# 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.

Задания

1. В ходе самостоятельной подготовки изучить основы работы с контейнерами;
2. Разработать программу на языке C++ согласно заданному варианту (Рисунок 1);

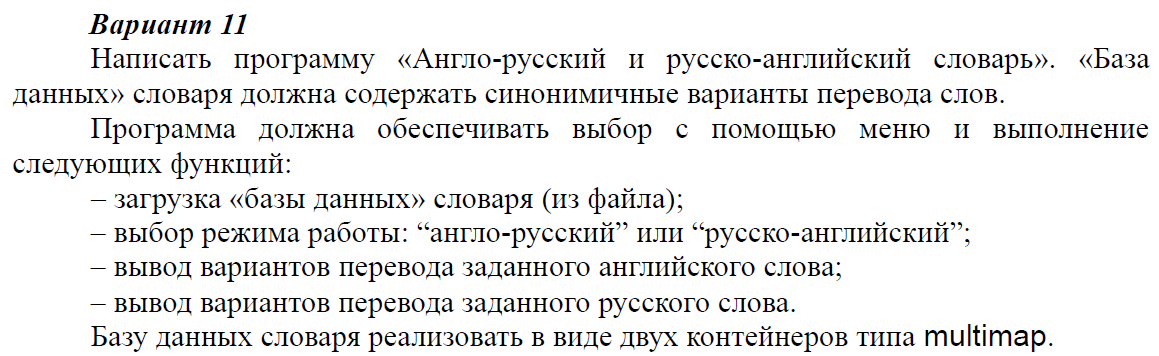


Рисунок 1 – Вариант задания

1. Разработать тестовые примеры и выполнить отладку программы;
2. Получить результаты работы программы и исследовать её свойства для различных режимов работы, сформулировать выводы;

Текст программы

#include <iostream>

#include <string>

#include <fstream>

#include <clocale>

#include <map>

#include <conio.h>

using namespace std;

// КОНСТАНТЫ

const int ENTER = 13;

const int ESQ = 27;

// Словарь (заполняется в зависимости от режима)

multimap<string, string> m;

// Прототипы функций

int printMapKey(string str);

int load(string fileName);

void menu1();

void menu2();

// Главная функция - вызов окон

int main()

{

system("chcp 1251");

while (true)

{

menu1(); // выбрать режим

menu2(); // перевод слов

}

return 0;

}

// Заполнение словаря

int load(string fileName)

{

ifstream fin(fileName, ios::in);

if (!fin)

{

cout << " Ошибка открытия файла \n\n";

system("pause");

return 0;

}

string key, word;

while (fin >> key) // считывание слов

{

fin >> word;

m.insert({ key, word });

}

fin.close();

return 1;

}

// Вывод всех полей по ключу

int printMapKey(string str)

{

multimap<string, string>::iterator iter = m.find(str);

if (iter == m.end())

{

cout << "\n Перевода не найдено \n";

return 0;

}

cout << "\n Варианты перевода: \n";

while ((iter != m.end()) && (iter->first == str))

{

cout << " " << iter->second << endl;

++iter;

}

return 1;

}

// Выбор режима словаря и заполнение контейнера

void menu1()

{

system("cls");

cout << "\n Выберите режим словаря: \n\n";

cout << " [0] - Русско-английский" << endl;

cout << " [1] - Англо-русский" << endl;

cout << " [Escape] - Окончание программы \n" << endl;

char c = \_getch();

while ((c != '0') && (c != '1') && (c != ESQ))

c = \_getch();

switch (c)

{

case '0':

load("RusEn.txt");

break;

case '1':

load("EnRus.txt");

break;

case ESQ:

cout << " Программа завершена \n";

exit(0);

break;

}

return;

}

// Перевод введённых слов

void menu2()

{

while (true)

{

system("cls");

cout << "\n Введите слово - ";

string str;

cin >> str;

printMapKey(str); // вывод синонимов

cout << endl;

cout << " [Enter] - Продолжить" << endl;

cout << " [Escape] - Выйти" << endl;

char c = \_getch();

while ((c != ESQ) && (c != ENTER))

c = \_getch();

if (c == ESQ) // выход в начальное окно

{

m.clear();

return;

}

}

}

Тестовые примеры

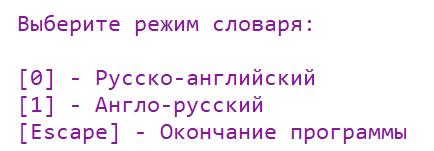


Рисунок 2 – Выбор режима переводчика

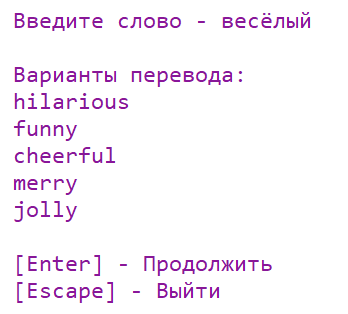


Рисунок 3 – Перевод с русского на английский

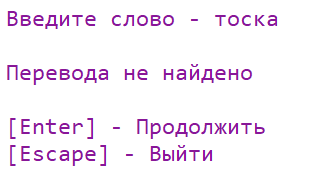


Рисунок 4 – Попытка перевести слово, отсутствующее в словаре

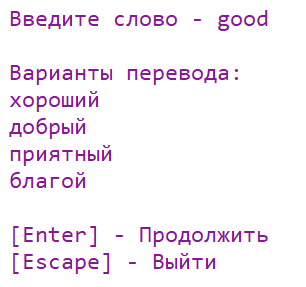


Рисунок 5 – Перевод с английского на русский

Вывод

В ходе работы были изучены контейнеры стандартной библиотеки шаблонов STL. Освоены операции и методы их обработки, механизм использования итераторов. В результате была написана программа, заполняющая словарь multimap значениями из файла и осуществляющая в нём поиск значений по ключам.