МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра информационно-аналитических систем безопасности

имени Л.С. Берштейна

**ОТЧЕТ**

**Основы алгоритмизации и программирования**

**Индивидуальное задание**

**Вариант №11**

Выполнил:

студент гр. КТбо1-8

Елоев Георгий

«\_\_\_» 2017 г.

Проверил:

Профессор кафедры ИАСБ

Беляков Станислав Леонидович

«\_\_\_» 2017 г.

Таганрог – 2017

Оглавление

[Цели работы 3](#_Toc501648410)

[Создание алгоритма 4](#_Toc501648411)

[Программный код 5](#_Toc501648412)

[Тестирование программы 6](#_Toc501648413)

[Вывод 7](#_Toc501648414)

# Цели работы

Написать программу, которая считывает настроечные параметры и формирует отчет по имеющимся данным в файлах. Отчет записывать в текстовый файл. Варианты заданий 11:

|  |  |  |
| --- | --- | --- |
| №варианта | Отчет | Параметры |
| 11 | Получить перечень услуг и телефонных номеров для заданных клиентов, с которыми заключён договор во 2-м квартале нынешнего года | Список ФИО клиентов |

# 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**

# **Создание алгоритма**



# Программный код

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <string.h>

#include <fstream>

#include <conio.h>

#include <locale>

#include <stdio.h>

#define LENGTH 512

char\* iniFile = "param.ini";

char\* clientsFile = "clients.txt";

char\* serviceFile = "service.txt";

char\* billFile = "bill.txt";

char\* reportFile = "report.txt";

int clientsSize = 0;

int serviceSize = 0;

int billSize = 0;

int keysSize = 0;

struct clients

{

char\* FIO = (char\*)(malloc(sizeof(char) \* LENGTH));

char\* phonenumber = (char\*)(malloc(sizeof(char) \* LENGTH));

char\* contractdate = (char\*)(malloc(sizeof(char) \* LENGTH));

char\* endcontractdate = (char\*)(malloc(sizeof(char) \* LENGTH));

float debt;

float credit;

};

struct service

{

char\* p\_servName = (char\*)(malloc(sizeof(char) \* LENGTH));

int code;

float cost;

char\* p\_Mera = (char\*)(malloc(sizeof(char) \* LENGTH));

};

struct bill

{

char\* p\_Number = (char\*)(malloc(sizeof(char) \* LENGTH));;

int code;

char\* p\_During = (char\*)(malloc(sizeof(char) \* LENGTH));;

char\* interval = (char\*)(malloc(sizeof(char) \* LENGTH));;

};

int getSize(char\* loc);

clients\* clientBase();

service\* serviceBase();

bill\* billBase();

char\*\* confIni();

bool is\_SecQuartal(char\* date);

void searchFunc(clients\* cBase, service\* sBase, bill\* bBase, char\*\* key);

void entryFile(FILE\* oFile, int a, int b, clients\* cBase, service\* sBase, bill\* bBase, char\*\* key);

int main()

{

setlocale(0, "rus");

clients\* CBase = clientBase();

service\* SBase = serviceBase();

bill\* BBase = billBase();

char\*\* keys = confIni();

searchFunc(CBase, SBase, BBase, keys);

printf("Конфигурация информации!\n");

system("pause");

return 0;

}

int getSize(char\* loc)

{

int \_size = 0;

FILE\* tmp = fopen(loc, "r");

char\* buff = (char\*)(malloc(sizeof(char) \* LENGTH));

while (!feof(tmp))

{

fgets(buff, LENGTH, tmp);

if (strcmp(buff, "\n"))

{

\_size++;

}

}

free(buff);

return \_size;

}

clients\* clientBase()

{

clientsSize = getSize(clientsFile);

clients\* CBase = (clients\*)(malloc(sizeof(clients)\*clientsSize));

FILE\* clntFile = fopen(clientsFile, "r");

int inc = 0;

while (!feof(clntFile))

{

char\* buff = (char\*)(malloc(sizeof(char) \* LENGTH));

fgets(buff, LENGTH, clntFile);

if (strcmp(buff, "\n"))

{

char\* tokken = NULL;

CBase[inc].FIO = strtok\_s(buff, ",", &tokken);

CBase[inc].phonenumber = strtok\_s(NULL, ", ", &tokken);

CBase[inc].contractdate = strtok\_s(NULL, ", ", &tokken);

CBase[inc].endcontractdate = strtok\_s(NULL, ",", &tokken);

CBase[inc].debt = atof(strtok\_s(NULL, ",", &tokken));

CBase[inc].credit = atof(strtok\_s(NULL, ",", &tokken));

inc++;

}

}

return CBase;

}

service\* serviceBase()

{

serviceSize = getSize(serviceFile);

service\* SBase = (service\*)(malloc(sizeof(service)\*serviceSize));

FILE\* SFile = fopen(serviceFile, "r");

int inc = 0;

while (!feof(SFile))

{

char\* buff = (char\*)(malloc(sizeof(char) \* LENGTH));

fgets(buff, LENGTH, SFile);

if (strcmp(buff, "\n"))

{

char\* tokken = NULL;

SBase[inc].p\_servName = strtok\_s(buff, ",", &tokken);

SBase[inc].code = atoi(strtok\_s(NULL, ",", &tokken));

SBase[inc].cost = atof(strtok\_s(NULL, ",", &tokken));

SBase[inc].p\_Mera = strtok\_s(NULL, ",", &tokken);

inc++;

}

}

return SBase;

}

bill\* billBase()

{

billSize = getSize(billFile);

bill\* BBase = (bill\*)(malloc(sizeof(service)\*billSize));

FILE\* BFile = fopen(billFile, "r");

int inc = 0;

while (!feof(BFile))

{

char\* buff = (char\*)(malloc(sizeof(char) \* LENGTH));

fgets(buff, LENGTH, BFile);

char\* tokken = NULL;

BBase[inc].p\_Number = strtok\_s(buff, ", ", &tokken);

BBase[inc].code = atoi(strtok\_s(NULL, ",", &tokken));

BBase[inc].p\_During = strtok\_s(NULL, ",", &tokken);

BBase[inc].interval = strtok\_s(NULL, ",", &tokken);

inc++;

}

return BBase;

}

char\*\* confIni()

{

keysSize = getSize(iniFile);

FILE\* conf = fopen(iniFile, "r");

char\*\* key = (char\*\*)(malloc(sizeof(char\*)\*keysSize));

int inc = 0;

while (!feof(conf))

{

char\* tokken = NULL;

char\* buff = (char\*)(malloc(sizeof(char) \* LENGTH));

fgets(buff, LENGTH, conf);

key[inc] = strtok\_s(buff, "\n", &tokken);

inc++;

}

return key;

}

bool is\_SecQuartal(char\* date)

{

if (date[9] == '7'&&date[8] == '1'&&date[7] == '0'&&date[6] == '2' && date[4]>'2' && date[4]<'6'&&date[3] == '0')

return true;

else return false;

}

void searchFunc(clients\* cBase, service\* sBase, bill\* bBase, char\*\* key)

{

FILE\* oFile = fopen(reportFile, "w+");

fputs("Фамилия: --- Номер телефона ---- Дата заключения договора ---- Услуги -----\n", oFile);

for (int a = 0; a < keysSize; a++)

{

for (int b = 0; b < clientsSize; b++)

{

entryFile(oFile, a, b, cBase, sBase, bBase, key);

}

}

}

void entryFile(FILE\* oFile, int a, int b, clients\* cBase, service\* sBase, bill\* bBase, char\*\* key)

{

if (!strcmp(cBase[b].FIO, key[a]) && is\_SecQuartal(cBase[b].contractdate))

{

fputs(key[a], oFile);

fputs(" ", oFile);

fputs(cBase[b].phonenumber, oFile);

fputs (" ", oFile);

fputs(cBase[b].contractdate, oFile);

fputs(" :", oFile);

for (int c = 0; c < billSize; c++)

{

if (!strcmp(cBase[b].phonenumber, bBase[c].p\_Number))

{

for (int d = 0; d < serviceSize; d++)

{

if (bBase[c].code == sBase[d].code)

{

fputs(sBase[d].p\_servName, oFile);

fputs("; ", oFile);

}

}

}

}

fputs("\n", oFile);

}

}

# Тестирование программы

Входные данные содержащиеся в файлах представлены на рисунках 1-4.

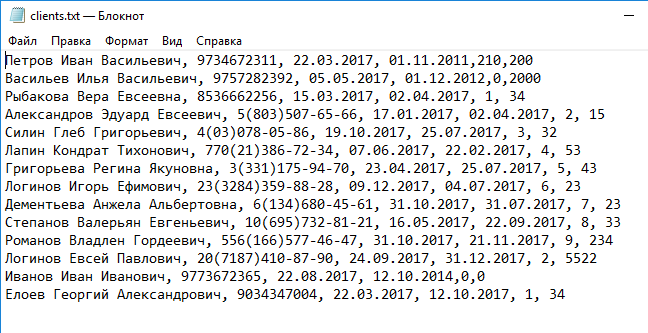


рис.1 содержимое файла информации о клиентах (clients.txt)

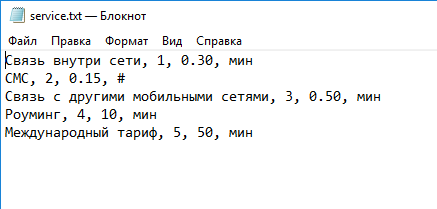


рис.2 содержимое файла информации об услугах (service.txt)

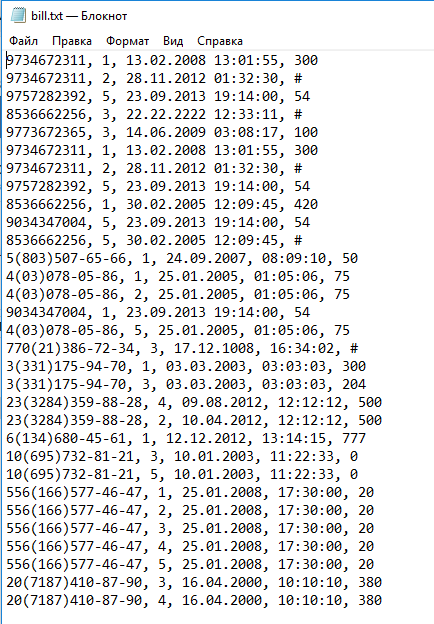


рис.3 содержимое файла информации об услугах, оказанных клиентам(bill.txt)

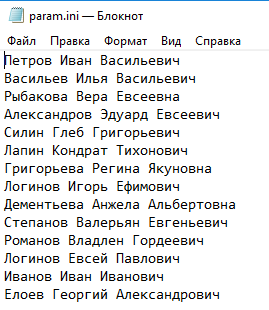


рис.4 содержимое файла параметров

Выходные данные представлены на рисунках 5-6.

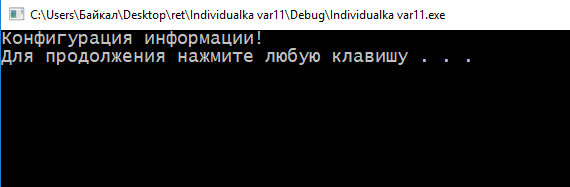


рис.5 работа программы

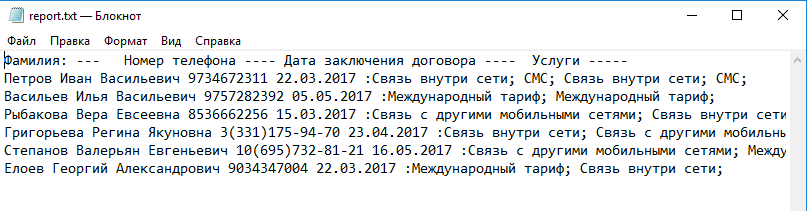


рис.6 результат работы программы

# Вывод

В результате выполнения индивидуального задания, приобрели практические навыки в программировании программ, а также улучшили свои знание языка C.