МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ  
  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА(САМАРСКИЙ УНИВЕРСИТЕТ)  
  
Институт информатики и кибернетики  
Кафедра программных систем

Дисциплина  
**Объектно-ориентированное программирование**

**ОТЧЕТ**по лабораторной работе № 2  
 **Основы языка С++.  
Язык С. Обработка строк**  
Вариант № 12

Студент: Мананников М.А.  
Группа: 6214-020302D  
  
Преподаватель: Баландин А.В.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2022

**Текст задания**

Написать программу, которая создает в текстовом режиме файл в текущем  
каталоге с именем, вводимым с клавиатуры, и заполняет его произвольными  
строками, которые также вводятся с клавиатуры. После ввода последней строки файл закрывается. Далее программа должна распечатать все строки файла, заканчивающиеся на заданный символ, вводимый с клавиатуры. Строки выводятся в порядке их следования в исходном файле.

**Объектно-ориентированный анализ (ООА)**

**Информационная модель предметной области**

Информационный объект Текстовый файл:

* Свойства:
* Имя файла;
* Набор записей.
* Процедуры:
* Создание файла с записями;
* Формирование и отображение набора структур;
* Аннулирование файла.

**Объектно-ориентированное проектирование (ООD)**

* Свойства:
* Имя Файла – символьная строка;
* Набор записей – массив строк.
* Процедуры:
* Создание файла с записями – конструктор с параметром;
* Формирование и отображение набора структур - метод;
* Аннулирование файла - деструктор.

**Структура алгоритма программы**

1. Создание и заполнение файла;
2. Чтение файла, формирование и вывод набора структур;
3. Завершить работу.

**Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS // устранение предупреждения о нерекомендуемых функциях для старых менее безопасных функций

#include <iostream>

#pragma once // директива, позволяющая открыть исходный файл строго один раз

#include <string.h> // библиотека для работы со си-строками

#include <stdio.h> // стандартная библиотека ввода\вывода

#include <stdlib.h> // поиск, сортировка и тд

#include <Windows.h> // использовать функионал ОС

class TextFile // создание конструктора

{

public: char\* str = (char\*)malloc(1);// поле предложения

TextFile(char\* str1)// конструктор с параметром, создающий файл, который заполняется пользователем

{

int i = 0;

while (i < strlen(str1))// запись имени файла

{

str[i] = str1[i];

i++;

str = (char\*)realloc(str, i + 1);// увеличение динамической памяти

}

str[i] = '\0';

FILE\* ptrFile = fopen(str, "wt");// создание и открытие файла

printf("Введите предложение: \n");

char p1 = 1;// переменная для складывания в нее символа строки

i = 0;

char\* str2 = (char\*)malloc(1);// массив символов, введенные пользователем

int flag = 0;

while (flag != 2)// цикл заполнения массива

{

p1 = getchar(); // получаем символ

if (p1 == '\n') { // проверка на кол-во переходов на новую строку

flag++;

}

else flag = 0; // если флаг равен 0, то записываем символы

str2[i] = p1;

i++;

str2 = (char\*)realloc(str2, i + 1);

}

str2[i] = '\0';

i = 0;

while (str2[i] != NULL)// цикл для посимвольной записи предложения в файл

{

fputc((int)str2[i], ptrFile);// запись предложения в файл

i++;

}

fclose(ptrFile);// закрытие файла

}

public: ~TextFile()// деструктор

{

if (str != NULL)

{

free(str);// освобождение зарезервированной памяти

str = NULL;

}

}

};

void Set(char\* str, char s)// процедура вывода строк, заканчивающихся на заданный символ

{

FILE\* mfile = fopen(str, "rt");// открытие файла на чтение

char\* res = (char\*)malloc(1);

char ch;

int i = 0;

while ((ch = fgetc(mfile)) != EOF)// цикл для считывания предложения, fgetc - получение символа из файла

{

res[i] = ch;

i++;

res = (char\*)realloc(res, i + 1);

}

res[i] = '\0';

i = 0;

int k = 0;

while (i < strlen(res))// цикл вывода строк, оканчивающихся на заданный символ, в порядке их записи

{

char\* res1 = (char\*)malloc(1);

while (res[i] != '\n' && i < strlen(res))// запись одной строки в массив

{

if (k == 0 && res[i] == ' ')

{

i++;

}

else

{

res1[k] = res[i];

i++;

k++;

res1 = (char\*)realloc(res1, k + 1);

}

}

res1[k] = '\0';

i++;

if (res1[k - 1] == s)// проверка на последний символ строки

{

puts(res1);// вывод строки, если она подходит

}

free(res1);

k = 0;

}

printf("\n");

}

int main()

{

system("chcp 1251");

printf("Лабораторная работа 2 (12 вариант) подготовил студент группы 6103 Мананников Максим\n");

printf("Введите название файла: ");

char p = 1;

int i = 0;

char\* str1 = (char\*)malloc(40); // выделение памяти для строки

gets\_s(str1, 40); // процедура получения строки

strcat(str1, ".txt"); // объединение двух строк

TextFile mf = TextFile(str1);//создание объекта класса

printf("Введите символ: ");

char s = 1;

scanf("%c", &s); // запись символа, на который заканчивается строка, %c - тип символа

Set(mf.str, s); // вызов процедуры вывода строк, заканчивающихся на заданный символ

system("pause");

return 0;

}

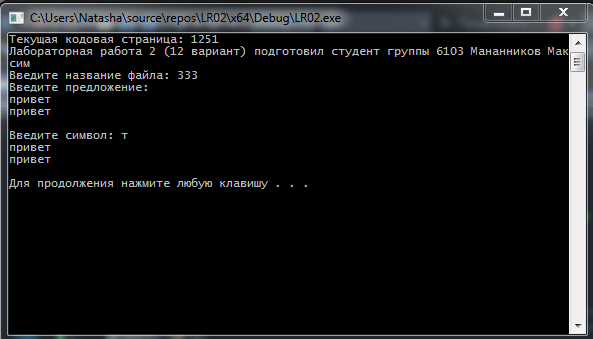


Рисунок 1 - Изображение окна работы программы

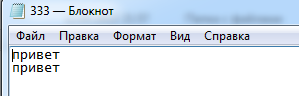


Рисунок 2 - Изображения окна текстового редактора с содержимым файлов