Лабораторная работа №3:

"Декомпозиция программы".

Дисциплина:"Разработка программного обеспечения систем управления"

Группа А-03-19

Работу выполнила Кубрак Екатерина Алексеевна

Дата 26.04.20.

## **Задание**

Программа для построения гистограммы из ЛР № 1 состоит из одной функции main() на более чем 100 строк, из-за чего в ней неудобно ориентироваться. Необходимо выделить части программы в функции:

* Ввод чисел:
  + принимает количество чисел, которое необходимо ввести;
  + возвращает вектор чисел.
* Поиск наибольшего и наименьшего значения:
  + принимает вектор чисел;
  + возвращает два результата — min и max.
* Расчет гистограммы:
  + принимает вектор чисел и количество корзин;
  + возвращает вектор количеств чисел в каждой корзине;
  + *вызывает* в процессе работы функцию поиска min и max.

**1)Клонировала репозиторий, изменила адрес репозитория и отправила на GitHub.**

$ git clone <https://github.com/PlushBeaver/cs19-lab01.git>

Cloning into 'cs19-lab01'...

remote: Enumerating objects: 8, done.

remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8

Receiving objects: 100% (8/8), done.

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git remote set-url origin <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 8, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 8 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (8/8), 1.58 KiB | 1.58 MiB/s, done.

Total 8 (delta 0), reused 8 (delta 0), pack-reused 0

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

\* [new branch] master -> master

**2) Вынесла ввод чисел в функцию.**

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

{

vector<double>result(count);

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

{

cin >> result[i];

}

return result;

}

**3) Заменила код ввода чисел в главной функции на вызов функции input\_numbers.**

vector<double> numbers = input\_numbers(number\_count);

**4) Сделала numbers константой и заменила vector<double> в главной функции на auto.**

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

**5) Вынесла поиск min и max в функцию.**

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

{

min = numbers[0];

max = numbers[0];

for (double number : numbers)

{

if (number < min)

{

min = number;

}

if (number > max)

{

max = number;

}

}

}

**6) Заменила код поиска max и min в главной функции на:**

double min, max;

find\_minmax(numbers, min, max);

**7) Использовала константную ссылку для параметра numbers для функции find\_minmax.**

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

{

min = numbers[0];

max = numbers[0];

for (double number : numbers)

{

if (number < min)

{

min = number;

}

if (number > max)

{

max = number;

}

}

}

**8) Выделила расчет количества чисел в столбцах гистограммы в функцию make\_histogram().**

**Функция:**

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

{

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

for (double number : numbers)

{

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

if (bin == bin\_count)

{

bin--;

}

bins[bin]++;

}

return (bins);

}

**Вызов функции make\_histogram в главной функции:**

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

**9) Выделила отображение гистограммы в функцию show\_histogram\_text().**

**Функция:**

void show\_histogram\_text(vector<size\_t>bins)

{

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 > MAX\_ASTERISK;

for (size\_t bin : bins)

{

if (bin < 100)

{

cout << ' ';

}

if (bin < 10)

{

cout << ' ';

}

cout << bin << "|";

size\_t height = bin;

if (scaling\_needed)

{

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

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

}

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

{

cout << '\*';

}

cout << '\n';

}

}

**Вызов функции show\_histogram\_text в главной функции:**

show\_histogram\_text(bins);

**10) Добавила коммит, отправила изменения на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "build:код с функциями"

[master 63e86db] build:код с функциями

3 files changed, 69 insertions(+), 35 deletions(-)

create mode 100644 bin/Debug/lab03-histogram.exe

create mode 100644 obj/Debug/main.o

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 11, done.

Counting objects: 100% (11/11), done.

Delta compression using up to 8 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (9/9), 562.27 KiB | 5.74 MiB/s, done.

Total 9 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 1 local object.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

72dfc06..63e86db master -> master

**11) Добавила функции вывода заголовка и окончания SVG.**

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";

}

**12) Добавила функцию графического вывода гистограммы и заменила вызов show\_histogram\_text(bins) вызовом show\_histogram\_svg(bins).**

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

{

svg\_begin(400, 300);

svg\_end();

}

..........

int main

{

...........

show\_histogram\_svg(bins);

return 0;

}

**13) Создала файл изображения.**

C:\Users\kubra\OneDrive\Рабочий стол\lab03\cs19-lab01\bin\Debug> lab03-histogram.exe <marks.txt >marks.svg

Enter number count: Enter numbers: Enter column count:

**В браузере marks.svg открывается, как пустая страница.**

**При нажатии Ctrl+U появляется исходный код страницы:**

<?xml version='1.0' encoding='UTF-8'?>

<svg width='400' height='300' viewBox='0 0 400 300' xmlns='http://www.w3.org/2000/svg'>

</svg>

**14) Для вывода подписей к столбцам написала функцию вывода текста в SVG, которая принимает координату по горизонтали (left), координату нижнего края текста по вертикали (baseline) и сам текст.**

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

{

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

}

**15) Для проверки вывела высоту первого столбца гистограммы.**

**Функция show\_histogram\_svg:**

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

{

svg\_begin(400, 300);

svg\_text(20, 20, to\_string(bins[0]));

svg\_end();

}

**Теперь в браузере marks.svg открывается, как страница с цифрой 2, а исходный код страницы выглядит так:**

<?xml version='1.0' encoding='UTF-8'?>

<svg width='400' height='300' viewBox='0 0 400 300' xmlns='http://www.w3.org/2000/svg'>

<text x='20' y='20'>2</text></svg>

**16) Закоммитила изменения и отправила на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

warning: LF will be replaced by CRLF in bin/Debug/marks.txt.

The file will have its original line endings in your working directory

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:вывод элементов SVG"

[master e09abf9] code:вывод элементов SVG

7 files changed, 33 insertions(+), 7 deletions(-)

delete mode 100644 .gitignore

rewrite bin/Debug/lab03-histogram.exe (63%)

create mode 100644 bin/Debug/marks.svg

create mode 100644 bin/Debug/marks.txt

delete mode 100644 marks.txt

rewrite obj/Debug/main.o (94%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 19, done.

Counting objects: 100% (19/19), done.

Delta compression using up to 8 threads

Compressing objects: 100% (7/7), done.

Writing objects: 100% (11/11), 303.03 KiB | 3.06 MiB/s, done.

Total 11 (delta 3), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (3/3), completed with 2 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

63e86db..e09abf9 master -> master

**17) Написала функцию для вывода прямоугольника в SVG.**

**Функция:**

void svg\_rect(double x, double y, double width, double height)

{

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

}

**Вызов функции в функции show\_histogram\_svg:**

svg\_rect(50, 0, bins[0] \* 10, 30);

**В браузере marks.svg открывается, как страница с цифрой 2 и прямоугольником чёрного цвета, а исходный код выглядит так:**

<?xml version='1.0' encoding='UTF-8'?>

<svg width='400' height='300' viewBox='0 0 400 300' xmlns='http://www.w3.org/2000/svg'>

<text x='20' y='20'>2</text><rect x='50' y='0' width='20' height='30'/>'</svg>

**18) Закоммитила изменения и отправила на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:вывод прямоугольника"

[master 9e3c79b] code:вывод прямоугольника

4 files changed, 7 insertions(+), 1 deletion(-)

rewrite obj/Debug/main.o (80%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 19, done.

Counting objects: 100% (19/19), done.

Delta compression using up to 8 threads

Compressing objects: 100% (7/7), done.

Writing objects: 100% (10/10), 245.78 KiB | 2.82 MiB/s, done.

Total 10 (delta 4), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (4/4), completed with 3 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

e09abf9..9e3c79b master -> master

**19) Заменила «магические числа» константами и проверила работу программы.**

**Код (изменения в функции show\_histogram\_svg):**

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;

const auto BLOCK\_WIDTH = 10;

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

svg\_text(TEXT\_LEFT, TEXT\_BASELINE, to\_string(bins[0]));

svg\_rect(TEXT\_WIDTH, 0, bins[0] \* BLOCK\_WIDTH, BIN\_HEIGHT);

svg\_end();

}

**В браузере marks.svg не изменилось.**

**20) Закоммитила изменения.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:добавила константы"

[master fe391f6] code:добавила константы

3 files changed, 10 insertions(+), 3 deletions(-)

**21) Вывод гистограммы.**

**Код функции show\_histogram\_svg:**

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;

const auto BLOCK\_WIDTH = 10;

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

top += BIN\_HEIGHT;

}

svg\_end();

}

**В браузере marks.svg открывается, как страница с гистограммой в чёрном цвете, а исходный код страницы выглядит так:**

<?xml version='1.0' encoding='UTF-8'?>

<svg width='400' height='300' viewBox='0 0 400 300' xmlns='http://www.w3.org/2000/svg'>

<text x='20' y='20'>2</text><rect x='50' y='0' width='20' height='30'/>'<text x='20' y='50'>5</text><rect x='50' y='30' width='50' height='30'/>'<text x='20' y='80'>3</text><rect x='50' y='60' width='30' height='30'/>'</svg>

**22) Закоммитила изменения и отправила на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:вывод гистограммы"

[master aa281fa] code:вывод гистограммы

4 files changed, 9 insertions(+), 3 deletions(-)

rewrite obj/Debug/main.o (81%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 28, done.

Counting objects: 100% (28/28), done.

Delta compression using up to 8 threads

Compressing objects: 100% (13/13), done.

Writing objects: 100% (19/19), 382.52 KiB | 2.85 MiB/s, done.

Total 19 (delta 6), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (6/6), completed with 3 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

9e3c79b..aa281fa master -> master

**23) Изменила цвета вывода гистограммы и проверила работу программы.**

**Код (изменения в функции svg\_rect):**

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 << "'/>'";

}

**Код (изменения в функции show\_histogram\_svg):**

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;

const auto BLOCK\_WIDTH = 10;

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

double top = 0;

string stroke = "mediumorchid";

string fill = "mediumorchid";

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, stroke, fill);

top += BIN\_HEIGHT;

}

svg\_end();

}

**В браузере marks.svg открывается, как гистограмма лавандового цвета, а исходный код выглядит так:**

<?xml version='1.0' encoding='UTF-8'?>

<svg width='400' height='300' viewBox='0 0 400 300' xmlns='http://www.w3.org/2000/svg'>

<text x='20' y='20'>2</text><rect x='50' y='0' width='20' height='30' storke='mediumorchid' fill='mediumorchid'/>'<text x='20' y='50'>5</text><rect x='50' y='30' width='50' height='30' storke='mediumorchid' fill='mediumorchid'/>'<text x='20' y='80'>3</text><rect x='50' y='60' width='30' height='30' storke='mediumorchid' fill='mediumorchid'/>'</svg>

**24) Добавила значения параметров по умолчанию функции svg\_rect().**

**Код (изменения в функции svg\_rect):**

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

{

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

}

**Код (изменения в функции show\_histogram\_svg):**

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;

const auto BLOCK\_WIDTH = 10;

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

top += BIN\_HEIGHT;

}

svg\_end();

}

**В браузере marks.svg не изменился.**

**25) Закоммитила изменения и отправила на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:поменяла цвета гистограммы"

[master d688839] code:поменяла цвета гистограммы

4 files changed, 3 insertions(+), 3 deletions(-)

rewrite bin/Debug/lab03-histogram.exe (62%)

rewrite obj/Debug/main.o (90%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 19, done.

Counting objects: 100% (19/19), done.

Delta compression using up to 8 threads

Compressing objects: 100% (7/7), done.

Writing objects: 100% (10/10), 294.41 KiB | 3.20 MiB/s, done.

Total 10 (delta 4), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (4/4), completed with 3 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

aa281fa..d688839 master -> master

**26) Добавила заголовочный файл histogram.h и перенесла функцию find\_minmax() в него.**

**Код заголовочного файла:**

#ifndef HISTOGRAM\_H\_INCLUDED

#define HISTOGRAM\_H\_INCLUDED

#include <vector>

using namespace std;

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

{

min = numbers[0];

max = numbers[0];

for (double number : numbers)

{

if (number < min)

{

min = number;

}

if (number > max)

{

max = number;

}

}

}

#endif // HISTOGRAM\_H\_INCLUDED

**27) Добавила файл histogram.cpp и перенесла в него определение функции find\_minmax().**

**Код в 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 в histogram.cpp:**

#include "histogram.h"

#include <iostream>

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

{

min = numbers[0];

max = numbers[0];

for (double number : numbers)

{

if (number < min)

{

min = number;

}

if (number > max)

{

max = number;

}

}

}

**28) Закоммитила изменения и отправила на GitHub:**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "build:добавлены файлы histogram.h и histogram.cpp"

[master 9a7cbfe] build:добавлены файлы histogram.h и histogram.cpp

7 files changed, 36 insertions(+), 17 deletions(-)

create mode 100644 histogram.cpp

create mode 100644 histogram.h

create mode 100644 lab03-histogram.depend

create mode 100644 obj/Debug/histogram.o

rewrite obj/Debug/main.o (86%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 21, done.

Counting objects: 100% (21/21), done.

Delta compression using up to 8 threads

Compressing objects: 100% (11/11), done.

Writing objects: 100% (13/13), 295.68 KiB | 2.64 MiB/s, done.

Total 13 (delta 5), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (5/5), completed with 2 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

d688839..9a7cbfe master -> master

**29) Cоздала проект для модульных тестов, добавила histogram.h и histogram.cpp и добавила файл для тестов.**

**Файл 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\_1()

{

double min = 0;

double max = 0;

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

assert(min == -3);

assert(max == -1);

}

void test\_2()

{

double min = 0;

double max = 0;

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

assert(min == 2);

assert(max == 2);

}

void test\_3()

{

double min = 0;

double max = 0;

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

assert(min == 2);

assert(max == 2);

}

void test\_4()

{

double min = 0;

double max = 0;

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

assert(min == 0);

assert(max == 0);

}

int main()

{

test\_positive();

test\_1();

test\_2();

test\_3();

test\_4();

}

**30)Сделала коммиты и отправила изменения на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:добавила тесты в один файл"

[master 737748d] code:добавила тесты в один файл

6 files changed, 53 insertions(+), 10 deletions(-)

rewrite lab03-test/bin/Debug/lab03-test.exe (61%)

rewrite lab03-test/obj/Debug/test.o (63%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 25, done.

Counting objects: 100% (25/25), done.

Delta compression using up to 8 threads

Compressing objects: 100% (10/10), done.

Writing objects: 100% (13/13), 236.82 KiB | 3.33 MiB/s, done.

Total 13 (delta 6), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (6/6), completed with 6 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

f40351b..737748d master -> master

**31)Добавила масштабирование, закоммитила и отправила на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:добавила масштабирование"

[master cdb93d4] code:добавила масштабирование

5 files changed, 23 insertions(+), 1 deletion(-)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 21, done.

Counting objects: 100% (21/21), done.

Delta compression using up to 8 threads

Compressing objects: 100% (9/9), done.

Writing objects: 100% (11/11), 295.37 KiB | 3.79 MiB/s, done.

Total 11 (delta 6), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (6/6), completed with 6 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

737748d..cdb93d4 master -> master

**32)Создала модуль с функциями SVG.**

**Файл svg.h:**

#ifndef SVG\_H\_INCLUDED

#define SVG\_H\_INCLUDED

#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 = "mediumorchid", string fill = "mediumorchid");

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

#endif // SVG\_H\_INCLUDED

**Файл svg.cpp:**

#include <iostream>

#include "svg.h"

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 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;

const auto BLOCK\_WIDTH = 10;

const size\_t SCREEN\_WIDTH = 80;

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

string stroke = "mediumorchid";

string fill = "mediumorchid";

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

double top = 0;

size\_t max\_count = 0;

for (size\_t count : bins)

{

if (count > max\_count)

{

max\_count = count;

}

}

const bool scaling\_needed = max\_count > MAX\_ASTERISK;

double scaling\_factor = 1;

for (size\_t bin : bins)

{

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

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

if (scaling\_needed)

{

scaling\_factor = (double)MAX\_ASTERISK / max\_count;

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

}

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT);

top += BIN\_HEIGHT;

}

svg\_end();

}

**Файл main.cpp:**

#include "histogram.h"

#include <iostream>

#include <vector>

#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)

{

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

for (double number : numbers)

{

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

if (bin == bin\_count)

{

bin--;

}

bins[bin]++;

}

return (bins);

}

void show\_histogram\_text(vector<size\_t>bins)

{

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 > MAX\_ASTERISK;

double scaling\_factor = 1;

for (size\_t bin : bins)

{

if (bin < 100)

{

cout << ' ';

}

if (bin < 10)

{

cout << ' ';

}

cout << bin << "|";

size\_t height = bin;

if (scaling\_needed)

{

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

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

}

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

double min, max;

find\_minmax(numbers, min, max);

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

// Вывод данных

show\_histogram\_svg(bins);

return 0;

}

**33)Закоммитила изменения и отправила на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:выделила в модуль все функции SVG"

[master cfaa73e] code:выделила в модуль все функции SVG

7 files changed, 88 insertions(+), 64 deletions(-)

rewrite obj/Debug/main.o (96%)

create mode 100644 obj/Debug/svg.o

create mode 100644 svg.cpp

create mode 100644 svg.h

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 22, done.

Counting objects: 100% (22/22), done.

Delta compression using up to 8 threads

Compressing objects: 100% (11/11), done.

Writing objects: 100% (13/13), 312.25 KiB | 1.35 MiB/s, done.

Total 13 (delta 7), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (7/7), completed with 3 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

cdb93d4..cfaa73e master -> master

**Индивидуальное задание.**

**Задача.**

Вычислите среднюю высоту столбца. Если столбец выше, цвет столбца должен быть красным, если ниже или равен средней высоте, цвет зеленый.

**1)Создала функцию, которая ищет среднюю высоту столбца, вынесла её в отдельный файл реализации sr\_ar.cpp и в заголовочный файл svg.h.**

**Код sr\_ar.cpp.**

#include <iostream>

#include "svg.h"

size\_t sred\_dlinna(size\_t n\_count, size\_t b\_count)

{

size\_t sred = n\_count/b\_count;

return sred;

}

**Код файла svg.h.**

#ifndef SVG\_H\_INCLUDED

#define SVG\_H\_INCLUDED

#include <vector>

#include <string>

using namespace std;

size\_t sred\_dlinna(size\_t n\_count, size\_t b\_count);

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, const size\_t sred);

#endif // SVG\_H\_INCLUDED

**2)Добавила в функцию show\_histogram\_svg условие для цвета гистограммы.**

if(bin > sred)

{

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT, "red", "red");

}

else

{

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT, "green", "green");

}

**3)Добавила два теста в test.cpp.**

void test\_5()

{

size\_t number\_count = 7;

size\_t bin\_count = 4;

double sred = sred\_dlinna(number\_count, bin\_count);

assert(sred == 1);

}

void test\_6()

{

size\_t number\_count = 0;

size\_t bin\_count = 4;

double sred = sred\_dlinna(number\_count, bin\_count);

assert(sred == 0);

}

**4)Закоммитила изменения и отправила изменения на GitHub.**

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

warning: LF will be replaced by CRLF in lab03-histogram.cbp.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in lab03-test/lab03-test.cbp.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in lab03-histogram.layout.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in lab03-test/lab03-test.layout.

The file will have its original line endings in your working directory

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:индивидуальное задание"

[master 7dcfb2e] code:индивидуальное задание

19 files changed, 124 insertions(+), 24 deletions(-)

rewrite bin/Debug/marks.svg (60%)

create mode 100644 lab03-histogram.layout

create mode 100644 lab03-test/lab03-test.layout

create mode 100644 lab03-test/obj/Debug/sr\_ar.o

rewrite lab03-test/obj/Debug/test.o (87%)

create mode 100644 obj/Debug/sr\_ar.o

create mode 100644 sr\_ar.cpp

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add .

warning: LF will be replaced by CRLF in lab03-histogram.layout.

The file will have its original line endings in your working directory

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit --amend

[master e035c87] code:индивидуальное задание

Date: Mon Apr 27 20:12:16 2020 +0300

19 files changed, 125 insertions(+), 24 deletions(-)

rewrite bin/Debug/marks.svg (60%)

create mode 100644 lab03-histogram.layout

create mode 100644 lab03-test/lab03-test.layout

create mode 100644 lab03-test/obj/Debug/sr\_ar.o

rewrite lab03-test/obj/Debug/test.o (87%)

create mode 100644 obj/Debug/sr\_ar.o

create mode 100644 sr\_ar.cpp

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push --force

Enumerating objects: 54, done.

Counting objects: 100% (54/54), done.

Delta compression using up to 8 threads

Compressing objects: 100% (25/25), done.

Writing objects: 100% (30/30), 536.56 KiB | 3.46 MiB/s, done.

Total 30 (delta 14), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (14/14), completed with 13 local objects.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

+ 7dcfb2e...e035c87 master -> master (forced update)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add .

warning: LF will be replaced by CRLF in lab03-histogram.layout.

The file will have its original line endings in your working directory

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit --amend

[master 18baa39] code:индивидуальное задание

Date: Mon Apr 27 20:12:16 2020 +0300

19 files changed, 125 insertions(+), 24 deletions(-)

rewrite bin/Debug/marks.svg (60%)

create mode 100644 lab03-histogram.layout

create mode 100644 lab03-test/lab03-test.layout

create mode 100644 lab03-test/obj/Debug/sr\_ar.o

rewrite lab03-test/obj/Debug/test.o (87%)

create mode 100644 obj/Debug/sr\_ar.o

create mode 100644 sr\_ar.cpp

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push --force

Enumerating objects: 160, done.

Counting objects: 100% (160/160), done.

Delta compression using up to 8 threads

Compressing objects: 100% (120/120), done.

Writing objects: 100% (160/160), 3.92 MiB | 133.00 KiB/s, done.

Total 160 (delta 61), reused 5 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (61/61), done.

To <https://github.com/Ekaterina-Kubrak/cs-lab03.git>

+ 38f396f...18baa39 master -> master (forced update)

**Сделала корректировку масштабирования после выполнения индивидуального задания и отправила изменения на GitHub.**

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

{

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;

const size\_t SCREEN\_WIDTH = 80;

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

double top = 0;

size\_t max\_count = 0;

for (size\_t count : bins)

{

if (count > max\_count)

{

max\_count = count;

}

}

const bool scaling\_needed = max\_count \* BLOCK\_WIDTH > MAX\_ASTERISK;

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

for (size\_t bin : bins)

{

size\_t height = bin;

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

if (scaling\_needed)

{

const double scaling\_factor = (double)MAX\_ASTERISK / (max\_count \* BLOCK\_WIDTH);

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

}

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

if(bin > sred)

{

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT, "red", "red");

}

else

{

svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT, "green", "green");

}

top += BIN\_HEIGHT;

}

svg\_end();

}

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git add -A

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git commit -m "code:корректировка масштабирования"

[master 180481e] code:корректировка масштабирования

9 files changed, 21 insertions(+), 20 deletions(-)

rewrite bin/Debug/marks.svg (70%)

kubra@DESKTOP-2QN1GL1 MINGW64 ~/OneDrive/Рабочий стол/lab03/cs19-lab01 (master)

$ git push

Enumerating objects: 29, done.

Counting objects: 100% (29/29), done.

Delta compression using up to 8 threads

Compressing objects: 100% (13/13), done.

Writing objects: 100% (15/15), 188.61 KiB | 2.90 MiB/s, done.

Total 15 (delta 9), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (9/9), completed with 9 local objects.

To https://github.com/Ekaterina-Kubrak/cs-lab03.git

d8296ad..180481e master -> master