

Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Факультет «О» Естественнонаучный

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №4**  
по дисциплине «Программирование на ЯВУ»  
на тему «STL»

Выполнил:  
Студент Костров Г. Ю.  
Группа О712Б  
Преподаватель: Васюков В.М.

Санкт-Петербург  
2022 г

## Постановка задачи:

Дан файл с английскими словами, разделёнными знаком '#'. Используя `std::getline` или `regex` составить множество (`std::set`) из уникальных слов в файле. Слова хранить в `std::string`. Также составить словарь (`std::map`) по следующему принципу: Ключ - число, значение - количество слов из указанного числа букв в файле.

Сгенерировать список из 100 структур со следующими полями:

Название - случайное слово из множества.

Основной параметр - используя `std::advance` получить значения трёх случайных элементов из словаря и перемножить их.

Вторичный параметр - квадрат значения из словаря, связанного с названием (в случае буквы брать первую букву названия).

Используя функции из библиотеки `algorithms` отсортировать список по возрастанию вторичного параметра.

Используя `std::copy_if`, создать новый список, содержащий только:

Элементы с чётной разностью основного и вторичного параметра (округлить разность до целого).

Сгенерировать на основе полученного отфильтрованного списка файл формата `json`.

Добавить возможность загрузки файла формата `json`, создания на основе данных в нём списка, перетасовки (`shuffle`) его и отображения полученного результата.

## Данные из файла:

generations#tyler#overall#committed#disorders#expires#georgia#parliamentary#sister#intel#frequencies#western#norway#spare#evening#mandatory#religious#subject#celtic#georgia#solid#tyler#humanities#original#indie#drug#toner#layout#western#holy#options#whore#occurs#healing#column#alerts#optical#atlanta#generations#reduction#possibilities#unable#options#simpsons#western#optical#equations#parenting#column#overall#quantities#toner#reduction#tournaments#agree#coleman#parliamentary#justin#penny#rental#position#holy#donald#ratio#shopping#sorts#dimension#seats#shoppercom#expires#parenting#lesbians#evening#alerts#beat#possibilities#wins#implement#alerts#subject#tyler#trained#touring#holy#optical#levitra#generations#integrating#wales#reviews#kits#hdtv#animated#column#operational#tyler#hash#donald#fundamentals#drug#parenting#norway#humanities#reviews#kilometers#penny#rental#quantities#ownership#wales#animated#mario#feeds#army#agree#rats#proteins#position#approx#trips#wales#feeds#dimension#parenting#drug#dimension#occurs#configuration#indie#evening#beach#trans#alphabetical#reduction#unable#implement#justin#donald#belong#touring#animated#generations#options#trips#left#justin#kilometers#integrating#equations#tournaments#implement#configuration#possibilities#drug#correlation#dimension#atlanta#perth#quantities#valentine#implement#norway#belong#shopping#planet#animated#wales#shopping#holy#ratio#original#ownership#rental#whore#justin#motels#spare#spare#integrating#reduction#touring#touring#subject#lesbians#intel#certified#feeds#quantities#evening#georgia#tournaments#lesbians#half#sister#alerts#equations#reviews#kilometers#indie#landing#perth#simpsons#layout#western#coleman#seats#left#position#proteins#sister#trained#whore#religious#unable#wins#shopping#trips#lesbians#perth#wins#indie#valentine#stakeholders#atlanta#seats#handjob#intel#hormone#norway#channels#sister#unable#verde#ownership#expire#intel

## Текст программы:

```
// Main.h
```

```
#include <iostream>
#include <fstream>
#include <set>
```

```

#include <map>
#include <string>
#include <list>
#include <time.h>
#include <algorithm>
#include "..\single_include\nlohmann\json.hpp"
#include <chrono>
#include <random>
#include "string_view"
#include "cstring"
#include "..\helpFunctions\functions.cpp"
#include <iomanip>

```

```
using namespace std;
```

```
struct element
```

```
{
    string name;
    int first;
    int second;
};
```

```
map<int,int> dictionary;
string dataFromFile;
set<string_view> uniqueElements;
list<element> List;
list<element> sortedList;
list<element> copiedList;
```

```
vector<nlohmann::json> JSON;
vector<element> masFromJson;
```

```
void createSetAndMap( string_view input_string, const char separator);
```

```
void generatorStructures();
```

```
void sortirovka();
```

```
list<element> copyIf();
```

```
void createJson();
```

```
void fromJson();
```

```
template <typename T>
void printInfo(T container);
```

## // Main.cpp

```
#include "main.h"
```

```
int main ()
{
    ifstream file("data.txt");
    getline(file,dataFromFile);
    file.close();
}
```

```

createSetAndMap(dataFromFile, '#');

generatorStructures();

sortirovka();

copiedList = copyIf();

createJson();

fromJson();

int variant = 0;
do {
    system("cls"); // очищаем экран

    string menu[] = {
        "What do you want to do?",
        "1. Create set",
        "2. Create map",
        "3. Create 100 structures",
        "4. Sort list of structures",
        "5. Copy list with new parameters",
        "6. Create new list in json",
        "7. Json file input"};
    printMenu(menu, 8);
    exitMenu(8);

    variant = getVariant(8);
    system("cls"); // очищаем экран
    switch (variant) {
        case 1: {
            // ВЫВОД МНОЖЕСТВА
            for(auto item : uniqueElements) {
                cout << item << endl;
            }
            break;
        }

        case 2: {
            // ВЫВОД СЛОВАРЯ
            for (auto i : dictionary) {
                cout << "Key " << i.first << ", value " << i.second << endl;
            }
            break;
        }

        case 3: {
            // ВЫВОД СТРУКТУР
            printInfo(List);

            break;
        }

        case 4: {
            printInfo(sortedList);
            break;
        }

        case 5: {
            printInfo(copiedList);
            break;
        }
    }
}

```

```

    }

    case 6: {
        for(auto i : JSON){cout << i <<endl;}
        break;
    }

    case 7: {
        printInfo(masFromJson);
        break;
    }

    }

    system("pause");
} while (variant != 8);

}

// Добавление элемента в множество
void createSetAndMap( string_view input_string, const char separator)
{
    size_t part_length = 0;
    while( ( part_length = input_string.find( separator ) ) != input_string.npos )
    {
        // set
        uniqueElements.insert(input_string.substr(0,part_length));

        // map
        if( dictionary.find( part_length ) == dictionary.end() ) {
            dictionary.insert( make_pair( part_length, 1 ) );
        } else {
            auto i = dictionary.find( part_length );
            i->second++;
        }

        input_string.remove_prefix( part_length + 1 );
    }

    if( !input_string.empty() ) {
        // set
        uniqueElements.insert(input_string.data());

        // map
        int lastWord = input_string.length();
        if( dictionary.find( lastWord ) == dictionary.end() ) {
            dictionary.insert( make_pair( lastWord, 1 ) );
        } else {
            auto i = dictionary.find( lastWord );
            i->second++;
        }
    }
}

int Random(int left, int right) {
    return rand()%right + left;
}

```

```

}
// Генератор структур
void generatorStructures()
{
    for(int j=0;j<100;j++)
    {
        element temp = {"", 1, NULL}; // элемента списка
        auto i = uniqueElements.begin();
        advance(i, Random(0,26));
        temp.name = *i; // название элемента

        // Основной параметр
        for ( int m = 0; m < 3; m++) {
            auto x = dictionary.begin();
            advance(x, Random(0, dictionary.size()));
            temp.first *= x->second;
        }

        // Вторичный параметр
        temp.second = dictionary[temp.name.length()] *
dictionary[temp.name.length()];

        List.push_back(temp);
    }
}

// Сортировка списка
void sortirovka()
{
    sortedList.assign(List.begin(), List.end());
    sortedList.sort([](const element &a, const element &b){
        return a.second < b.second;
    });
}

// создание копии списка
list<element> copyIf() {
    list<element> copiedList(List.size());

    copy_if(
        List.begin(),
        List.end(),
        copiedList.begin(),
        [](const element &a) {return ((a.first - a.second) % 2 == 0);}
    );
    copiedList.remove_if([](const element &a) {return a.first+a.second == 0;});
    return copiedList;
}

// запись в json
void createJson() {
    nlohmann::json j{};

    for(auto i : copiedList)
    {
        j["name"] = i.name;
        j["first"] = i.first;
        j["second"] = i.second;
        JSON.push_back(j);
    }
}

```

```

        ofstream file("new.json");
        file << JSON;
        file.close();
    }

    // открыть json
    void fromJson()
    {
        ifstream file;
        file.open("new.json");

        nlohmann::json g{};
        file >> g;
        for(auto i : g)
        {
            element temp;
            i.at("name").get_to(temp.name);
            i.at("first").get_to(temp.first);
            i.at("second").get_to(temp.second);
            masFromJson.push_back(temp);
        }

        file.close();

        shuffle(masFromJson.begin(), masFromJson.end(), default_random_engine(rand()));
    }

template <typename T>
void printInfo(T container) {
    cout << "|  N  |           Name           |  First  |  Second  |" << endl;
    cout << "-----" << endl;
    for(int N = 1; auto item : container) {
        cout << "|" << setw(4) << N++ << " | ";
        cout << setw(17) << item.name << " | ";
        cout << setw(7) << item.first << " | ";
        cout << setw(8) << item.second << " | " << endl;
    };
    cout << "-----" << endl;
}

```

## Результат работы программы:

Рисунок 1: Основное меню.

Рисунок 2: Создание множества элементов.

Рисунок 3: Создание словаря.

Рисунок 4: Создание структур.

Рисунок 5: Сортировка структур.

Рисунок 6: Копирование по признаку.

Рисунок 7: Создание JSON файла.

Рисунок 8: Чтение JSON файла.

```
What do you want to do?
1. Create set
2. Create map
3. Create 100 structures
4. Sort list of structures
5. Copy list with new parameters
6. Create new list in json
7. Json file input
-----
8. Exit

> █
```

Рисунок 1

```
seats
shoppercom
shopping
simpsons
sister
solid
sorts
spare
stakeholders
subject
toner
touring
tournaments
trained
trans
trips
tyler
unable
valentine
verde
wales
western
whore
wins
Для продолжения нажмите любую клавишу . . . █
```

Рисунок 2



```

Key 4, value 20
Key 5, value 48
Key 6, value 39
Key 7, value 44
Key 8, value 22
Key 9, value 30
Key 10, value 10
Key 11, value 13
Key 12, value 3
Key 13, value 7
Для продолжения нажмите любую клавишу . . . █

```

Рисунок 3

N	Name	First	Second
1	committed	5720	900
2	beach	198	2304
3	agree	19800	2304
4	certified	29952	900
5	army	5070	400
6	drug	37752	400
7	atlanta	18720	1936
8	celtic	3900	1521
9	animated	819	484
10	army	4620	400
11	feeds	82368	2304
12	disorders	50688	900
13	correlation	858	169
14	atlanta	1183	1936
15	coleman	1008	1936
16	dimension	23400	900
17	alphabetical	7800	9
18	expires	900	1936
19	approx	2800	1521
20	approx	6336	1521
21	feeds	21296	2304
22	celtic	2000	1521
23	drug	462	400

Рисунок 4

N	Name	First	Second
1	alphabetical	7800	9
2	alphabetical	180	9
3	alphabetical	4320	9
4	alphabetical	900	9
5	alphabetical	3510	9
6	alphabetical	351	9
7	configuration	56160	49
8	configuration	2002	49
9	correlation	858	169
10	correlation	19773	169
11	army	5070	400
12	drug	37752	400
13	army	4620	400
14	drug	462	400
15	army	14400	400
16	drug	8580	400
17	drug	147	400
18	army	1716	400
19	beat	490	400
20	drug	8580	400
21	beat	819	400
22	army	23400	400
23	beat	13552	400

Рисунок 5

N	Name	First	Second
1	committed	5720	900
2	beach	198	2304
3	agree	19800	2304
4	certified	29952	900
5	army	5070	400
6	drug	37752	400
7	atlanta	18720	1936
8	army	4620	400
9	feeds	82368	2304
10	disorders	50688	900
11	coleman	1008	1936
12	dimension	23400	900
13	expires	900	1936
14	feeds	21296	2304
15	drug	462	400
16	expires	13104	1936
17	evening	630	1936
18	atlanta	2600	1936
19	army	14400	400
20	evening	22308	1936
21	drug	8580	400
22	committed	4400	900
23	certified	8112	900

Рисунок 6

```

{"first":5720,"name":"committed","second":900}
{"first":198,"name":"beach","second":2304}
{"first":19800,"name":"agree","second":2304}
{"first":29952,"name":"certified","second":900}
{"first":5070,"name":"army","second":400}
{"first":37752,"name":"drug","second":400}
{"first":18720,"name":"atlanta","second":1936}
{"first":4620,"name":"army","second":400}
{"first":82368,"name":"feeds","second":2304}
{"first":50688,"name":"disorders","second":900}
{"first":1008,"name":"coleman","second":1936}
{"first":23400,"name":"dimension","second":900}
{"first":900,"name":"expires","second":1936}
{"first":21296,"name":"feeds","second":2304}
{"first":462,"name":"drug","second":400}
{"first":13104,"name":"expires","second":1936}
{"first":630,"name":"evening","second":1936}
{"first":2600,"name":"atlanta","second":1936}
{"first":14400,"name":"army","second":400}
{"first":22308,"name":"evening","second":1936}
{"first":8580,"name":"drug","second":400}
{"first":4400,"name":"committed","second":900}
{"first":8112,"name":"certified","second":900}
{"first":25168,"name":"coleman","second":1936}
{"first":1170,"name":"committed","second":900}

```

Рисунок 7

N	Name	First	Second
1	agree	6160	2304
2	agree	27456	2304
3	evening	630	1936
4	atlanta	10648	1936
5	evening	39600	1936
6	evening	1440	1936
7	equations	2904	900
8	feeds	82368	2304
9	committed	4320	900
10	beach	198	2304
11	certified	29952	900
12	donald	1911	1521
13	equations	462	900
14	certified	8112	900
15	beach	910	2304
16	coleman	3080	1936
17	celtic	4563	1521
18	expires	2640	1936
19	expires	3510	1936
20	drug	8580	400
21	drug	8580	400
22	alphabetical	351	9
23	committed	4400	900

Рисунок 8