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 программного обеспечения систем управления»

Декомпозиция и контроль корректности программ

Выполнил студент группы А-02-20

Минаев Дмитрий Алексеевич

Проверил

Мохов А.С.

Козлюк Д.А.

Василькова П.Д.

Москва 2021

Цель работы

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

1. **Выделил все команды в функции.**

```
vector<double> input_numbers(size_t count)
{
   vector<double> result(count);
   cerr << "Enter numbers: ";
   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, 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;
}

int show_histogram_text(const vector<double> &numbers, size_t
bin_count, 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';
    }
    return max_count;
}

int shkala(const auto &max_name, int &int_shkal, int &j)
{
        int kof_shkal = max_name/int_shkal + 1;

            for (j=0; j<kof_shkal; j++)
            {
                cout<<"|";
                for (int z=0; z<int_shkal-1; z++)
                {
                    cout<<"-";
                }
            }

            cout<<"|"<<endl;
            cout<<"   ";
```

```
   int chislo_shkal=-int_shkal;

        for (j=0; j<kof_shkal*int_shkal; j++)
        {
           chislo_shkal = chislo_shkal + int_shkal;

           if (chislo_shkal <= int_shkal || chislo_shkal ==
kof_shkal*int_shkal) //Проверка для вывода 1ого, 2ого и последнего
числа под шкалой
           {
              cout<<chislo_shkal;
           }

           if (chislo_shkal < 10)
           {
              for (int z=0; z<int_shkal-1; z++)
              {
                 cout<<" ";
              }
           }
           if ((chislo_shkal>=10)&&(chislo_shkal<100))
           {
              for (int z=0; z<int_shkal; z++)
              {
                 cout<<" ";
              }
           }
           if (chislo_shkal>=100)
           {
              for (int z=0; z<int_shkal+1; z++)
              {
                 cout<<" ";
```

```
            }
        }
    }
}
```

2. **Создал модульный тест (для функции find_minmax()).**

histogram.h:

```
#ifndef HISTOGRAM_H_INCLUDED
#define HISTOGRAM_H_INCLUDED
#include <vector>

using namespace std;

void find_minmax(vector<double> numbers, double& min, double& max);

#endif // HISTOGRAM_H_INCLUDED
```

histogram.cpp:

```
#include "histogram.h"

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;
        }
    }
}
```

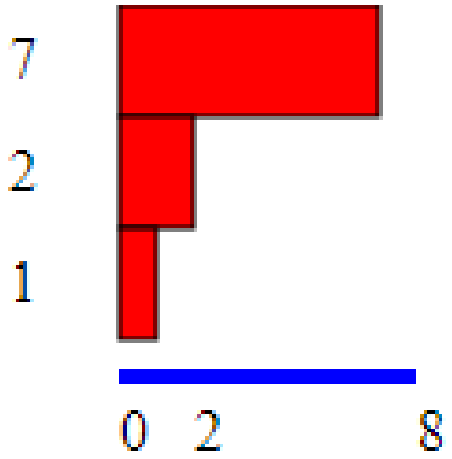
test.cpp:

```
#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_negativ() {
    double min = 0;
    double max = 0;
    find_minmax({-1, -2, -3}, min, max);
    assert(min == -3);
    assert(max == -1);
}

void
test_odinakov() {
    double min = 0;
    double max = 0;
    find_minmax({1, 1, 1}, min, max);
    assert(min == 1);
    assert(max == 1);
```

```
}

void
test_one() {
    double min = 0;
    double max = 0;
    find_minmax({1}, min, max);
    assert(min == 1);
    assert(max == 1);
}

int main()
{
        test_positive();
        test_negativ();
        test_odinakov();
        test_one();
}
```

3. **Пример SVG.**



Оформление данного SVG:

```
<?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'>7</text><rect x='50' y='0' width='70'
height='30' stroke='black' fill='red' /><text x='20'
y='50'>2</text><rect x='50' y='30' width='20' height='30'
stroke='black' fill='red' /><text x='20' y='80'>1</text><rect x='50'
y='60' width='10' height='30' stroke='black' fill='red' />
<line x1='50' y1='100' x2='130' y2='100' stroke='blue' stroke-
width='4'/>
<text x='50' y='120'>0</text><text x='70' y='120'>2</text><text
x='130' y='120'>8</text>
</svg>
```