

ЛАБОРАТОРНАЯ РАБОТА № 8
«ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ РАБОТЫ ПРОГРАММ
С ИСПОЛЬЗОВАНИЕМ КОНТЕЙНЕРОВ. СТАНДАРТНАЯ
БИБЛИОТЕКА ШАБЛОНОВ»

Цель работы: Изучить способы реализации и особенности работы программ с использованием контейнеров стандартной библиотеки шаблонов.

Вариант задания

Вариант 11

Написать программу «Англо-русский и русско-английский словарь». «База данных» словаря должна содержать синонимичные варианты перевода слов.

Программа должна обеспечивать выбор с помощью меню и выполнение следующих функций:

- загрузка «базы данных» словаря (из файла);
- выбор режима работы: “англо-русский” или “русско-английский”;
- вывод вариантов перевода заданного английского слова;
- вывод вариантов перевода заданного русского слова.

Базу данных словаря реализовать в виде двух контейнеров типа `multimap`.

2. Код программы на языке C++

```
#include <iostream>
#include <fstream>
#include <map>
#include <vector>
#include <regex>

using namespace std;
void Search(multimap<string, string> Slovar, string key) {
    auto iter = Slovar.find(key);
    string res = "Результат: ";
    for (; iter != Slovar.end(); iter++) {
        if (iter->first != key) { cout << res << endl; return; }
        else res += iter->second+", ";
    }
}

multimap<string, string> RussDB;
multimap<string, string> EnglisDB;

std::vector<std::string> split(const string& input, const string& regex);
void LoadDB();

int main()
{
    system("chcp 1251");
    int item = 1; bool re = false;
    bool flag = false;
    while (true) {
        system("cls");
        cout << "1 - Загрузить данные из файла" << endl;
        if (flag) {
            cout << "2 - Переключить режим перевода. Текущий режим : " << (re ? "
англо-русский" : " русско-английский") << endl;
            cout << "3 - Перевести слово согласно режиму" << endl;
        }
        cout << "0 - Выход" << endl;
        cout << "Номер пункта -->";
        cin >> item;
        switch (item)
        {
            case 1:
                LoadDB();
                flag = true;
                break;
            case 2:
                re = !re;
                break;
            case 3:
                while (true) {
                    system("cls");
                    string word;
                    cout << "Введите слово для перевода, 0 для выхода: "; cin >>
word;

                    if (word == "0") break;
                    re ? Search(EnglisDB, word) : Search(RussDB, word);
                    system("pause");
                }
                break;
            case 0:
                return 1;
                break;
            default:
                break;
        }
    }
}
```

```
}
```

```
void LoadDB() {
    ifstream fileE("English.txt"); ifstream fileR("Russian.txt");
    string line;
    while (getline(fileE, line)) {
        string key = ""; int KeySize = 0;
        for (; line[KeySize] != '-'; KeySize++) {
            key += line[KeySize];
        }
        string values = line.substr(KeySize + 1, line.size() - KeySize + 1);
        vector<string> valuess = split(values, "\\s*");
        for (auto& str : valuess) {
            EnglisDB.insert(pair<string, string>{key, str});
        }
    }
    while (getline(fileR, line)) {
        string key = ""; int KeySize = 0;
        for (; line[KeySize] != '-'; KeySize++) {
            key += line[KeySize];
        }
        string values = line.substr(KeySize + 1, line.size() - KeySize + 1);
        vector<string> valuess = split(values, "\\s*");
        for (auto& str : valuess) {
            RussDB.insert(pair<string, string>{key, str});
        }
    }
    fileE.close();
    fileR.close();
    cout << "Данные успешно загружены с базы данных" << endl;
    system("pause");
}
```

```
std::vector<std::string> split(const string& input, const string& regex) {
    std::regex re(regex);
    std::sregex_token_iterator
        first{ input.begin(), input.end(), re, -1 },
        last;
    return { first, last };
}
```

3. Тестирование и отладка

Для тестирования данной программы были созданы два файла текстового формата, содержащие в себе данные о слове в определенном формате, а именно <слово> - <значение 1>,<значение 2>.....

Два этих файла в дальнейшем будут использоваться для создания программой базы данных слов, с использованием контейнеров типа multimap. В данном случае, контейнер multimap создается по принципу <ключ> <значение>.

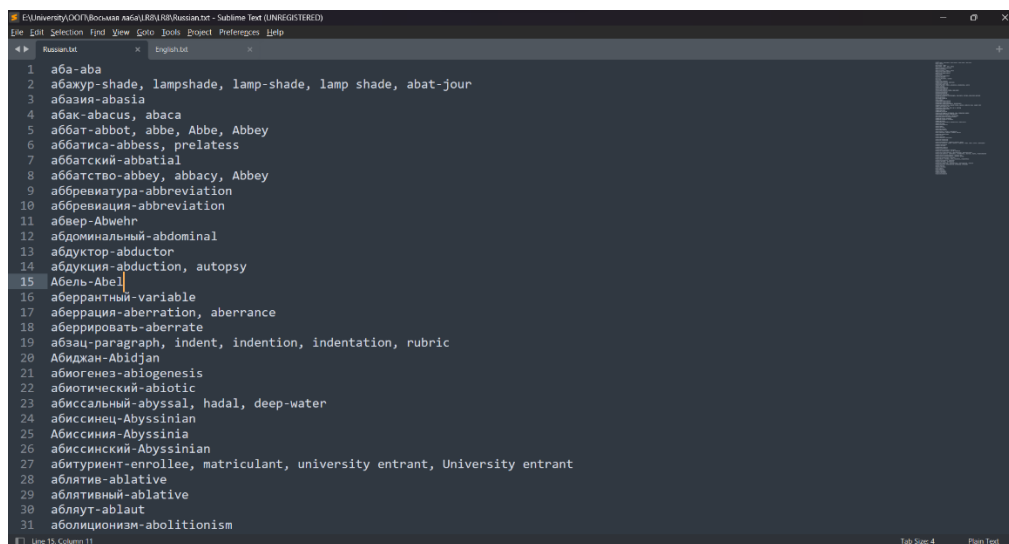


Рисунок 1 – Файл, использующийся для хранения базы данных русско-английского словаря

После запуска программы, у нас есть меню. В меню выбираем загрузку базы данных. После загрузки БД, нам становится доступен перевод в двух режимах, с русского на английский и наоборот.

```
1 - Загрузить данные из файла
0 - Выход
Номер пункта -->
```

Рисунок 2 – Главное меню программы

```
1 - Загрузить данные из файла
2 - Переключить режим перевода. Текущий режим : русско-английский
3 - Перевести слово согласно режиму
0 - Выход
Номер пункта -->
```

Рисунок 3 – Главное меню после загрузки данных.

Введите слово для перевода, 0 для выхода: аббатиса
Результат: abbess, prelatess,
Для продолжения нажмите любую клавишу . . .

Введите слово для перевода, 0 для выхода: аббревиатура
Результат: abbreviation,
Для продолжения нажмите любую клавишу . . .

Введите слово для перевода, 0 для выхода: абитуриент
Результат: enrollee, matriculant, university entrant, University entrant,
Для продолжения нажмите любую клавишу . . .

Введите слово для перевода, 0 для выхода: abatement
Результат: снижение, уменьшение, ослабление, скидка, смягчение, прекращение, аннулирование,
Для продолжения нажмите любую клавишу . . .

Введите слово для перевода, 0 для выхода: abbreviate
Результат: сокращать, использовать сокращения, урезать,
Для продолжения нажмите любую клавишу . . . |

Рисунок 4 – Работа программы (перевод слов в различных режимах)

Вывод

При выполнении данной лабораторной работы были получены навыки работы с контейнерами из стандартной шаблонной библиотеки STL. Также были повторно закреплены навыки работы с потоками и исключениями.