**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы информатики»

Отчет по лабораторной работе №10

# «Ввод, сортировка и двоичный поиск в массиве структур»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-11 |  | преподаватель каф. ИУ5 |
| Анцифров Никита |  | Аксёнова М.В. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

**Постановка задачи**

Англо-русский словарь построен в виде массива структур Dictionary и хранится в файле. Структура содержит английское слово и соответствующее ему русское слово. Максимальный размер словаря – 100 пар слов.

Разработать программу, которая:

* обеспечивает формирование словаря (добавление и удаление записей);
* записывает словарь, отсортированный по английским значениям слов, в файл;
* обеспечивает просмотр словаря;
* выполняет перевод слов с английского на русский, используя для поиска слова в словаре метод двоичного поиска в отсортированном массиве;
* выполняет перевод слов с русского на английский, используя для поиска слов в словаре метод перебора;

Программа должна обеспечивать диалог с помощью меню. Начальное число слов в словаре равно 10.

**Описание входных, выходных и вспомогательных данных**

const int **word\_length** – значение длины одного слова

char **answer** – символ ответа в меню

int **i**, **j** – счетчики циклов

char **eng\_word[]** – массив, отведенный под английское слово

char **rus\_word[]** – массив, отведенный под русское слово

ifstream **fin** – переменная для чтения файла

int **word\_amount** – переменная для количества слов

char\* **eng\_word** – ссылочная переменная для записи английского слова

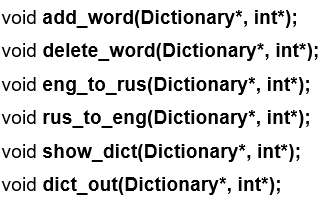
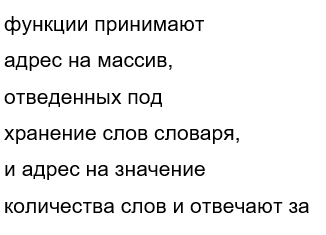
char\* **rus\_word** – ссылочная переменная для записи русского слова

char **temp\_arr** – массив под временное слово

ofstream **fout** – переменная для записи в файл

**Описание прототипов функций**

char **menu();** - функция отвечает за вывод меню выбора и возвращает символ ответа пользователя













struct **Dictionary** – структурная переменная, состоящая из char\* **eng** и char\* **rus**, где **eng** отвечает за английское слово в структуре, а **rus** – за русское

**Алгоритм**

Функция **main**





Функция **add\_word**



Функция **delete\_word**



Функция **eng\_to\_rus**



Функция **rus\_to\_eng**



Функция **show\_dict**



Функция **dict\_out**



**Текст программы**

*main.cpp*

#include <fstream>

#include <iostream>

#include <Windows.h>

#include "structure.h"

#include "functions\_proto.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int word\_length=31;

int i=0, j=0;

char eng\_word[word\_length], rus\_word[word\_length];

Dictionary d[100];

ifstream fin;

int word\_amount=0;

cout << "Убедитесь, что с в папке с проектом существует файл словаря с названием \"data.txt\"." << endl << "Пары слов в этом файле должны располагаться на отдельной строке." << endl << "Каждая строка должна иметь вид: \"\\*английское слово\\* \\*его перевод на русский язык\\*\"" << endl << endl;

fin.open("data.txt");

if (!fin.is\_open())

{

cout << "Ошибка: Файл \"data.txt\" не найден!" << endl;

}

else

{

cout << "Файл \"data.txt\" успешно открыт." << endl << endl;

while (!fin.eof())

{

fin >> eng\_word >> rus\_word;

d[i].eng=new char[strlen(eng\_word)+1];

d[i].rus=new char[strlen(rus\_word)+1];

strcpy\_s(d[i].eng, strlen(eng\_word)+1, eng\_word);

strcpy\_s(d[i].rus, strlen(rus\_word)+1, rus\_word);

word\_amount++;

i++;

}

for (i=0; i<(word\_amount+1); i++)

{

for (j=i+1; j<word\_amount; j++)

{

if (strcmp(d[i].eng, d[j].eng)>0)

{

swap(d[i].eng, d[j].eng);

swap(d[i].rus, d[j].rus);

}

}

}

do

{

switch(menu())

{

case '1':

add\_word(d, &word\_amount);

break;

case '2':

delete\_word(d, &word\_amount);

break;

case '3':

eng\_to\_rus(d, &word\_amount);

break;

case '4':

rus\_to\_eng(d, &word\_amount);

break;

case '5':

show\_dict(d, &word\_amount);

break;

case '6':

dict\_out(d, &word\_amount);

break;

case '7':

cout << endl << "Выход" << endl << endl;

return 0;

default:

cout << "Ошибка: Ваш символ не соответствует пункту меню! Повторите попытку" << endl << endl;

break;

}

}

while (1);

}

system("pause");

return 0;

}

*functions.cpp*

#include <fstream>

#include <iostream>

#include "structure.h"

#include "functions\_proto.h"

using namespace std;

char menu()

{

char answer;

setlocale(LC\_ALL, "Russian");

cout << "Меню: " << endl;

cout << "1. Добавить новое слово в словарь" << endl;

cout << "2. Удалить слово из словаря" << endl;

cout << "3. Перевести слова с английского на русский" << endl;

cout << "4. Перевести слова с русского на английский" << endl;

cout << "5. Показать словарь" << endl;

cout << "6. Вывести словарь в файл \"out.txt\"" << endl;

cout << "7. Выйти" << endl;

cout << "Выберете действие: ";

cin >> answer;

return answer;

}

void add\_word(Dictionary\* d, int\* word\_amount)

{

setlocale(LC\_ALL, "Russian");

char\* eng\_word;

char\* rus\_word;

char temp\_arr[31];

int i=0;

cout << endl << "Введите новое слово на английском языке: ";

cin >> temp\_arr;

eng\_word=new char[strlen(temp\_arr)+1];

strcpy\_s(eng\_word, strlen(temp\_arr)+1, temp\_arr);

cout << "Введите слово, переведенное на русский язык: ";

cin >> temp\_arr;

rus\_word=new char[strlen(temp\_arr) + 1];

strcpy\_s(rus\_word, strlen(temp\_arr)+1, temp\_arr);

if ((\*word\_amount)==0)

{

strcpy\_s(d[i].eng, strlen(eng\_word)+1, eng\_word);

strcpy\_s(d[i].rus, strlen(rus\_word)+1, rus\_word);

}

else

{

d[(\*word\_amount)].eng = new char[strlen(eng\_word)+1];

d[(\*word\_amount)].rus = new char[strlen(rus\_word)+1];

strcpy\_s(d[(\*word\_amount)].eng, strlen(eng\_word)+1, eng\_word);

strcpy\_s(d[(\*word\_amount)].rus, strlen(rus\_word)+1, rus\_word);

for (i=(\*word\_amount); i>0; i--)

{

if (strcmp(d[i].eng, d[i-1].eng)<0)

{

swap(d[i].eng, d[i-1].eng);

swap(d[i].rus, d[i-1].rus);

}

}

}

(\*word\_amount)++;

cout << "Слово успешно добавлено в словарь." << endl << endl;

}

void delete\_word(Dictionary\* d, int\* word\_amount)

{

setlocale(LC\_ALL, "Russian");

char temp\_arr[31];

int i=0;

if ((\*word\_amount)<=0)

{

cout << endl << "Ошибка: Словарь пуст!" << endl << endl;

}

else

{

cout << endl << "Введите слово, которое вы хотите удалить (на английском языке): ";

cin >> temp\_arr;

for (i=0; i<(\*word\_amount); i++)

{

if (strcmp(d[i].eng, temp\_arr)==0)

{

for (int j=i; j<((\*word\_amount)-1); j++)

{

d[j]=d[j+1];

}

cout << "Слово успешно удалено." << endl << endl;

(\*word\_amount)--;

break;

}

if (i==((\*word\_amount)-1))

{

cout << "Ошибка: Данного слова нет в словаре!" << endl << endl;

}

}

}

}

void eng\_to\_rus(Dictionary\* d, int\* word\_amount)

{

setlocale(LC\_ALL, "Russian");

char temp\_arr[31];

int i=0;

if ((\*word\_amount)<=0)

{

cout << endl << "Ошибка: Словарь пуст! " << endl << endl;

}

else

{

cout << endl << "Введите слово, которое вы хотите перевести с английского на русский: ";

cin >> temp\_arr;

for (i=0; i<(\*word\_amount); i++)

{

if (strcmp(d[i].eng, temp\_arr)==0)

{

cout << "Перевод этого слова на русский: " << d[i].rus << endl << endl;

break;

}

if (i==((\*word\_amount)-1))

{

cout << "Ошибка: Этого слова нет в словаре!" << endl << endl;

break;

}

}

}

}

void rus\_to\_eng(Dictionary\* d, int\* word\_amount)

{

setlocale(LC\_ALL, "Russian");

char temp\_arr[31];

int i=0;

if ((\*word\_amount)<=0)

{

cout << endl << "Ошибка: Словарь пуст! " << endl << endl;

}

else

{

cout << endl << "Введите слово, которое вы хотите перевести с русского на английский: ";

cin >> temp\_arr;

for (i=0; i<(\*word\_amount); i++)

{

if (strcmp(d[i].rus, temp\_arr)==0)

{

cout << "Перевод этого слова на английский: " << d[i].eng << endl << endl;

break;

}

if (i==((\*word\_amount)-1))

{

cout << "Ошибка: Этого слова нет в словаре!" << endl << endl;

}

}

}

}

void show\_dict(Dictionary\* d, int\* word\_amount)

{

int i=0;

cout << endl << "Словарь [ENG - RUS]: " << endl;

for (i=0; i<(\*word\_amount); i++)

{

cout << d[i].eng << " - " << d[i].rus << endl;

}

cout << endl;

}

void dict\_out(Dictionary\* d, int\* word\_amount)

{

ofstream fout;

int i=0;

fout.open("out.txt");

for (i=0; i<(\*word\_amount); i++)

{

fout << d[i].eng << " - " << d[i].rus << endl;

}

cout << endl << "Словарь успешно выведен в файл \"out.txt\"." << endl << endl;

}

*functions\_proto.h*

#include "structure.h"

char menu();

void add\_word(Dictionary\*, int\*);

void delete\_word(Dictionary\*, int\*);

void eng\_to\_rus(Dictionary\*, int\*);

void rus\_to\_eng(Dictionary\*, int\*);

void show\_dict(Dictionary\*, int\*);

void dict\_out(Dictionary\*, int\*);

*structure.h*

#pragma once

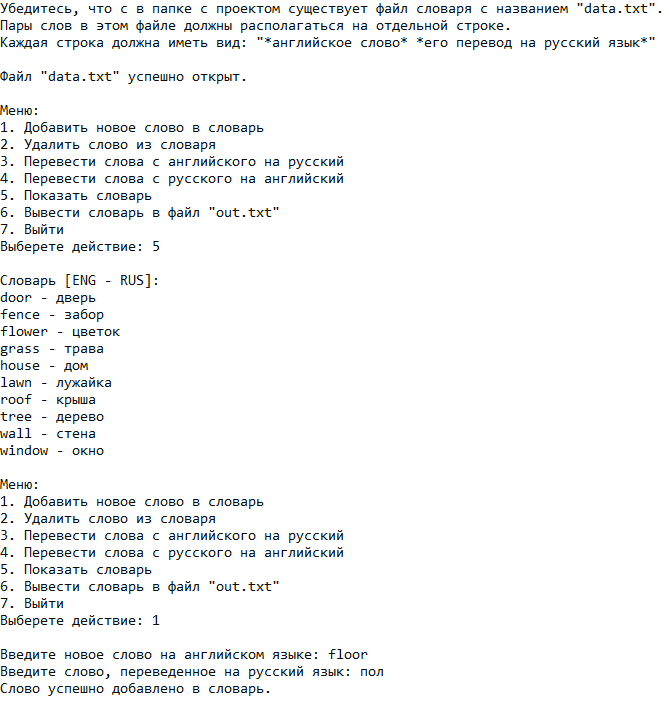
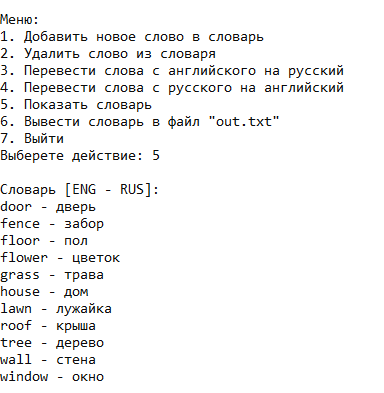
struct Dictionary {

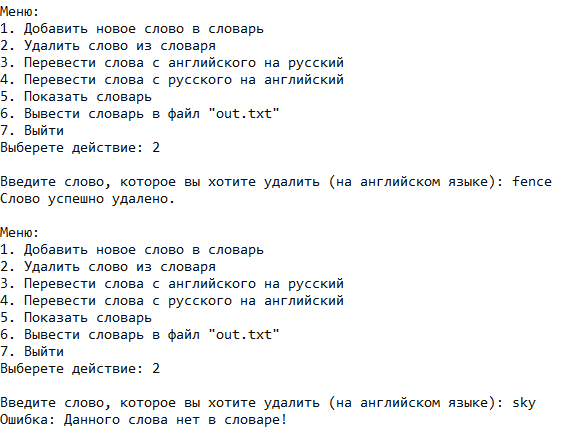
char\* eng;

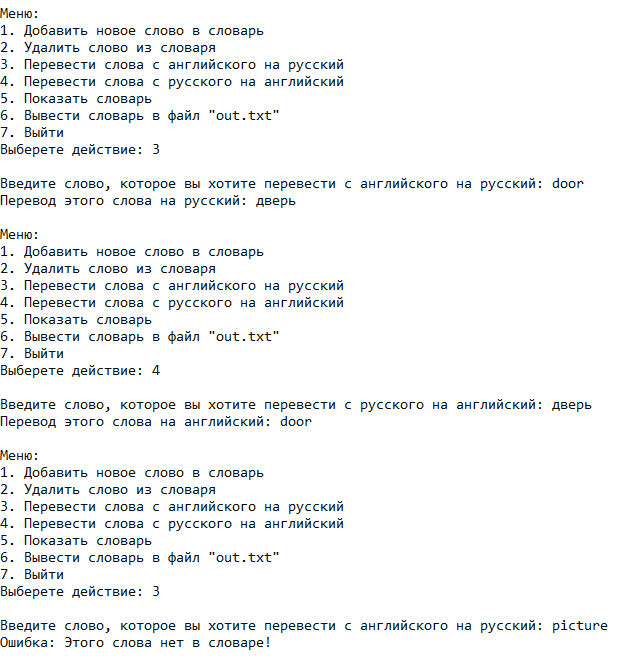
char\* rus;

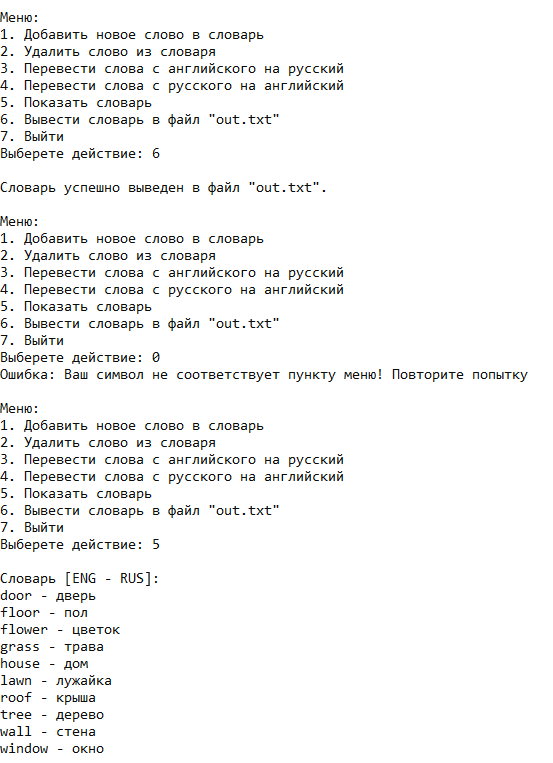
};

**Анализ результатов**









**Файлы**

data.txt out.txt

