МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Кафедра физико-математических основ инженерного образования



**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №1**

по курсу «Алгоритмизация и программирование»

**ВАРИАНТ 4**

Выполнил:

студент гр. КТбо2-1

Пономарев Д. Ю.  
Проверил:

Доцент кафедры САиТ

Свиридов А. С.

Таганрог 2023

**Содержание**

[**1.** **Техническое задание** 3](#_Toc153296124)

[**2.** **Описание задания** 3](#_Toc153296125)

[**3.** **Требования к представлению результатов работы программы** 3](#_Toc153296126)

[**4.** **Выполнение работы** 3](#_Toc153296127)

[**5.** **Исходный код программы** 3](#_Toc153296128)

[**6.** **Тексты файлов данных** 3](#_Toc153296129)

[**7.** **Результат работы программы** 3](#_Toc153296130)

[**8.** **Вывод** 3](#_Toc153296131)

# **Техническое задание**

Программирование на языке Си/С++. Задание состоит в разработке программы, которая считывает настроечные параметры и формирует отчет по имеющимся данным в файлах. Отчет записывается в текстовый файл. Число записей в каждом из описанных выше файлов произвольно.

# **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.

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).

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).

# **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. Параметры настройки записываются в текстовый файл с именем P aram.ini. Каждый параметр – в отдельной строке.

Даны минимальное и максимальное значения диапазона. Получить перечень услуг, предоставленных в текущем месяце, с суммарным значением в заданном диапазоне значений. Результирующий отчет (результат обработки) записывается в файл с именем Report.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 Report.txt: каждая строка списка отделяется символами перевода строки. Если в строке несколько полей, они разделяются символами «,».

# **Выполнение работы**

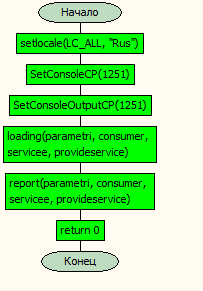
**Main:**

Рисунок 1 (Блок-схема[main])

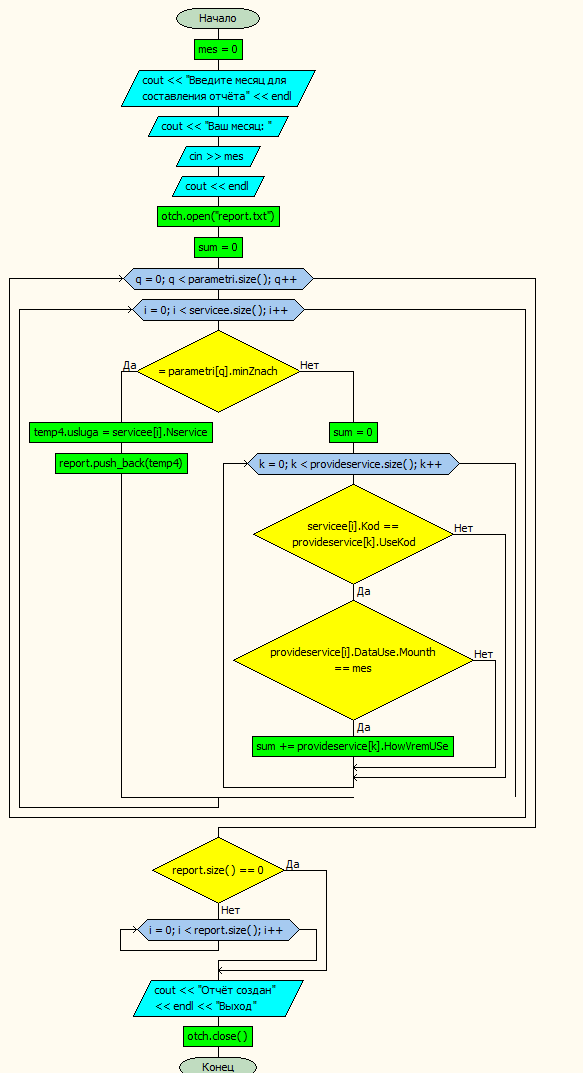
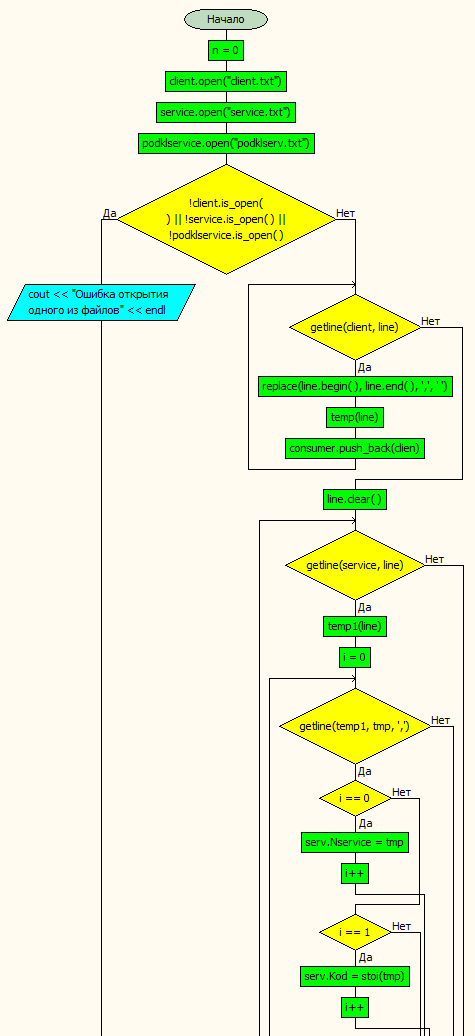
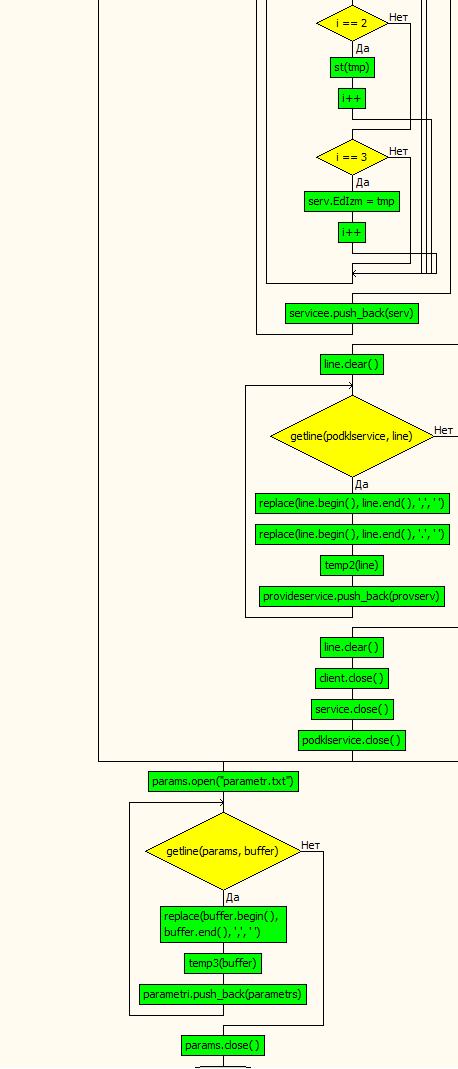
**Функция Report:**

Рисунок 2 (Блок-схема[report])

**Функция loading:**



Рисунок 3 (Блок-схема[loading])



1. **Исходный код программы**

**Main.cpp:**

#include <iostream>

#include <algorithm>

#include <string>

#include <fstream>

#include <sstream>

#include <windows.h>

#include <vector>

using namespace std;

struct Data {

int Day = 0;

int Mounth = 0;

int Year = 0;

};

struct Clients {

string Name;

string Surname;

string Otchestvo;

string Phone;

string Nacahlo;

string Konez;

int Dolg = 0;

int Vdolg = 0;

};

struct Service {

string Nservice;

int Kod = 0;

float Cost = 0;

string EdIzm;

};

struct PodklService {

string Phone;

int UseKod = 0;

struct Data DataUse;

string VremUse;

int HowVremUSe;

};

struct Report

{

string usluga;

};

struct Parametr

{

int minZnach = 0;

int maxZnach = 0;

};

void loading(vector<Parametr>& parametri, vector<Clients>& consumer, vector <Service>& servicee, vector <PodklService>& provideservice);

void report(vector<Parametr>& parametri, vector<Clients>& consumer, vector <Service>& servicee, vector <PodklService>& provideservice);

void loading(vector<Parametr>& parametri, vector<Clients>& consumer, vector <Service>& servicee, vector <PodklService>& provideservice) {

ifstream client;

int n = 0;

client.open("client.txt");

ifstream service;

service.open("service.txt");

ifstream podklservice;

podklservice.open("podklserv.txt");

if (!client.is\_open() || !service.is\_open() || !podklservice.is\_open())

{

cout << "Ошибка открытия одного из файлов" << endl;

}

else

{

Clients clien;

Service serv;

PodklService provserv;

string line;

while (getline(client, line))

{

replace(line.begin(), line.end(), ',', ' '); istringstream temp(line);

temp >> clien.Surname >> clien.Name >> clien.Otchestvo >> clien.Phone >> clien.Nacahlo >> clien.Konez >> clien.Dolg >> clien.Vdolg; consumer.push\_back(clien);

}

line.clear();

while (getline(service, line))

{

stringstream temp1(line);

string tmp;

int i = 0;

while (getline(temp1, tmp, ','))

{

if (i == 0)

{

serv.Nservice = tmp;

i++;

continue;

}

if (i == 1)

{

serv.Kod = stoi(tmp);

i++;

continue;

}

if (i == 2)

{

stringstream st(tmp);

st >> serv.Cost;

i++;

continue;

}

if (i == 3)

{

serv.EdIzm = tmp;

i++;

continue;

}

}

servicee.push\_back(serv);

}

line.clear();

while (getline(podklservice, line))

{

replace(line.begin(), line.end(), ',', ' ');

replace(line.begin(), line.end(), '.', ' ');

istringstream temp2(line);

temp2 >> provserv.Phone >> provserv.UseKod >> provserv.DataUse.Day >> provserv.DataUse.Mounth >> provserv.DataUse.Year >> provserv.VremUse >> provserv.HowVremUSe;

provideservice.push\_back(provserv);

}

line.clear();

client.close();

service.close();

podklservice.close();

}

Parametr parametrs;

ifstream params;

string buffer;

params.open("parametr.txt");

while (getline(params, buffer))

{

replace(buffer.begin(), buffer.end(), ',', ' ');

istringstream temp3(buffer);

temp3 >> parametrs.minZnach >> parametrs.maxZnach;

parametri.push\_back(parametrs);

}

params.close();

}

void report(vector<Parametr>& parametri, vector<Clients>& consumer, vector <Service>& servicee, vector <PodklService>& provideservice)

{

vector <Report> report;

Report temp4;

int mes = 0;

cout << "Введите месяц для составления отчёта" << endl;

cout << "Ваш месяц: ";

cin >> mes;

cout << endl;

Parametr fuka;

ofstream otch;

otch.open("report.txt");

int sum = 0;

for (int q = 0; q < parametri.size(); q++)

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

{

if (sum <= parametri[q].maxZnach && sum >= parametri[q].minZnach)

{

temp4.usluga = servicee[i].Nservice;

report.push\_back(temp4);

}

else

{

sum = 0;

for (int k = 0; k < provideservice.size(); k++)

if (servicee[i].Kod == provideservice[k].UseKod)

if (provideservice[i].DataUse.Mounth == mes)

sum += provideservice[k].HowVremUSe;

}

}

if (report.size() == 0)

{

otch << "Нет данных." << endl;

}

else

{

otch << "Список услуг, предоставленных в " << mes << " месяце, которые попадают в задный диапозон." << endl;

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

{

otch << report[i].usluga << endl;

}

}

cout << "Отчёт создан" << endl << "Выход";

otch.close();

}

int main() {

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector <Clients> consumer;

vector <Service> servicee;

vector <PodklService> provideservice;

vector <Parametr> parametri;

loading(parametri, consumer, servicee, provideservice);

report(parametri, consumer, servicee, provideservice);

return 0;

}

# **Тексты файлов данных**

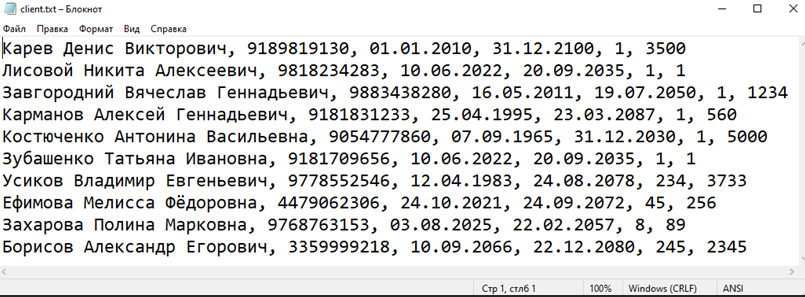
**Client.txt:**

Рисунок 4 (данные из файла)

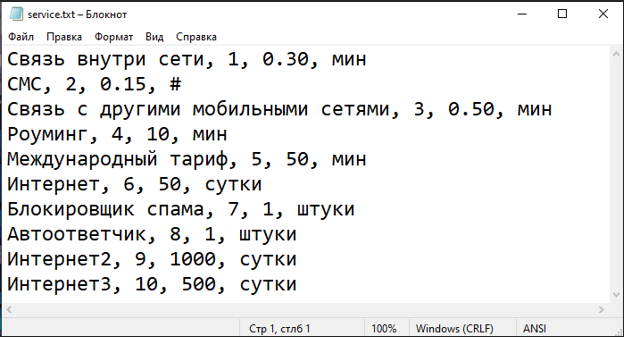
**Service.txt:**

Рисунок 5 (данные из файла)

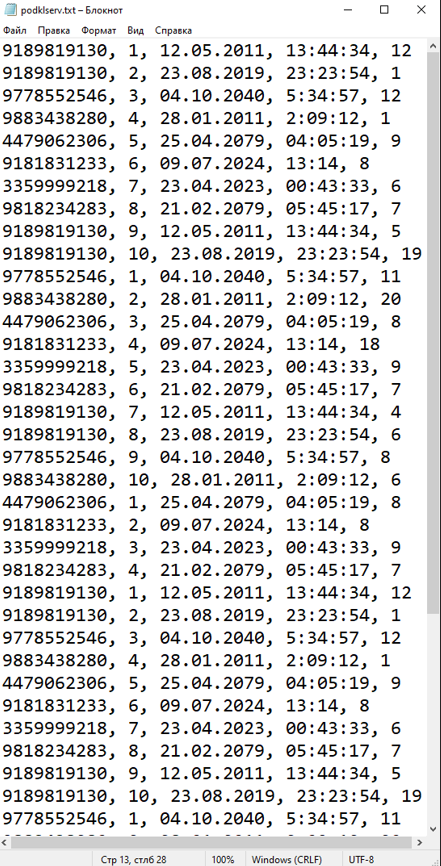
**Podklserv.txt:**

Рисунок 6 (данные из файла)

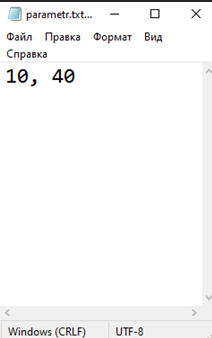
**Param.ini:**

Рисунок 7 (данные из файла)

# **Результат работы программы**

**Изображение выглядит как текст, снимок экрана

Автоматически созданное описаниеReport.txt:**

Рисунок 8 (Результирующие данные в файле)

# **Вывод**

В ходе выполнения индивидуальной работы по дисциплине алгоритмизация и программирования я познакомился и изучил такие темы, как работа с файлами и структурами.