|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Национальный исследовательский институт «МЭИ» |
| Лабораторная работа №3 |
| «Функции и структуры» |
|  |
|  |

|  |
| --- |
|  |

Студент: Никоноров ДС.

Группа: А-03-18

Преподаватели: Мохов А.С.,

Козлюк Д.А.

Москва 2019 НИУ «МЭИ»

**Вариант 13**

**Цель работы :**

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

**Основной код :**

|  |
| --- |
| #include <iostream> |
|  | #include <vector> |
|  | #include "laba3.h" |
|  | using namespace std; |
|  | //ôóíêöèÿ ââîäà |
|  | 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(vector<double> numbers, size\_t bin\_count) |
|  | { |
|  | double min, max; |
|  | find\_minmax(numbers, min, 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; |
|  |  |
|  | } |
|  | 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'; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | 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 = "blue", string fill = "blue") |
|  | { |
|  | 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; |
|  | for (size\_t bin : bins) { |
|  | const double bin\_width = 10 \* bin; |
|  | svg\_text(TEXT\_LEFT, top + TEXT\_BASELINE, to\_string(bin)); |
|  | svg\_rect(TEXT\_WIDTH, top, bin\_width, BIN\_HEIGHT,"#03E27F","#FD5ED4"); |
|  | top += BIN\_HEIGHT; |
|  | } |
|  | svg\_end(); |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | int main() { |
|  |  |
|  | 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; |
|  |  |
|  | const auto bins = make\_histogram(numbers, bin\_count); |
|  | show\_histogram\_svg(bins); |
|  |  |
|  | return 0; |
|  | } |

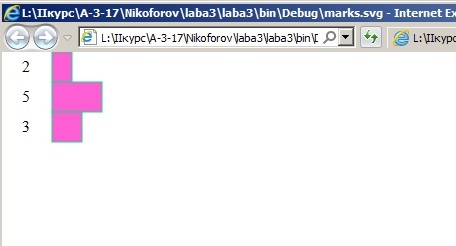
Код тестов :

|  |
| --- |
| #include "laba3.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\_otric() { |
|  | double min = 0; |
|  | double max = 0; |
|  | find\_minmax({-1, -2, -3}, min, max); |
|  | assert(min == -3); |
|  | assert(max == -1); |
|  | } |
|  |  |
|  |  |
|  | void |
|  | test\_odinak() { |
|  | double min = 0; |
|  | double max = 0; |
|  | find\_minmax({55, 55, 55}, min, max); |
|  | assert(min == 55); |
|  | assert(max == 55); |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  | void |
|  | test\_odin() { |
|  | double min = 0; |
|  | double max = 0; |
|  | find\_minmax({1337}, min, max); |
|  | assert(min == 1337); |
|  | assert(max == 1337); |
|  | } |
|  |  |
|  |  |
|  | void |
|  | test\_pusto() { |
|  | double min = 0; |
|  | double max = 0; |
|  | find\_minmax({}, min, max); |
|  | assert(min == ); |
|  | assert(max == ); |
|  | } |
|  | int |
|  | main() |
|  | { |
|  | test\_positive(); |
|  | test\_otric(); |
|  | test\_odinak(); |
|  | test\_odin(); |
|  | test\_pusto(); |
|  | } |

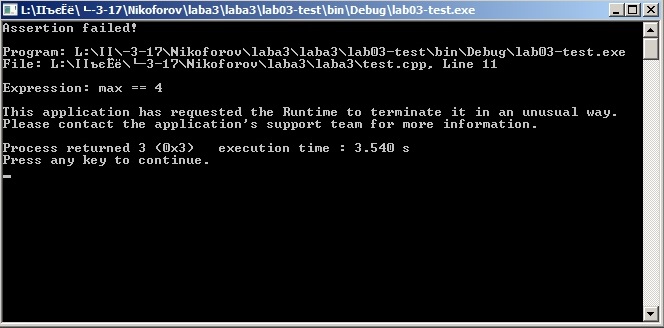
Заголовочный файл :

|  |
| --- |
| #ifndef LABA3\_H\_INCLUDED |
|  | #define LABA3\_H\_INCLUDED |
|  |  |
|  | #include <vector> |
|  | using namespace std; |
|  |  |
|  | void |
|  | find\_minmax(vector<double> numbers, double& min, double& max); |
|  |  |
|  |  |
|  | #endif // LABA3\_H\_INCLUDED |
|  |  |

*Вывод гистограммы*

**

*Вывод ошибки*

**