

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Дальневосточный федеральный университет»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент программной инженерии и искусственного интеллекта

**ОТЧЕТ**

**о выполнении индивидуального задания по теме «Записи»**

в рамках освоения дисциплины

«Основы алгоритмизации и программирования»

направления «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| Отчет защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_ФИО\_руководителя         (подпись)  «\_26\_» апрель 2025 г. |  | Выполнил студент  группы Б9123-09.03.04  Д.С. Глумова\_\_ И.О. Фамилия  (подпись) |
|  |  | Руководитель  ученая степень, ученое звание, должность ФИО\_руководителя |

Владивосток

2025

**Аннотация**

Алгоритм решения задачи реализован в виде программного кода, написанного на алгоритмическом языке C++

Общее назначение программы: чтение и анализ входных данных, предоставление статистики по заданным параметрам.

Проект программы разработан с использованием принципов нисходящего проектирования; алгоритм программы основан на принципеструктурного программирования.

**Оглавление**

[Неформальная постановка задачи 4](#_Toc95420537)

[Формальная постановка задачи 4](#_Toc95420538)

[Алгоритм решения задачи 4](#_Toc95420539)

[Спецификация данных 4](#_Toc95420540)

[Спецификация функций 4](#_Toc95420541)

[Проектирование программы 5](#_Toc95420542)

[Описание данных программы 5](#_Toc95420543)

[Алгоритм программы на языке PDL 5](#_Toc95420544)

[Тесты 5](#_Toc95420545)

[Тестирование программы 5](#_Toc95420546)

[Список использованных источников 6](#_Toc95420547)

Неформальная постановка задачи

Реализовать программу, решающую следующую задачу:

Одна запись детализации разговоров с некоторого мобильного номера состоит из даты (YYYY.MM.DD), времени начала разговора (HH:MM:SS), времени окончания разговора (HH:MM:SS), номера абонента (только 11 цифр) и формы соединения (входящий /

исходящий). Найти:

1) количество и общую длительность входящих и исходящих соединений с заданным номером абонента;

2) запись о самом длительном разговоре;

3) дату и номер абонента, с которым состоялся самый короткий разговор.

Формальная постановка задачи

**Input:**

**N(D, T1, T2, P, V)** – множество записей, где:

**D = {d1, d2, …, dn} (**гдеn – количество записей в множестве N) D – множество, содержащее информацию о дате звонка, в котором di () представляет собой строку, состоящую только из последовательности цифр и символа . , имеющую вид (YYYY.MM.DD), где:

YYYY – подстрока, представляющая собой год, когда была произведена запись, подстрока должна состоять только из 4 подряд идущих цифр без пробелов, число

MM – подстрока, представляющая собой месяц, когда была произведена запись, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, который принимает только значения {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”, “10”, “11”, “12”}

DD - подстрока, представляющая собой день, когда была произведена запись, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, DD может принимать значения [10, 31], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}

**T1 = {t11, t12, …, t1n}** – **(**гдеn – количество записей в множестве N) множество содержащее информацию о времени начала звонка, в котором ti  () представляет собой строку, состоящую только из последовательности цифр и символа : , имеющую вид (HH2:MM2:SS2), где:

HH1 – подстрока, представляющая собой час, в который был произведен звонок, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, HH1 может принимать значения [10, 23], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}.

MM1 – подстрока, представляющая собой минуту, в которую был произведен звонок, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, MM1 может принимать значения [10, 59], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}.

SS1 - подстрока, представляющая собой секунду, в которую был произведен звонок, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, SS1 может принимать значения [10, 59], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}.

**T2 = {t21, t21, …, t21}** – **(**гдеn – количество записей в множестве N) множество содержащее информацию о времени начала звонка, в котором t2i () представляет собой строку, состоящую только из последовательности цифр и символа : , имеющую вид (HH:MM:SS), где:

HH2– подстрока, представляющая собой час, в который звонок был завершён, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, HH может принимать значения[10, 23], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}.

MM2 – подстрока, представляющая собой минуту, в которую звонок был завершён, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, MM2 может принимать значения [10, 59], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}.

SS2 – подстрока, представляющая собой секунду, в которую звонок был завершён, подстрока должна состоять только из 2 подряд идущих цифр без пробелов, SS2 может принимать значения [10, 59], а также {“01”, “02”, “03”, “04”, “05”, “06”, “07”, “08”, “09”}.

**P = {p1, p2, …, pn}** – **(**гдеn – количество записей в множестве N) множество содержащее информацию о номерах телефона, с которого был совершён или принят звонок, в котором pi () представляет собой набор из 11 подряд идущих цифр без пробелов и специальных символов.

**V = {v1, v2, …, vn}** – **(**гдеn – количество записей в множестве N) множество содержащее информацию о том, является звонок входящим или исходящим, vi [] представляет собой строку «ishodyashi» или строку «vhodyashi» в зависимости от типа звонка.

InputNum – строка, введённая пользователем, содержащая 11 цифр без пробелов и специальных символов.

**Output:**

1. **QualityAndTime** – строка, содержащая количество и общую длительность входящих и исходящих соединений с заданным номером абонента.
2. **MaxTimeCall** – строка, содержащая запись о самом длительном разговоре.
3. RecordMinTimeCall – строка, содержащая дату и номер абонента, с которым состоялся самый короткий разговор

**Связи**

Dif = {dif1, dif2, …, difn} – множество, содержащее информацию о длительности каждого звонка, difi (, (n – количество записей в множестве N)

представляет собой число – длительность каждого звонка

- длительность каждого звонка с заданным номером, где

, где tij (см. в п. Input), где (n – количество записей в множестве N)

– максимальная длительность звонка

Алгоритм решения задачи

1. Начало
2. Ввод данных о записи (дате, времени начала, времени конца звонка, номере абонента и типа звонка) из текстового файла:
   1. Если файл не открывается, то вывести сообщение об ошибке и перейти к п. 10, иначе - продолжить.
   2. Если файл пуст, то вывести сообщение об ошибке и перейти к п. 10, иначе - продолжить

Каждая запись содержит: дату, время начала, время окончания, номер абонента и тип звонка.

1. Ввод номера абонента пользователем с клавиатуры.

4. Выполнить проверку ввода:

4.1. Если длина номера ≠ 11, то вывести сообщение об ошибке и перейти к п. 10.

4.2. Если номер содержит нецифровые символы, то вывести сообщение об ошибке и перейти к п. 10.

5. Если номер абонента совпадает с введённым:

5.1. Увеличить счётчик звонков.

5.2. Вычислить длительность звонка (в секундах).

5.3. В зависимости от типа звонка вычислить общую длительность входящих и исходящих вызовов абонента.

6. Поиск самого длительного звонка:

6.1. Вычислить длительность текущего разговора.

6.2. Найти максимум между этим разговором и текущим максимумом времени разговора, запомнить индекс этой записи.

6.3. Перейти к следующей записи.

7. Поиск самого короткого разговора:

7.1. Вычислить длительность разговора.

7.2. Найти минимум между этим разговором и текущим минимумом времени разговора, запомнить индекс этой записи.

7.3 Перейти к следующей записи.

8. Пока пользователь не завершит выполнение программы:

8.1. Вывести меню с вариантами действий.

8.2. В зависимости от варианта:

8.2.а) 1: Вывести количество звонков и общее время входящих/исходящих вызовов.

8.2.б) 2: Вывести данные о самом длинном разговоре.

8.2.в) 3: Вывести дату и номер абонента самого короткого разговора.

8.2.г) 4: Завершить программу.

8.2.д) Если был выбран другой вариант, то вывести сообщение об ошибке.

9. Завершение работы программы.

10. Конец.

Спецификация данных

**Спецификация ввода**

PhoneBook.txt – текстовый файл, содержащий строки с данными о телефонных звонках. Порядок данных имеет значение:

<date>

<timeStart>

<timeEnd>

<phone>

<callType>

**Спецификация вывода**

Вывод всех следующих сообщений выполняется на экран. В текстовый файл выводятся только сообщения 1 – 3.

Сообщение 1: «Enter phone number: »

Сообщение 2: «The number does not have 11 digits»

Сообщение 3: «Invalid characters in the number, the number must consist only of numbers »

Сообщение 4: « Failed to open file»

Сообщение 5: «1. Find total calls number and time of incoming and outgoing calls»

Сообщение 6: «2. Record of the longest call»

Сообщение 7: «3. Date and number of the subscriber with whom the shortest conversation took place»

Сообщение 8: «4. Leave»

Сообщение 9: « Quality calls: » + <cntCallsNumber> + «; Time incoming call: » + <callTimeVh> + «; Time outgoing call: » + <callTimeIs>

Сообщение 10: «Record of the longest call» + <call[indMx].date> + < call[indMx].timeStart> + « » + < call[indMx].timeEnd> + « » + <call[indMx].phone> + « » + <call[indMx].callType>

Сообщение 11: «Date » + < call[indMn].date> + «Number » + <call[indMn].phone>

Сообщение 12: «Error»

Сообщение 13: «File is Empty»

Сообщение 14: «Some fields empty »

Сообщение 15: «Incorrect number of action»

Спецификация функций

1. Считывание данных из файла и их проверка.
2. Ввод данных пользователем с клавиатуры и их проверка.
3. Перевод времени начала и окончания звонка в секунды.
4. Вычисление общей длительности и количества входящих и исходящих звонкой.

5. Поиск самого длительного звонка.

6. Поиск самого короткого разговора.

7. Вывод меню действий.

8. В зависимости от варианта:

а) 1: Вывести количество звонков и общее время входящих/исходящих вызовов.

б) 2: Вывести данные о самом длинном разговоре.

в) 3: Вывести дату и номер абонента самого короткого разговора.

г) 4: Завершить программу.

д) Если был выбран другой вариант, то вывести сообщение об ошибке.

9. Вывод сообщения об ошибке.

Проектирование программы

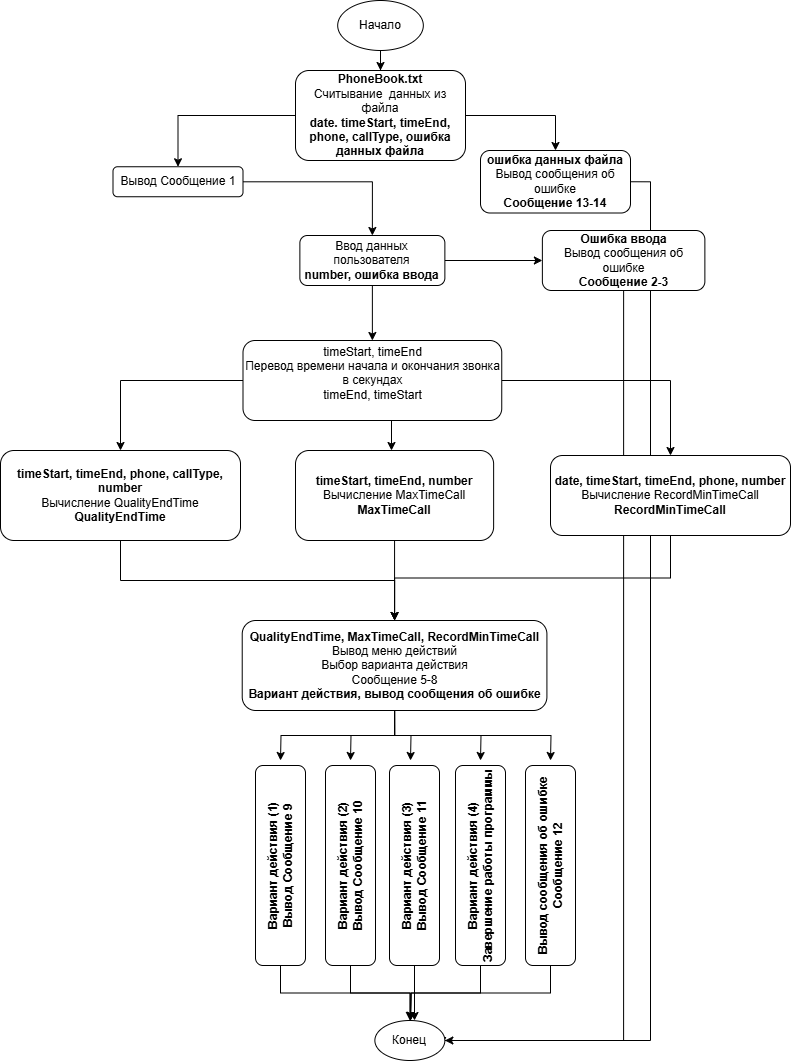


Таблица 1 - Описание данных программы

|  |  |  |
| --- | --- | --- |
| Идентификатор | Назначение | Описание |
| cntCallNumber | Переменная, хранящая в себе количество звонков номера | integer |
| callTime | Переменная, хранящая в себе длительность звонка | integer |
| mxCallTime | Переменная, хранящая в себе максимальную длительность звонка | integer |
| mnCallTime | Переменная, хранящая в себе минимальную длительность звонка | integer |
| indMx | Переменная, хранящая в себе индекс записи, в которой был совершён самый долгий звонок | integer |
| indMn | Переменная, хранящая в себе индекс записи, в которой был совершён самый короткий звонок | integer |
| startTime | Переменная, хранящая в себе время начала звонка | integer |
| endTime | Переменная, хранящая в себе время окончания звонка | integer |
| line | Переменная, хранящая каждую прочитанную из файла строку | string |
| number | Переменная, хранящая введённый пользователем номер, для которого будет предоставлена информация | string |
| choice | Переменная, хранящая введённое пользователем число – номер команды, которую необходимо выполнить | intrger |
| Идентификатор | Назначение | Описание |
| isEmpty | Переменная, хранящая булевое значение, показывающее пуст ли файл | bool |
| Локальный контекст структуры Reсord | | |
| date | Переменная, хранящая в себе дату звонка | string |
| timeStart | Переменная, хранящая в себе время начала звонка | string |
| timeEnd | Переменная, хранящая в себе время конца звонка | string |
| phone | Переменная, хранящая в себе номер телефона абонента | String |
| callType | Переменная, хранящая в себе значение «ishodyashi» или «vhodyashi» в зависимости от типа звонка | string |
| Локальный контекст функции TimeInSecond (функция возвращает значение integer – общее количество секунд в заданном количестве часов, минут и секунд) | | |
| str | Строка, в которой записано время в формате HH.MM.SS | string |

Алгоритм программы на языке PDL

#include <fstream>;

#include <iostream>;

#include <vector>;

#include <string>;

#include <set>;

using namespace std;

struct Records {

string date;

string timeStart;

string timeEnd;

string phone;

string callType;

};

int TimeInSeconds(string &str) {

return stoi(str.substr(0, 2)) \* 3600 + stoi(str.substr(3, 2)) \* 60 + stoi(str.substr(6, 2));

}

int main() {

ifstream file("PhoneBook.txt");

Records call[10];

int count = 0;

int cntCallsNumber = 0;

int callTimeVh = 0;

int callTimeIs = 0;

int mxTimeCall = 0;

int mnTimeCall = 100 \* 100;

int indMx = 0;

int indMn = 0;

int startTime = 0;

int endTime = 0;

int choice;

string line;

string number;

bool isEmpty = true;

set<char> allowedChar = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

if (!file.is\_open()) {

cerr << "Failed to open file" << endl;

}

while (getline(file, line)) {

if (!line.empty()) {

isEmpty = false;

}

}

if (isEmpty) {

cout << "File is Empty";

return 0;

}

file.clear();

file.seekg(0);

while (count < 10 && file

>> call[count].date

>> call[count].timeStart

>> call[count].timeEnd

>> call[count].phone

>> call[count].callType) {

count++;

};

while (count < 10) {

if (call[count].date.empty() || call[count].timeStart.empty() ||

call[count].timeEnd.empty() || call[count].phone.empty() ||

call[count].callType.empty()) {

cout << "Some fields empty " << endl;

return 0;

}

count++;

};

count = 0;

std:: cout << "Enter phone number: ";

cin >> number;

for (char c : number) {

if (allowedChar.find(c) == allowedChar.end()) {

std::cout << "Invalid characters in the number, the number must consist only of numbers" << endl;

return 0;

}

else if (number.length() != 11) {

std::cout << "The number does not have 11 digits" << endl;

return 0;

}

}

file.close();

for (int i = 0; i < 10; i++) {

if (call[i].phone == number) {

cntCallsNumber++;

startTime = TimeInSeconds(call[i].timeStart);

endTime = TimeInSeconds(call[i].timeEnd);

if (call[i].callType == "vhodyashi") {

callTimeVh += endTime - startTime;

}

if (call[i].callType == "ishodyashi") {

callTimeIs += endTime - startTime;

}

}

}

for (int i = 0; i < 10; i++) {

startTime = TimeInSeconds(call[i].timeStart);

endTime = TimeInSeconds(call[i].timeEnd);

if (mxTimeCall < (endTime - startTime)) {

mxTimeCall = endTime - startTime;

indMx = i;

}

if (mnTimeCall > (endTime - startTime)) {

mnTimeCall = endTime - startTime;

indMn = i;

}

}

while (true) {

std::cout << "\nMeny\n";

std::cout << "1. Find total calls number and time of incoming and outgoing calls\n";

std::cout << "2. Record of the longest call\n";

std::cout << "3. Date and number of the subscriber with whom the shortest conversation took place\n";

std::cout << "4. Leave\n";

std::cout << "Choice action: ";

cin >> choice;

if (choice != 1 && choice != 2 && choice != 3 && choice != 4) {

std::cout << "Incorrect number of action";

break;

}

else if (choice == 1)

std::cout << endl << "Quality calls: " << cntCallsNumber << "; Time incoming call: " << callTimeVh << "; Time outgoing call: " << callTimeIs << endl;

else if (choice == 2)

std::cout << endl << "Record of the longest call" << endl << " " << call[indMx].date << endl << " " << call[indMx].timeStart << endl << " " << call[indMx].timeEnd << endl << " " << call[indMx].phone << endl << " " << call[indMx].callType << endl;

else if (choice == 3)

std::cout << endl << "Date" << endl << " " << call[indMn].date << endl << "Number" << " " << call[indMn].phone << endl;

else if (choice == 4) {

break;

}

else {

std::cout << endl << "Error\n";

break;

}

}

return 0;

}

Тесты

Метод тестирования: белый ящик.

Таблица 2 - Тестирование

|  |  |  |
| --- | --- | --- |
| № | Input | Output |
| 1 | Файл некорректен | Failed to open file |
| 2 | Файл пуст | File is Empty |
| 3 | Входной файл:  2023.10.2608:15:00 08:17:30 ishodyashi | Some fields empty |
| 4 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi | Enter phone number |
| 5 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  1 | The number does not have 11 digits |
| 6 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  A | Invalid characters in the number, the number must consist only of numbers |
| № | Input | Output |
| 7 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  79161234567 | Meny  1. Find total calls number and time of incoming and outgoing calls  2. Record of the longest call  3. Date and number of the subscriber with whom the shortest conversation took place |
| 8 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  79161234567  a | Incorrect number of action |
| 9 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  79161234567  1 | Quality calls: 1; Time incoming call: 0; Time outgoing call: 150 |
| 10 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  79161234567  2 | Record of the longest call  2023.10.26  08:15:00  08:17:30  79161234567  ishodyashi |
|  |  |  |
| 11 | Входной файл:  2023.10.26 08:15:00 08:17:30 79161234567 ishodyashi  Введено с клавиатуры:  79161234567  3 | Date  2023.10.26  Number 79161234567 |

Тестирование программы

|  |  |
| --- | --- |
| № | Результат |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |

Список использованных источников

1. Учебное пособие «Программирование на C++» - В.И. Рейзлин. Издательство томского политехнического университета 2021 года.
2. «Язык программирования C++» Бьярне Страуструп. Издательство Addison-Wesley 2013 года.
3. «C++ Primer» - Стэнли Липпман, Жозе Лажуа, Барбара Му. Издательство Addison-Wesley 2012 года.