Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

**Кафедра информационных технологий**

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**(практика по получению первичных профессиональных умений и навыков)**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Василин Ю.В.

студент ФКТиПМ, 1 курса, 11 гр.,

направление 01.03.02 «Прикладная математика и информатика»

Руководитель учебной практики

(практика по получению

первичных профессиональных

умений и навыков) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лукащик Е.П.

к.ф.-м.н., доцент кафедры

информационных технологий

Краснодар 2018

* 1. ***Условие задачи***

Каждая запись вводного файла содержит фамилию и имя студента:

<Фамилия> <Имя>

Здесь <Фамилия> – строка, состоящая не более чем из 20 символов; <Имя> – строка, состоящая не более чем из 15 символов.

Требуется написать программу, которая формирует и печатает уникальный логин для каждого студента по следующему правилу: если фамилия встречается первый раз, то логин – это данная фамилия, если фамилия встречается второй раз, то логин – это фамилия, в конец которой приписывается число 2 и т.д. Полученные логины записать в выводной файл.

* 1. ***Описание алгоритма решения и структур хранения данных***

Для обработки файлов – считывания и записи данных – будем использовать библиотеку **<fstream>**.

Опишем структуру **list\_s**, содержащую поля surname типа string (фамилию студента/ов), **kolvo** типа **int** (переменная-счётчик, количество повторений данной фамилии в файле) и указатель **\*next** типа **list\_s** на следующий элемент.

Программа будет считывать фамилию с файла и заносить данные в структуру. Если фамилия ранее не встречалась: создаётся новый элемент структуры, фамилия записывается в выходной файл. Если фамилия уже есть в структуре: счётчик kolvo данного элемента структуры увеличивается на 1, затем в выходной файл записывается эта фамилия и kolvo в качестве строки.

Данные в исходный файл *students.txt*, если он не задан пользователем заранее, будут записываться при помощи функции ***input****,* аргумент которой – переменная n типа **int**, число студентов в будущем списке (также задаётся пользователем). Данные вводятся пользователем с клавиатуры.

Функция **\*list** типа **list\_s** осуществляет инициализацию элементов структуры list\_s.

Функция **working** типа **list\_s** составляет основную часть программы. В ней будет проходить запись уникальных фамилий в структуру данных и формирование логина путём сравнения с уже сохранёнными фамилиями и нумерации повторяющихся. Получившиеся данные будут записываться в файл *logins.txt*.

Функция **print** будет считывать данные с файла logins.txt и выводить их на экран.

Таким образом используются следующие переменные и типы:

1. *students.txt* содержит входные данные, заданные заранее или введённые через консоль
2. *logins.txt* содержит выходные данные
3. структура **lst** с полями:
   1. массив **surname** типа **string**, содержащий фамилию студента
   2. переменную **kolvo** типа **int**, считающую количество повторений данной фамилии в файле
   3. указатель **\*next** типа **list\_s** на следующий элемент
4. переменную **n** типа **int** – число студентов
5. логическую переменную **f** – ответ на вопрос: «Создать файл students.txt (1) или использовать готовый (0) ?»
6. массив **fio** типа **string**, использующийся для записи фамилий и имён студентов в файл
7. массив **surname** типа **string**, хранящий в себе обрабатываемую в данный момент строку с фамилией студента
   1. ***Описание входных и выходных данных***

Ввод данных будем осуществлять из текстового файла(students.txt), в котором построчно описаны фамилии и имена студентов.

Полученные в ходе работы программы логины записываются в файл logins.txt.

* 1. ***Текст программы***

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

struct list\_s

{

string surname;

int kolvo;

list\_s \*next;

};

list\_s \*list(string surname)

{

list\_s \*lst = new list\_s;

lst->surname = surname;

lst->next = NULL;

lst->kolvo = 0;

return lst;

}

list\_s \*working()

{

list\_s \*cur, \*head, \*cur2;

string surname;

bool f;

ifstream fin;

fin.open("students.txt");

ofstream fout;

fout.open("logins.txt");

if (fin.is\_open())

{

getline(fin, surname, ' ');

cur = list(surname);

head = cur;

fout << surname;

getline(fin, surname);

while (!fin.eof())

{

getline(fin, surname, ' ');

cur2 = head;

f = true;

while (cur2 != 0)

{

if (surname == cur2->surname)

{

(cur2->kolvo)++;

fout << endl << surname + to\_string(cur->kolvo);

f = false;

break;

}

else

{

cur2 = cur2->next;

}

}

if (f)

{

fout << endl << surname;

cur->next = list(surname);

}

getline(fin, surname);

}

fin.close();

fout.close();

return head;

}

else

{

cout << "Ошибка открытия файла" << endl;

return 0;

}

}

void print()

{

setlocale(LC\_ALL, "rus");

ifstream fin;

string login;

fin.open("logins.txt");

if (fin.is\_open())

{

cout << "Логины:" << endl << endl;

while (!fin.eof())

{

getline(fin, login);

cout << login << endl;

}

fin.close();

}

else cout << "Ошибка открытия файла" << endl;

}

void input(int n)

{

int i;

string fio;

ofstream fout;

fout.open("students.txt");

for (i = 1; i < n; i++)

{

cout << "Введите фамилию и имя студента " << i << endl;

cin >> fio;

fout << fio;

cin >> fio;

fout << " " << fio << endl;

}

cout << "Введите фамилию и имя студента " << i << endl;

cin >> fio;

fout << fio;

cin >> fio;

fout << " " << fio;

fout.close();

}

int main()

{

setlocale(LC\_ALL, "rus");

int n;

bool f;

cout << "Укажите число студентов" << endl;

cin >> n;

cout << "Создать файл students.txt (1) или использовать готовый(0) ?" << endl;

cin >> f;

if (f) input(n);

list\_s \*students = working();

list\_s \*cur = students;

print();

system("pause");

}

* 1. ***Тестовый запуск программы и вывод результатов работы.***

В файле students.txt содержатся фамилии и имена студентов:

Smith Liam

Jensen Olivia

Williams Benjamin

Jensen Adam

Brown Michael

Pritchard Fransis

Smith Owen

Miller Lily

Wilson Charlotte

Reed Isaac

Smith Zoe

Anderson Elizabeth

Anderson Jayden

Jackson Amelia

Reed Megan

Smith George

Thompson Carter

Sarif David

Программа обрабатывает.

Программа записывает логины в logins.txt.

Содержимое logins.txt:

Smith

Jensen

Williams

Jensen

Brown

Pritchard

Smith1

Miller

Wilson

Reed

Smith2

Anderson

Anderson2

Jackson

Reed

Smith3

Thompson

Sarif

Программа работает корректно.