Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Инженерно-технологическая академия ЮФУ

Институт компьютерных технологий и информационной безопасности

**Кафедра Математическое обеспечение и администрирование информационных систем Физико- математических основ и основ инженерного образования**

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

по курсу: «Алгоритмизация и программирование»

Выполнил

студент группы КТбо 1-1

№ КТ-02.03.03.01-о1 М.Р. Гончар

№ зачетки

Проверил

Ст. преподаватель кафедры ВТ Д.С. Кочубей

«27» декабря 2023

Таганрог, 2023

# Техническое задание

## Цель работы

Задание состоит в разработке программы, которая считывает настроечные параметры и формирует отчет по имеющимся данным в файлах (см. Таблица 1). Отчет записывается в текстовый файл. Число записей в каждом из описанных выше файлов произвольно.

Таблица 1 – Задание варианта 7

|  |  |  |
| --- | --- | --- |
| № варианта | Отчёт | Параметры |
| 7 | Сформировать список и подсчитать стоимость услуг, которыми пользовался указанный клиент в текущем квартале | ФИО клиента |

**Описание задания**

Имеется информация о клиентах телефонной компании и предоставляемых им услугах. Каждая услуга имеет собственный тариф, а каждый клиент может пользоваться произвольным набором услуг в течение ограниченного интервала времени (соответственно срокам договора). Имеются данные о фактическом использовании услуг.

Вся указанная информация представлена текстовыми файлами, структура которых выглядит следующим образом.

Файл информации о клиентах содержит фамилию, имя, отчество клиента, номер телефона, дату заключения договора, дату окончания договора, размер задолженности, допустимый кредит. Каждое поле отделяется запятой, запись – это строка текста. Пример содержимого в файле:

Иванов Иван Иванович, 9773672365, 12.10.2012, 12.10.2014,0,0

Петров Иван Васильевич, 9734672311, 22.01.2008, 01.11.2011,210,200

Васильев Илья Васильевич, 9714679805, 05.09.2010, 01.12.2012,0,2000

Файл информации об услугах состоит также из записей, состоящих из полей, разделенных запятыми. Каждая запись включает наименование услуги, ее код, тариф (в рублях), временной интервал измерения (мин., сутки, месяц. Если временной привязки нет, ставится символ #). Пример данных в файле:

Cвязь внутри сети, 1, 0.30, мин

СМС, 2, 0.15, #

Cвязь с другими мобильными сетями, 3, 0.50, мин

Роуминг, 4, 10, мин

Международный тариф, 5, 50, мин

Файл информации об услугах, оказанных клиентам, включает в себя записи, состоящие из полей номера телефона, кода услуги, даты и времени ее использования в секундах. Знак # используется в случае, если время не определяется (например, отправляется СМС). Пример содержимого в файле:

9734672311, 1, 13.02.2008 13:01:55, 300

9734672311, 2, 28.11.2012 01:32:30, #

9757282392, 5, 23.09.2013 19:14:00, 54

**Требования к представлению результатов работы программы**

Каждый файл хранится в том же каталоге, что и разработанная программа.

Формат файлов текстовый, каждый из них создан и редактируется текстовым редактором.

Содержание файла настройки определяется вариантом в таблице индивидуальных заданий (см. ниже).

Параметры настройки записываются в текстовый файл с именем Param.ini. Каждый параметр – в отдельной строке.

Результирующий отчет (результат обработки) записывается в файл с именем Отчёт.txt. Если информация отсутствует, вывести в файл строку «Нет данных».

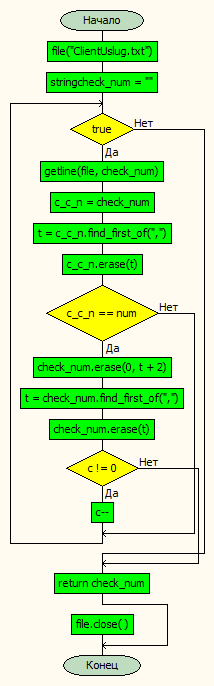
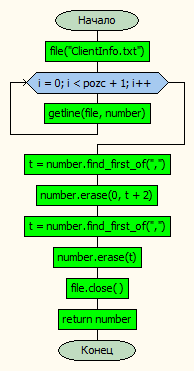
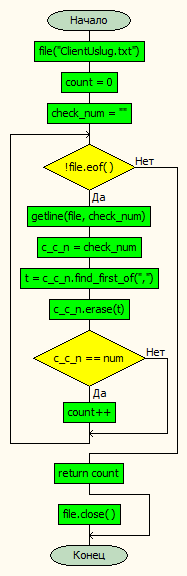
Вывести сообщение в окно программы о результатах её выполнения – есть или отсутствуют данные в результирующем файле. Язык сообщения английский либо русский на транслите.

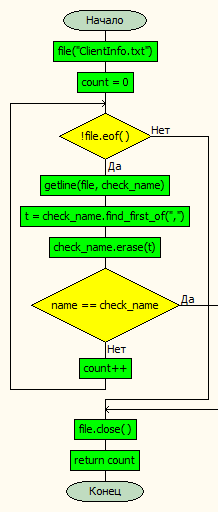
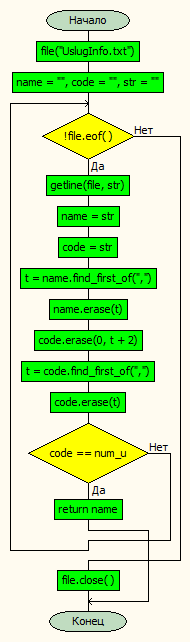
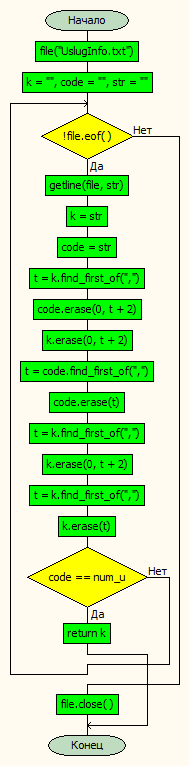
Формат данных в файле Отчёт.txt: каждая строка списка отделяется символами перевода строки. Если в строке несколько полей, они разделяются символами « , ».

# Выполнение работы (БЛОК-СХЕМА)

На рисункахРисунок 1 и Рисунок 2 представлена блок-схема программы.

Рисунок 1,2,3,4,5,6,7,8,9 – Блок-схема функций программы





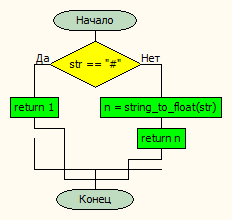
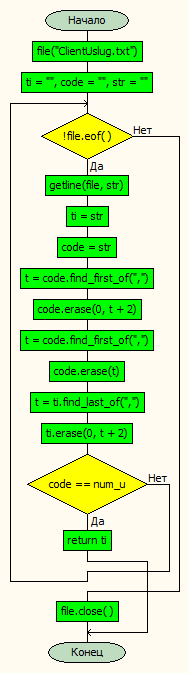
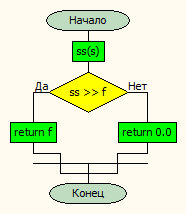


Рисунок 10 – Блок-схема программы

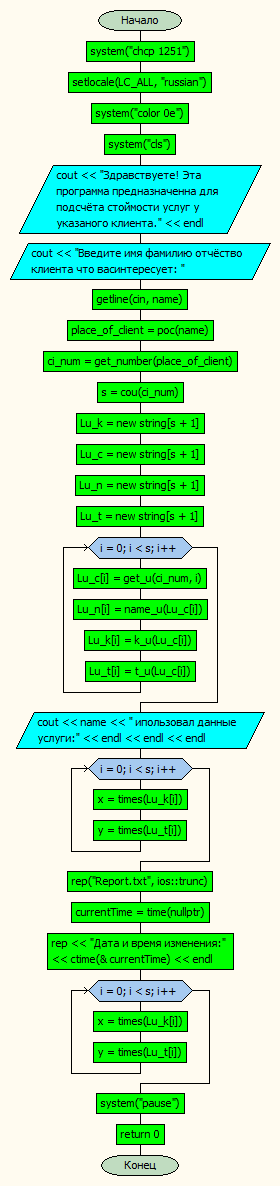


Рисунок 2 – Блок-схема программы ч.2

# Исходный код программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<string>

#include<fstream>

#include<Windows.h>

#include <sstream>

#include <ctime>

using namespace std;

//Функцмя для нахождения клиента в файле об информации о клиентах. return count;

int poc(string name) {

ifstream file("ClientInfo.txt");

string check\_name;

int count = 0;

while (!file.eof()) {

getline(file, check\_name);

int t = check\_name.find\_first\_of(",");

check\_name.erase(t);

if (name == check\_name) { break; }

count++;

}

file.close();

return count;

}

//Функция находящая номер клиента. return number;

string get\_number(int pozc) {

ifstream file("ClientInfo.txt");

string number;

for (int i = 0; i < pozc+1; i++) { getline(file, number); }

int t = number.find\_first\_of(",");

number.erase(0,t+2);

t = number.find\_first\_of(",");

number.erase(t);

file.close();

return number;

}

//Функция находящия колличество услуг у клиента. return count;

int cou(string num) {

ifstream file("ClientUslug.txt");

int count = 0;

string check\_num = "";

while (!file.eof()) {

getline(file, check\_num);

string c\_c\_n = check\_num;

int t = c\_c\_n.find\_first\_of(",");

c\_c\_n.erase(t);

if (c\_c\_n == num) {

count++;

}

}

return count;

file.close();

}

//функция находящая номер услуги клиента. return check\_num;

string get\_u(string num,int c) {

ifstream file("ClientUslug.txt");

string check\_num="";

while (true) {

getline(file, check\_num);

string c\_c\_n = check\_num;

int t = c\_c\_n.find\_first\_of(",");

c\_c\_n.erase(t);

if (c\_c\_n == num) {

check\_num.erase(0, t+2);

t = check\_num.find\_first\_of(",");

check\_num.erase(t);

if (c != 0) { c--; }

else { break; }

}

}

return check\_num;

file.close();

}

//Фуекция узнающая имя учлуги. return name;

string name\_u(string num\_u) {

ifstream file("UslugInfo.txt");

string name = "", code = "", str = "";

while (!file.eof()) {

getline(file, str);

name = str;

code = str;

int t = name.find\_first\_of(",");

name.erase(t);

code.erase(0, t+2);

t = code.find\_first\_of(",");

code.erase(t);

if (code == num\_u) {

return name;

}

}

file.close();

}

//Фуекция узнающая коалифицент учлуги. return k;

string k\_u(string num\_u) {

ifstream file("UslugInfo.txt");

string k = "", code = "", str = "";

while (!file.eof()) {

getline(file, str);

k = str;

code = str;

int t = k.find\_first\_of(",");

code.erase(0, t + 2);

k.erase(0, t + 2);

t = code.find\_first\_of(",");

code.erase(t);

t = k.find\_first\_of(",");

k.erase(0, t + 2);

t = k.find\_first\_of(",");

k.erase(t);

if (code == num\_u) {

return k;

}

}

file.close();

}

//Фуекция узнающая время учлуги. return ti;

string t\_u(string num\_u) {

ifstream file("ClientUslug.txt");

string ti = "", code = "", str = "";

while (!file.eof()) {

getline(file, str);

ti = str;

code = str;

int t = code.find\_first\_of(",");

code.erase(0, t + 2);

t = code.find\_first\_of(",");

code.erase(t);

t = ti.find\_last\_of(",");

ti.erase(0, t + 2);

if (code == num\_u) {

return ti;

}

}

file.close();

}

float string\_to\_float(string s) {

stringstream ss(s);

float f;

if (ss >> f) { return f; }

else { return 0.0; }

}

float times(string str) {

if (str == "#") {

return 1;

}

else { float n = string\_to\_float(str); return n; }

}

int main() {

system("chcp 1251");

setlocale(LC\_ALL, "russian");

system("color 0e");

system("cls");

cout << "Здравствуете! Эта программа предназначенна для подсчёта стоймости услуг у указаного клиента." << endl;

cout << "Введите имя фамилию отчёство клиента что васинтересует: ";

string name, secondname, thirdname;

//cin >> name >> secondname >> thirdname;

//name = secondname + " " + name + " " + thirdname;

getline(cin, name);

int place\_of\_client = poc(name);//poc - position of client

string ci\_num = get\_number(place\_of\_client);

int s = cou(ci\_num);

string\* Lu\_k = new string[s + 1];//калифицент

string\* Lu\_c = new string[s + 1];//код услуги

string\* Lu\_n = new string[s + 1];//имя услуги

string\* Lu\_t = new string[s + 1];//врямя использованя услуги

for (int i = 0; i < s; i++) {

Lu\_c[i] = get\_u(ci\_num, i);

Lu\_n[i] = name\_u(Lu\_c[i]);

Lu\_k[i] = k\_u(Lu\_c[i]);

Lu\_t[i] = t\_u(Lu\_c[i]);

}

cout << name << " ипользовал данные услуги:" << endl << endl << endl;

for (int i = 0; i < s; i++) {

float x = times(Lu\_k[i]);

float y = times(Lu\_t[i]);

cout << Lu\_n[i] << ", стоимость = " << x\*y << "." << endl << endl;

}

ofstream rep("Report.txt", ios::trunc);

time\_t currentTime = time(nullptr);

rep << "Дата и время изменения:" << ctime(&currentTime) << endl;

rep << name << " ипользовал данные услуги:" << endl << endl << endl;

for (int i = 0; i < s; i++) {

float x = times(Lu\_k[i]);

float y = times(Lu\_t[i]);

rep << Lu\_n[i] << ", стоимость = " << x \* y << endl << endl;

}

system("pause");

return 0;

}

# Пример работы программы

На рисунках 11-15 представлены данный входных файлов.

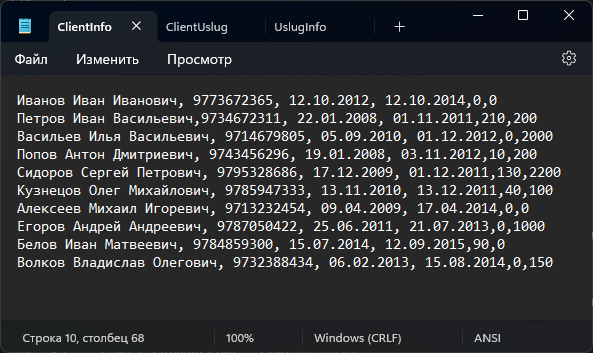


Рисунок 11 – Файл ClientInfo.txt

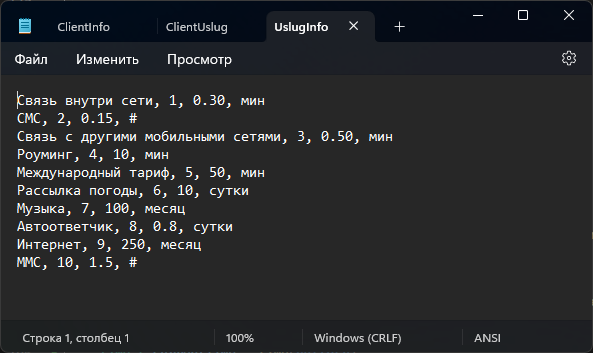


Рисунок 12 – Файл UslugInfo.txt

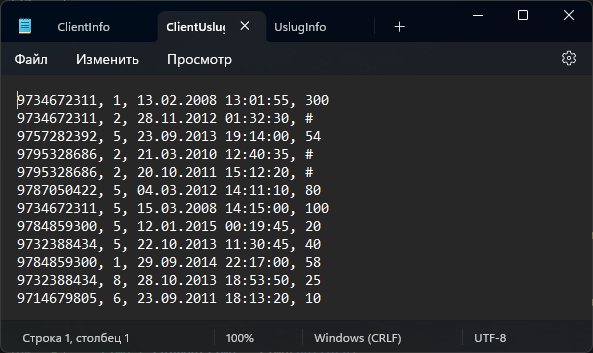


Рисунок 13 – ClientUslug.txt

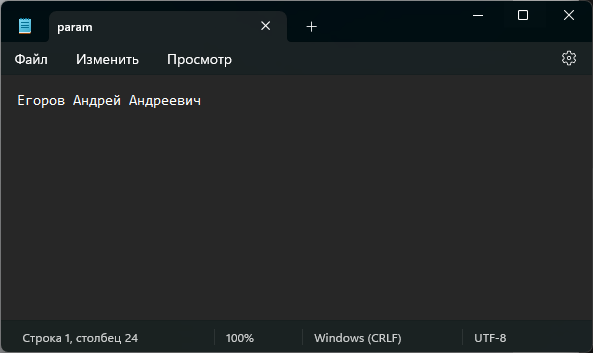


Рисунок 14 – param.ini.txt

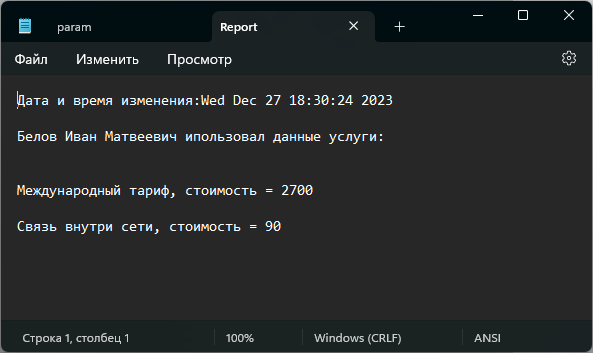


Рисунок 15 – Report.txt

Из рисунков мы можем убедиться в том, что программа работает корректно, исходя из данных в файлах видно, что пользователи Белов Иван Матвеевич ипользовал данные услуги: международный тариф, стоимость = 2700, связь внутри сети, стоимость = 90. Из чего следует, что все они должны быть записаны в файл Report.txt.

# Вывод

В ходе выполнения индивидуального, творческого задания, я разработал программу, которая, обрабатывая четыре входных файла, записывает данные того или иного пользователя в результирующий файл. При этом во время обработки информации также учитывается время, когда абонент пользовался услугами.

При выполнении индивидуального задания я научился закрепил знания и навыки такие как: работа с файлами, работа со структурами, циклы, обработка строк, выделение динамической памяти.