

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №9
«Ввод, сортировка и двоичный поиск в массиве структур»

Выполнил:		Проверил:
Студент группы ИУ5-15Б		преподаватель каф. ИУ5
Расулов А. Н.		Аксенова М. В.
Подпись и дата:		Подпись и дата:

Москва, 2022 г.

Постановка задачи

Англо-русский словарь построен в виде массива структур Dictionary. 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.

Для исключения проблем, связанных с вводом кириллицы, вводите русские слова латинскими буквами, например: kot (cat), sobaka(dog) и т.п.

Разработка алгоритма

Входные данные:

dict.txt – файл с записанными в него парами слов, сначала идет английское слово, потом русское

Выходные данные:

dict.txt -- файл с записанными в него парами слов, которые хранились в памяти программы (сначала идет английское слово, потом русское)

Структура словаря:

```
struct Dictionary {  
    char* eng;  
    char* rus;  
};
```

Функции:

- int menu() -- функция для вывода меню действий пользователю и последующего ввода им команды для выполнения
 - int ans – переменная, хранящая значение пункта, который ввел пользователь
- void addWord(Dictionary* dict, int& n) – функция для добавления пары слов в словарь с размером n, на который указывает dict.

Внутри функции определены:

- char buff[BUFF_SIZE] – массив символов для хранения слова, введенного пользователем
 - int i – индекс массива для вставки туда новой пары слов
 - void deleteWord(Dictionary* dict, int& n) – функция для удаления пары слов из словаря с размером n, на который указывает dict.
- Внутри функции определены:
- char buff[BUFF_SIZE] – массив символов для хранения слова, введенного пользователем
 - void printDict(Dictionary* dict, int& n) – функция для вывода словаря с размером n, на который указывает dict, в консоль
 - void translateFromEnglish(Dictionary* dict, int& n) – функция для перевода слова с английского на русский. dict – указатель на словарь с длиной n.
 - char buff[BUFF_SIZE] – массив символов для хранения слова, введенного пользователем
 - int l, r, mid – соответственно левая, правая и середина границы поиска исходного слова в словаре (двоичный поиск в словаре)
 - int res – результат сравнения слова, введенного пользователем, и английского слова в словаре с индексом mid
 - void translateFromRussian(Dictionary* dict, int& n) – функция для перевода слова с русского на английский. Dict – указатель на словарь с длиной n.
 - char buff[BUFF_SIZE] – массив символов для хранения слова, введенного пользователем

- void writeToFile(Dictionary* dict, int& n) – функция для записи словаря в файл. Dict – указатель на словарь с длиной n, из которого будут брать пары слов для записи в файл
 - ofstream f(DICTIONARY_FILENAME) — потоковая переменная для записи словаря в файл
- void readDictFile(Dictionary* dict, int& n) – функция для чтения словаря из файла. Dict – указатель на словарь с длиной n, в который записаны пары слов из файла
 - ifstream f(DICTIONARY_FILENAME) — потоковая переменная для чтения файла с парами слов
 - char buff[BUFF_SIZE] — переменная для хранения текущего прочитанного слова из файла
- void dictCleanUp(Dictionary* dict, int& n) — функция для освобождения памяти, выделенной под словарь

main():

- Dictionary dict[MAX_SIZE]; -- словарь (массив пар слов)
- int n – число записей в словаре

Текст программы

dict.h

```
#include <cstring>
#include <fstream>
#include <iostream>
```

```
#define MAX_SIZE 100
#define BUFF_SIZE 31
#define DICTIONARY_FILENAME "dict.txt"
```

```
using namespace std;
```

```
struct Dictionary {
    char* eng;
    char* rus;
};
```

```
int menu();
void addWord(Dictionary* dict, int& n);
void deleteWord(Dictionary* dict, int& n);
void printDict(Dictionary* dict, int& n);
void translateFromEnglish(Dictionary* dict, int& n);
void translateFromRussian(Dictionary* dict, int& n);
void writeToFile(Dictionary* dict, int& n);
void readDictFile(Dictionary* dict, int& n);
void dictCleanUp(Dictionary* dict, int& n);
```

dict.cpp

```
#include "dict.h"
#include <cstring>
```

```
int menu() {
    cout << "Выберите действие:\n\
        1 -- добавить слово в словарь\n\
        2 -- удалить слово из словаря\n\
        3 -- перевод слов с английского на русский\n\
        4 -- перевод слов с русского на английский\n\
        5 -- просмотр словаря (вывод на экран словаря из ОП)\n\
        6 -- вывод словаря в файл\n\
        7 -- выход\n";

    int ans;
    cin >> ans;
    return ans;
}
```

```
void addWord(Dictionary* dict, int& n) {
    if (n == MAX_SIZE) {
        cout << "Ошибка! В словаре уже максимальное количество слов!\n";
        return;
    }
}
```

```

}

cout << "Введите через пробел слова, которые хотите добавить (сначала английское слово,
затем русское)\n";
char buff[BUFF_SIZE];
if (n == 0) {
    cin >> buff;
    dict[0].eng = new char[strlen(buff)];
    strcpy(dict[0].eng, buff);

    cin >> buff;
    dict[0].rus = new char[strlen(buff)];
    strcpy(dict[0].rus, buff);
    ++n;
    return;
}

cin >> buff;

// Находим место в массиве, куда мы закинем новую пару слов
int i = n;
while (i > 0 && strcmp(buff, dict[i - 1].eng) < 0) {
    --i;
}

// Сдвигаем слова после индекса i направо
for (int j = n; j >= i; --j) {
    dict[j] = dict[j - 1];
}

dict[i].eng = new char[strlen(buff)];
strcpy(dict[i].eng, buff);
cin >> buff;
dict[i].rus = new char[strlen(buff)];
strcpy(dict[i].rus, buff);
++n;
}

void deleteWord(Dictionary* dict, int& n) {
    cout << "Введите слово, которое хотите удалить\n";
    char buff[BUFF_SIZE];
    cin >> buff;

    for (int i = 0; i < n; ++i) {
        if (strcmp(dict[i].eng, buff) == 0 || strcmp(dict[i].rus, buff) == 0) {
            delete dict[i].eng;
            delete dict[i].rus;
            for (int j = i; j < n - 1; ++j) {
                dict[j] = dict[j + 1];
            }

            --n;
            cout << "Слово успешно удалено\n";
            return;
        }
    }
    cout << "Ошибка! Такого слова нет в словаре!\n";
}

void printDict(Dictionary* dict, int& n) {
    if (n == 0)
        cout << "Словарь пустой\n";
    else
        for (int i = 0; i < n; ++i)
            cout << dict[i].eng << ' ' << dict[i].rus << endl;
}

void translateFromEnglish(Dictionary* dict, int& n) {
    char buff[BUFF_SIZE];
    cout << "Введите слово, которое нужно перевести\n";

```

```

cin >> buff;

int l = 0, r = n - 1, mid, res;
while (l <= r) {
    mid = (l + r) / 2;

    res = strcmp(dict[mid].eng, buff);

    if (res == 0) {
        cout << "Перевод слова " << buff << ": " << dict[mid].rus << endl;
        return;
    }
    else if (res > 0) {
        r = mid - 1;
    }
    else {
        l = mid + 1;
    }
}

cout << "Ошибка! Слово не найдено!\n";
}

void translateFromRussian(Dictionary* dict, int& n) {
    char buff[BUFF_SIZE];
    cout << "Введите слово, которое нужно перевести\n";
    cin >> buff;

    for (int i = 0; i < n; ++i) {
        if (!strcmp(dict[i].rus, buff)) {
            cout << dict[i].eng << endl;
            return;
        }
    }
    cout << "Слово не найдено в словаре!\n";
}

void writeToFile(Dictionary* dict, int& n) {
    ofstream f(DICTIONARY_FILENAME);

    if (!f) {
        cerr << "Ошибка! невозможно открыть файл dict.txt для записи\n";
        return;
    }

    for (int i = 0; i < n; ++i) {
        f << dict[i].eng << ' ' << dict[i].rus << endl;
    }

    f.close();
}

void readDictFile(Dictionary* dict, int& n) {
    ifstream f(DICTIONARY_FILENAME);
    if (!f) {
        cerr << "Ошибка! невозможно открыть файл dict.txt для чтения\n";
        return;
    }

    n = 0;
    char buff[BUFF_SIZE];
    while (!f.eof()) {
        f >> buff;
        if (f.eof())
            break;
        dict[n].eng = new char[strlen(buff)]; // Выделяем столько памяти, сколько необходимо
        strcpy(dict[n].eng, buff);

        f >> buff;
        dict[n].rus = new char[strlen(buff)];
    }
}

```

```

        strcpy(dict[n].rus, buff);
        ++n;
    }
    f.close();
}

void dictCleanUp(Dictionary* dict, int& n) {
    for (int i = 0; i < n; ++i) {
        delete dict[i].eng;
        delete dict[i].rus;
    }
}

```

main.cpp

```
#include "dict.h"
```

```

int main() {
    Dictionary dict[MAX_SIZE];
    int n = 0; // Число записей в словаре

    readDictFile(dict, n);
    while (true) {
        switch ( menu() ) {
            case 1:
                addWord(dict, n);
                break;
            case 2:
                deleteWord(dict, n);
                break;
            case 3:
                translateFromEnglish(dict, n);
                break;
            case 4:
                translateFromRussian(dict, n);
                break;
            case 5:
                printDict(dict, n);
                break;
            case 6:
                writeToFile(dict, n);
                break;
            case 7:
                return 0;
            default:
                cout << "Нужно вводить число от 1 до 7\n";
                break;
        }
    }

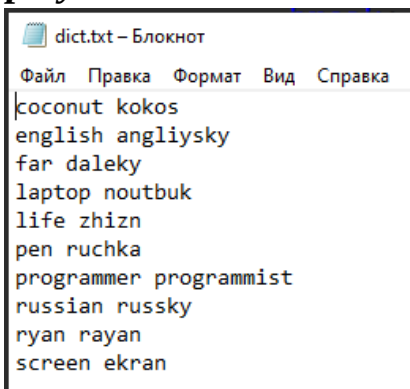
    dictCleanUp(dict, n); // Нужно не забывать освобождать выделенную память!

    return 0;
}

```

Анализ результатов

Файл со словарем до запуска программы:



Просмотр словаря:

```
Active code page: 65001
Выберите действие:
    1 -- добавить слово в словарь
    2 -- удалить слово из словаря
    3 -- перевод слов с английского на русский
    4 -- перевод слов с русского на английский
    5 -- просмотр словаря (вывод на экран словаря из ОП)
    6 -- вывод словаря в файл
    7 -- выход
5
coconut kokos
english angliysky
far daleky
laptop noutbuk
life zhizn
pen ruchka
programmer programmer
russian russky
ryan ryan
screen ekran
Выберите действие:
    1 -- добавить слово в словарь
    2 -- удалить слово из словаря
    3 -- перевод слов с английского на русский
    4 -- перевод слов с русского на английский
    5 -- просмотр словаря (вывод на экран словаря из ОП)
    6 -- вывод словаря в файл
    7 -- выход
_
```

Добавления слова в словарь

```
Выберите действие:
    1 -- добавить слово в словарь
    2 -- удалить слово из словаря
    3 -- перевод слов с английского на русский
    4 -- перевод слов с русского на английский
    5 -- просмотр словаря (вывод на экран словаря из ОП)
    6 -- вывод словаря в файл
    7 -- выход
1
Введите через пробел слова, которые хотите добавить (сначала английское слово, затем русское)
japan yaponiya
```

Удаления слова из словаря:

```
Выберите действие:
    1 -- добавить слово в словарь
    2 -- удалить слово из словаря
    3 -- перевод слов с английского на русский
    4 -- перевод слов с русского на английский
    5 -- просмотр словаря (вывод на экран словаря из ОП)
    6 -- вывод словаря в файл
    7 -- выход
2
Введите слово, которое хотите удалить
гуан
Слово успешно удалено
Выберите действие:
    1 -- добавить слово в словарь
    2 -- удалить слово из словаря
    3 -- перевод слов с английского на русский
    4 -- перевод слов с русского на английский
    5 -- просмотр словаря (вывод на экран словаря из ОП)
    6 -- вывод словаря в файл
    7 -- выход
5
coconut kokos
english angliysky
far daleky
japan yaponiya
laptop noutbuk
life zhizn
pen ruchka
programmer programmer
russian russky
screen ekran
_
```

Перевод слова с английского на русский

```
Выберите действие:
1 -- добавить слово в словарь
2 -- удалить слово из словаря
3 -- перевод слов с английского на русский
4 -- перевод слов с русского на английский
5 -- просмотр словаря (вывод на экран словаря из ОП)
6 -- вывод словаря в файл
7 -- выход
3
Введите слово, которое нужно перевести
life
Перевод слова life: zhizn
```

Перевод слова с русского на английский

```
Выберите действие:
1 -- добавить слово в словарь
2 -- удалить слово из словаря
3 -- перевод слов с английского на русский
4 -- перевод слов с русского на английский
5 -- просмотр словаря (вывод на экран словаря из ОП)
6 -- вывод словаря в файл
7 -- выход
4
Введите слово, которое нужно перевести
kokos
Перевод слова kokos: coconut
```

Запись словаря в файл:

```
Выберите действие:
1 -- добавить слово в словарь
2 -- удалить слово из словаря
3 -- перевод слов с английского на русский
4 -- перевод слов с русского на английский
5 -- просмотр словаря (вывод на экран словаря из ОП)
6 -- вывод словаря в файл
7 -- выход
6
```



dict.txt – Блокнот

Файл Правка Формат Вид

```
coconut kokos
english angliysky
far daleky
japan yaponiya
laptop noutbuk
life zhizn
pen ruchka
programmer programmist
russian russky
screen ekran
```