl.h>*

*#include<sstream>*

*#include <string>*

*#include "histogram.h"*

*#include "svg.h"*

*#include <cstdio>*

*#include <windows.h>*

*using namespace std;*

*vector<double> input\_numbers(istream& in, size\_t count)*

*{*

*vector <double> result(count);*

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

*{*

*in>>result[i];*

*}*

*return result;*

*}*

*Input*

*read\_input(istream& in, bool prompt )*

*{*

*Input Input;*

*size\_t number\_count;*

*if (prompt)*

*{*

*cerr << "Enter number count: ";*

*in >> number\_count;*

*cerr << "Enter numbers: ";*

*Input.numbers = input\_numbers(in, number\_count);*

*cerr << "Enter bin count: ";*

*in >> Input.bin\_count;*

*}*

*else*

*{*

*in >> number\_count;*

*Input.numbers = input\_numbers(in, number\_count);*

*in >> Input.bin\_count;*

*}*

*return Input;*

*}*

*size\_t*

*write\_data(void\* items, size\_t item\_size, size\_t item\_count, void\* ctx)*

*{*

*const size\_t data\_size = item\_size \* item\_count;*

*const char\* new\_items = reinterpret\_cast<const char\*>(items);*

*stringstream\* buffer = reinterpret\_cast<stringstream\*>(ctx);*

*buffer->write(new\_items, data\_size);*

*return data\_size;*

*}*

*vector<size\_t>*

*make\_histogram( struct Input Input)*

*{*

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

*double min;*

*double max;*

*find\_minmax(Input.numbers, min, max);*

*for(double x: Input.numbers)*

*{*

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

*if (bin\_index==Input.bin\_count)*

*{*

*bin\_index--;*

*}*

*bins[bin\_index]++;*

*}*

*return bins;*

*}*

*Input*

*download(const string& address)*

*{*

*stringstream buffer;*

*curl\_global\_init(CURL\_GLOBAL\_ALL);*

*CURL \*curl = curl\_easy\_init();*

*if(curl)*

*{*

*CURLcode res;*

*curl\_easy\_setopt(curl, CURLOPT\_URL, address.c\_str());*

*curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, write\_data);*

*curl\_easy\_setopt(curl, CURLOPT\_WRITEDATA, &buffer);*

*res = curl\_easy\_perform(curl);*

*if (res != CURLE\_OK)*

*{*

*cout << curl\_easy\_strerror(res) << endl;*

*curl\_version\_info\_data \*asd=curl\_version\_info(CURLVERSION\_NOW);*

*cout <<asd->version<<"\n";*

*cout <<asd->ssl\_version<<"\n";*

*exit(1);*

*}*

*curl\_easy\_cleanup(curl);*

*}*

*return read\_input(buffer, false);*

*}*

*int main(int argc, char\* argv[])*

*{*

*Input input;*

*if (argc > 1)*

*{*

*input = download(argv[1]);*

*}*

*else {*

*input = read\_input(cin, true);*

*}*

*const auto bins = make\_histogram (input);*

*show\_histogram\_svg(bins,input.bin\_count);*

*return 0;*

*}*

***Histogram.cpp***

*#include<vector>*

*#include <windows.h>*

*#include <iostream>*

*#include <string>*

*#include <sstream>*

*#include"histogram.h"*

*using namespace std;*

*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;*

*}*

*}*

*}*

*}*

*string make\_info\_text()*

*{*

*stringstream buffer;*

*DWORD mask = 0x0000ffff;*

*DWORD mask\_major = 0b00000000'00000000'00000000'11111111;*

*DWORD info = GetVersion();*

*DWORD platform = info >> 16;*

*DWORD version = info & mask;*

*DWORD version\_major = version & mask\_major;*

*DWORD version\_minor = version >> 8;*

*if ((info & 0x40000000) == 0)*

*{*

*DWORD build = platform;*

*buffer << "Windows v" << version\_major << "." << version\_minor << "(build " << build << ")\n";*

*}*

*char computer\_name[MAX\_COMPUTERNAME\_LENGTH + 1];*

*DWORD size = sizeof(computer\_name);*

*GetComputerNameA(computer\_name, &size);*

*buffer << "Computer name: " << computer\_name << "\n";*

*return buffer.str();*

*}*

***Histogram.h***

*#ifndef HISTOGRAM\_H\_INCLUDED*

*#define HISTOGRAM\_H\_INCLUDED*

*#include <iostream>*

*#include<vector>*

*#include<string>*

*#include <sstream>*

*using namespace std;*

*struct Input*

*{*

*vector<double> numbers;*

*size\_t bin\_count;*

*size\_t number\_count;*

*};*

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

*string make\_info\_text();*

*#endif // HISTOGRAM\_H\_INCLUDED*

***Svg.cpp***

*#include "svg.h"*

*#include"histogram.h"*

*#include <vector>*

*#include <string>*

*#include <iostream>*

*#include"block\_width.h"*

*using namespace std;*

*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 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 svg\_text1(double left, double baseline, string text)*

*{*

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

*}*

*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;*

*}*

*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, "black", "yellow");*

*top += BIN\_HEIGHT;*

*}*

*svg\_text1(TEXT\_LEFT, top+BIN\_HEIGHT, make\_info\_text());*

*svg\_end();*

*}*

***Svg.h***

*#ifndef SVG\_H\_INCLUDED*

*#define SVG\_H\_INCLUDED*

*#include <iostream>*

*#include<vector>*

*#include "string"*

*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, size\_t &number\_count);*

*void svg\_text1(double left, double baseline, string text);*

*#endif // SVG\_H\_INCLUDED*

***Block\_width.h***

*#include "block\_width.h"*

*#include <iostream>*

*using namespace std;*

*void block\_width(double &BLOCK\_WIDTH, int &flag)*

*{*

*while (flag==1)*

*{*

*cerr<<"Please enter the width of one block of the histogram\n";*

*cin>>BLOCK\_WIDTH;*

*if (BLOCK\_WIDTH<3 || BLOCK\_WIDTH>30)*

*cerr << "The width of one block of the histogram cannot be more then 30 px or less then 3 px\n";*

*if (BLOCK\_WIDTH>=3 && BLOCK\_WIDTH<=30)*

*flag=0;*

*}*

*}*

***Block\_width.cpp***

*#ifndef BLOCK\_WIDTH\_H\_INCLUDED*

*#define BLOCK\_WIDTH\_H\_INCLUDED*

*#include <vector>*

*using namespace std;*

*#include <iostream>*

*void block\_width(double &BLOCK\_WIDTH,int &flag);*

*#endif // BLOCK\_WIDTH\_H\_INCLUDED*