**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**

**(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Кафедра №301**

# Курсовая работа

по дисциплине

«**Программирование и основы алгоритмизации**»

на тему:

«Разработка программы, которая позволяет автоматизировать процесс тестирования»

Выполнили: студенты гр. 3О-102Б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Е. Головков

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.И. Лысов

Принял: доцент кафедры 301

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_C.C. Кананадзе

## Москва – 2015

СОДЕРЖАНИЕ

стр.

Текст задания 3

Описание операций 3

Текст программы 8

Иллюстрации, подтверждающие выполнение задания 9

**Текст задания**

Программа позволяет автоматизировать процесс тестирования. В окне программы изображается текст – последовательность вопросов, на которые испытуемый должен ответить путем выбора правильного ответа. Вопросы загружаются из файла на жестком диске. Имя файла теста передается программе при ее запуске. На ответы дается конечное время, которое устанавливает экзаменатор.

**Описание операций**

На съемном накопителе, приложенном к текстовому варианту курсовой работы содержится следующая корневая структура:

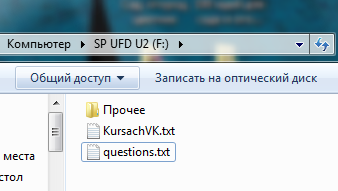


Рис. 1 - Изображение структуры съемного накопителя

Здесь лежат главные два файла, которым мы и уделим наше внимание. Это файлы под названиями «KursachVK.txt» и «questions.txt». Что же это за файлы, давайте опишем их роли:

1. «KursachVK.txt» - это файл, содержащий код программы
2. «questions.txt» - это файл, из которого программа будет загружать вопросы. В этот файл записываем вопросы для программы и числа соответствующие правильным ответам.

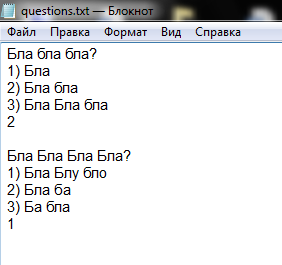


Рис. 2 - Содержание файла «questions.txt»

Для начала работы с программой, запишем в файл «questions.txt» свои вопросы и числа, соответствующие правильным ответам. Придумаем и запишем такие вопросы:

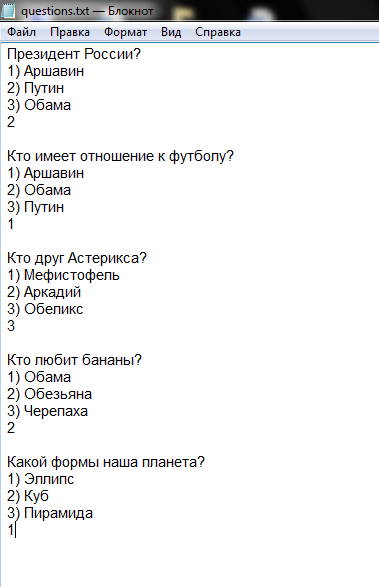


Рис. 3 - Текст, содержащийся в файле «questions.txt»

После того, как файл «questions.txt» заполнен, приступаем к вызову исполняемого файла «Ekzamenator.exe»

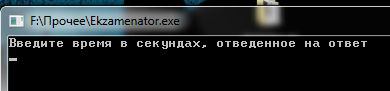
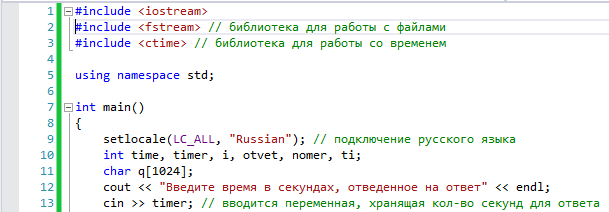


Рис. 4 - Программное уведомление о вводе времени

Этот процесс выполняет данная часть программы



После ввода времени, программа загружает первый вопрос из файла «questions.txt» и ожидает ввод ответа

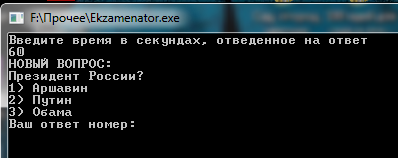
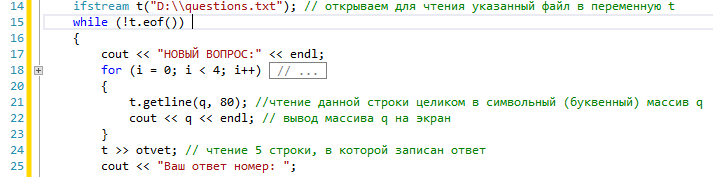


Рис. 5 - Вывод первого вопроса и ожидание ввода ответа

За эту процедуру отвечают следующие строчки кода



После ввода ответа, программа сравнивает введенный символ с тем, что записан в файле и выводит соответствующее решение.

Решений может быть три: 1) Ответ правильный 2) Ответ неправильный 3) Ответ правильный, но время истекло. Третий вариант имеет место быть, так как мы ввели переменную для хранения заданного времени

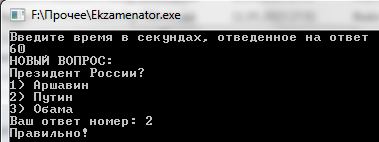


Рис. 6 - Ответ правильный

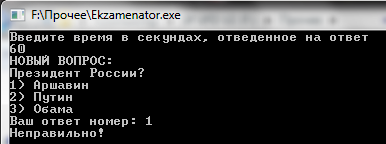


Рис. 7 - Ответ неправильный

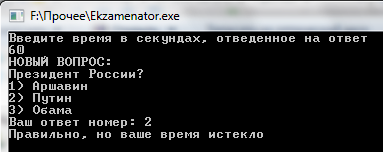
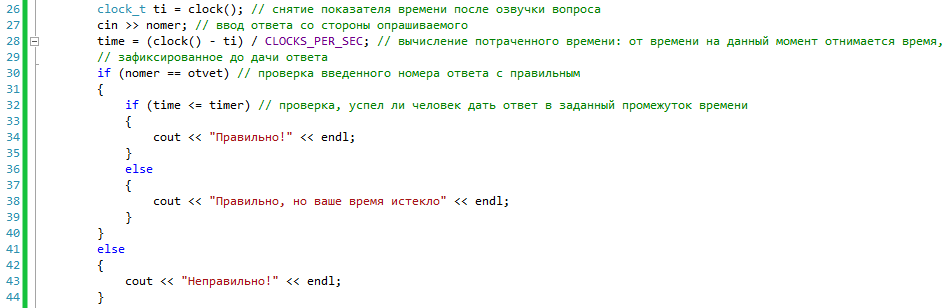


Рис. 8 - Ответ правильный, но время истекло

Эта часть программы выполняется в следующих строках кода



После ответа на вопрос, выводится следующий. Строки разделены между собой пробелами.

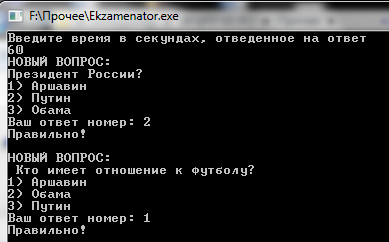
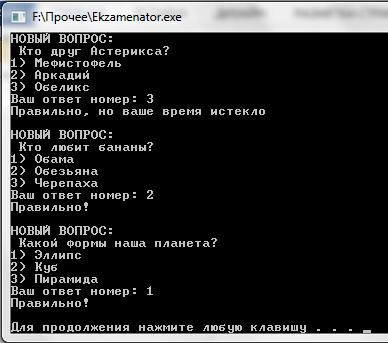


Рис. 9 - Вывод следующего вопроса

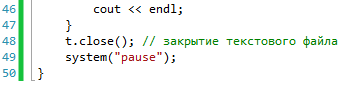
Отступ строки выполняется следующей строкой кода

C:\Users\Владимир\Desktop\Screenshot_13.png

После прочтения всех вопросов и получения на них ответов от пользователя, программа заканчивает работу, о чем свидетельствует последняя строка консоли



Завершение программы содержится в последних строках кода



Программа завершила выполнение своей задачи.

**Текст программы**

#include <iostream>

#include <fstream>

#include <ctime>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int time, timer, i, otvet, nomer, ti;

char q[1024];

cout << "Введите время в секундах, отведенное на ответ" << endl;

cin >> timer;

ifstream t("D:\\questions.txt");

while (!t.eof())

{

cout << "НОВЫЙ ВОПРОС:" << endl;

for (i = 0; i < 4; i++)

{

t.getline(q, 80);

cout << q << endl;

}

t >> otvet;

cout << "Ваш ответ номер: ";

clock\_t ti = clock();

cin >> nomer;

time = (clock() - ti) / CLOCKS\_PER\_SEC;

if (nomer == otvet)

{

if (time <= timer)

{

cout << "Правильно!" << endl;

}

else

{

cout << "Правильно, но ваше время истекло" << endl;

}

}

else

{

cout << "Неправильно!" << endl;

}

t >> q;

cout << endl;

}

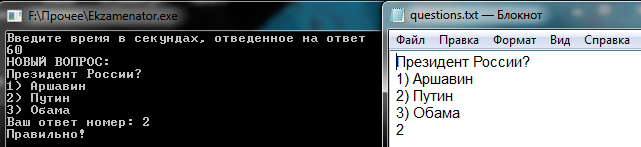
t.close();

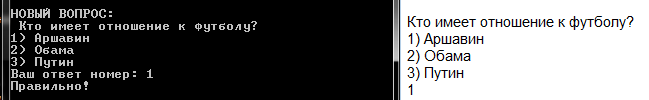
system("pause");

}

**Иллюстрации, подтверждающие работу программы**

1)



2)

3)

