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**

**По индивидуальному заданию**

По курсу «Основы АиП»

Вариант №4

Выполнил:

студент гр. КТбо1-8

Васильченко Н.С.

«\_\_\_» 2017 г.

Проверил:

Профессор ИКТИБ

кафедры ИАСБ

Беляков С.Л

«\_\_\_» 2017 г.

**Таганрог – 2017**

Оглавление

[Вариант задания 2](#_Toc501651933)

[Алгоритм работы 3](#_Toc501651934)

[Входные данные 4](#_Toc501651935)

[Результат работы 4](#_Toc501651936)

[Заключение 5](#_Toc501651937)

[Код программы 5](#_Toc501651938)

# Вариант задания

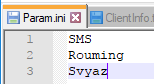
Разработать программу, которая считывает настроечные параметры и выводит в файл ФИО клиентов, пользовавшихся указанной услугой в дневное время (с 12:00 до 18:00) по имеющимся данным в файлах. Число записей в каждом из описанных выше файлов произвольно.

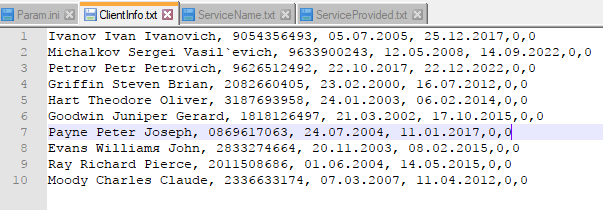
# Алгоритм работы

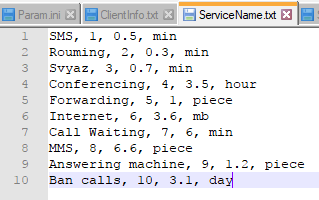
Построим блок-схему

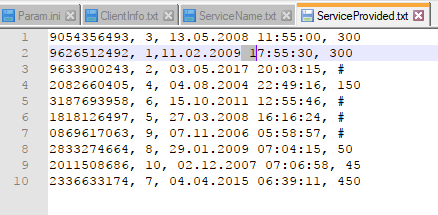


Входные данные

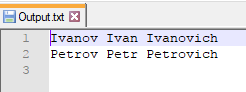








# Результат работы



# Заключение

В данной работе мы научились разрабатывать программу, которая считывает настроечные параметры и формирует отчет по имеющимся данным в файлах. Также отчет записывать в текстовый файл.

# Код программы

#include "stdafx.h"

#include "iostream"

#include <fstream>

#include "string"

#include "vector"

using namespace std;

vector<string> Params;

vector<string> InfoClient;

vector<string> ServiceName;

vector<string> InfoServiceProvided;

vector<string> ParamNumber;

vector<string> ClientsNames;

void \_\_ParamsRead\_\_()

{

string text = {};

int first = 0;

ifstream TextInput("D:/projects/IndZad/Debug/Param.ini");

while (!TextInput.eof())

{

getline(TextInput, text, '\n');

Params.insert(Params.end(), text);

}

TextInput.close();

}

void \_\_SearchParamNuber\_\_()

{

for (int i = 0; i < Params.size(); i++)

{

for (int m = 0; m < ServiceName.size()-1; m=m+2)

{

if (Params[i] == ServiceName[m])

{

ParamNumber.insert(ParamNumber.end(), ServiceName[m+1]);

}

}

}

}

void \_\_InfoClientRead\_\_()

{

string text = {};

int Count = 0;

int pos = 0;

ifstream TextInput("D:/projects/IndZad/Debug/ClientInfo.txt");

while (!TextInput.eof())

{

getline(TextInput, text, '\n');

for (int i = 0; i < text.length(); i++)

{

if ((text[i] == ',')||(i==text.length()-1))

{

Count += 1;

if (Count < 3)

{

InfoClient.insert(InfoClient.end(), text.substr(pos, i - pos));

pos = i + 2;

}

}

}

Count = 0;

pos = 0;

}

TextInput.close();

}

void \_\_ServiceNameRead\_\_()

{

string text = {};

int Count = 0;

int pos = 0;

ifstream TextInput("D:/projects/IndZad/Debug/ServiceName.txt");

while (!TextInput.eof())

{

getline(TextInput, text, '\n');

for (int i = 0; i < text.length(); i++)

{

if ((text[i] == ',') || (i == text.length() - 1))

{

Count += 1;

if (Count < 3)

{

ServiceName.insert(ServiceName.end(), text.substr(pos, i - pos));

pos = i + 2;

}

}

}

Count = 0;

pos = 0;

}

TextInput.close();

}

void \_\_InfoServiceProvidedRead\_\_()

{

string text = {};

int pos = 0;

ifstream TextInput("D:/projects/IndZad/Debug/ServiceProvided.txt");

while (!TextInput.eof())

{

getline(TextInput, text, '\n');

for (int i = 0; i < text.length(); i++)

{

if (text[i] == ',')

{

InfoServiceProvided.insert(InfoServiceProvided.end(), text.substr(pos, i - pos));

pos = i + 2;

}

else if (i == text.length() - 1)

{

InfoServiceProvided.insert(InfoServiceProvided.end(), text.substr(pos, i - pos + 1));

}

}

pos = 0;

}

TextInput.close();

}

void \_\_TimeSort\_\_()

{

int len = 0;

for (int i = 2; i < InfoServiceProvided.size() - 1; i = i + 4)

{

len = InfoServiceProvided[i].length();

for (int m = 0; m < len - 1; m++)

{

string Hours;

string Minutes;

string Seconds;

int Time = 0;

int Pos = 0;

if (InfoServiceProvided[i][m] == ' ')

{

InfoServiceProvided[i] = InfoServiceProvided[i].substr(m + 1, len - m - 1);

Hours = InfoServiceProvided[i].substr(0, 2);

Minutes = InfoServiceProvided[i].substr(3, 2);

Seconds = InfoServiceProvided[i].substr(6, 2);

InfoServiceProvided[i] = to\_string(atoi(Hours.c\_str()) \* 3600 + atoi(Minutes.c\_str()) \* 60 + atoi(Seconds.c\_str()));

}

len = InfoServiceProvided[i].length();

}

}

}

void \_\_Search\_\_()

{

int AddTime = 0;

int DayStart = 43200;

int DayFinist = 64800;

for (int i = 1; i < InfoServiceProvided.size(); i = i + 4)

{

if (InfoServiceProvided[i + 2] != "#")

{

AddTime = atoi(InfoServiceProvided[i + 2].c\_str());

}

for (int m = 0; m < ParamNumber.size(); m++)

{

if (InfoServiceProvided[i] == ParamNumber[m])

{

int SubTime = atoi(InfoServiceProvided[i + 1].c\_str());

if (((SubTime >= DayStart) && (SubTime <= DayFinist))||((SubTime<DayStart)&&(SubTime+AddTime>=DayStart)))

{

for (int n = 1; n < InfoClient.size(); n = n + 2)

{

if (InfoClient[n] == InfoServiceProvided[i - 1])

{

ClientsNames.insert(ClientsNames.end(), InfoClient[n - 1]);

}

}

}

}

}

}

}

void \_\_OutInFile\_\_()

{

ofstream TextOut("D:/projects/IndZad/Debug/Output.txt");

for (int i = 0; i < ClientsNames.size(); i++)

{

TextOut << ClientsNames[i] << "\n";

}

TextOut.close();

}

int main()

{

\_\_ParamsRead\_\_();

\_\_ServiceNameRead\_\_();

\_\_SearchParamNuber\_\_();

\_\_InfoClientRead\_\_();

\_\_InfoServiceProvidedRead\_\_();

\_\_TimeSort\_\_();

\_\_Search\_\_();

\_\_OutInFile\_\_();

return 0;

}