1. **Цель работы**

Целью работы является изучение файловых операций ввода-вывода.

1. **Задание**

Согласно варианту №3:

Написать программу, которая считывает текст из файла и записывает в другой файл только те строки, которые содержат двузначные числа.

1. **Описание созданных функций**

I способ

**Имя:** LineSeparation

**Назначение:** Записывает строки с двузначными числами в другой файл

**Входные данные:**

* Fbegin – Файл чтения
* Fansw – Файл записи

**Выходные данные:**

* -

**Побочный эффект:** Запись в файл Fansw

**Тестовые данные:**

|  |  |
| --- | --- |
| Вход | Выход |
| 56 876 987 Пврк 43 | 56 876 Пврк 43 |
| 9 |  |

**Прототип:** void LineSeparation(ifstream& Fbegin, ofstream& Fansw)

**Псевдокод**

Пройтись по каждой строке   
Если есть двузначное число  
 Вписать строку в другой файл  
 Перейти к другой строке

**Блок-схема**

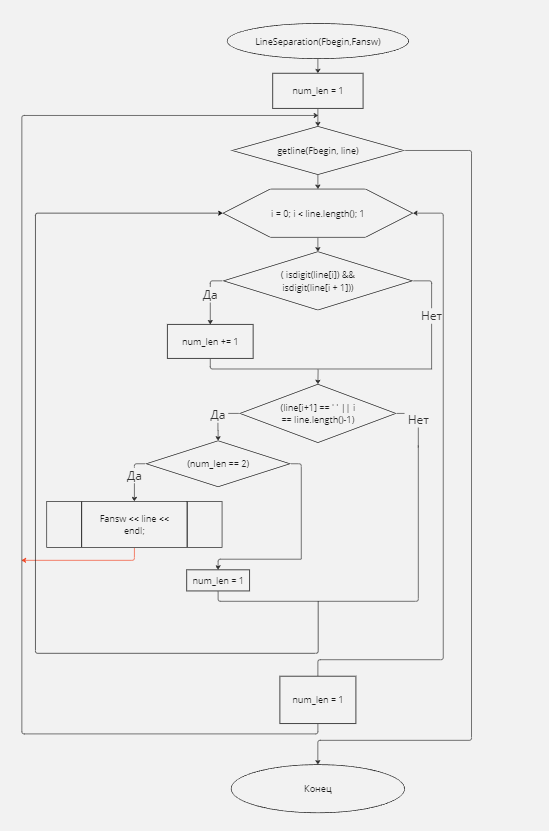


Рисунок 1 – блок-схема LineSeparation

II способ

**Имя:** LineSeparation

**Назначение:** Записывает строки с двузначными числами в другой файл

**Входные данные:**

* Fbegin – Файл чтения
* Fansw – Файл записи

**Выходные данные:**

* -

**Побочный эффект:** Запись в файл Fansw

**Тестовые данные:**

|  |  |
| --- | --- |
| Вход | Выход |
| 56 876 987 Пврк 43 | 56 876 Пврк 43 |
| 9 |  |

**Прототип:** void LineSeparation(FILE \*Fbegin, FILE \*Fansw)

**Псевдокод**

Пройтись по каждой строке   
Если есть двузначное число  
 Вписать строку в другой файл  
 Перейти к другой строке

**Блок-схема**

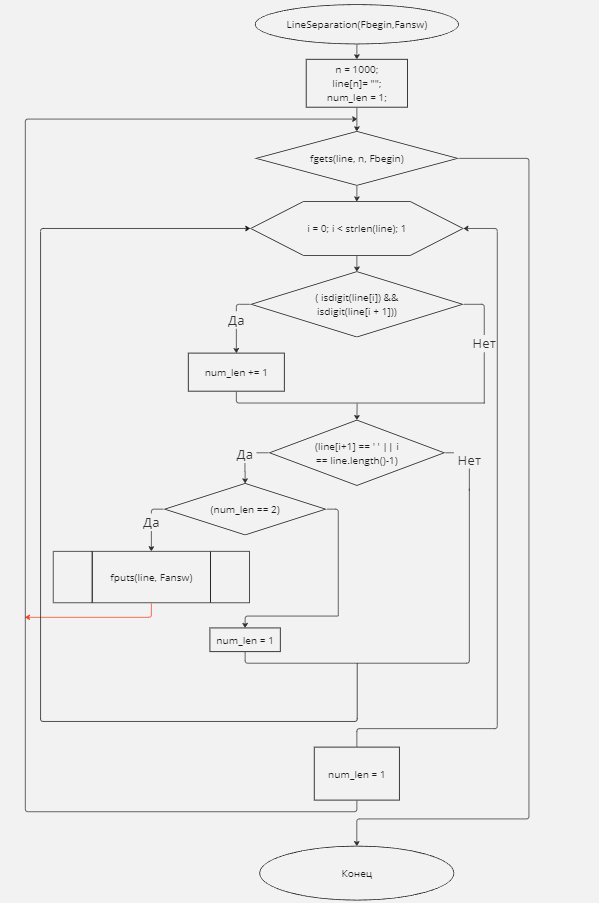


Рисунок 2 – блок-схема LineSeparation

1. **Текст программы**

I Способ

#include <iostream>

#include <Windows.h>

#include <string>

#include <fstream>

using namespace std;

void LineSeparation(ifstream& Fbegin, ofstream& Fansw)

{

string line;

int num\_len = 1;

while (getline(Fbegin, line)) {

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

{

if ( isdigit(line[i]) && isdigit(line[i + 1]))

{

num\_len += 1;

}

if (line[i+1] == ' ' || i == line.length()-1)

{

if (num\_len == 2)

{

Fansw << line << endl;

break;

}

num\_len = 1;

}

}

num\_len = 1;

}

}

int main() {

setlocale(LC\_ALL, "ru");

string Fname, Foutput = "otvet.txt";

ifstream Fbegin;

ofstream Fansw;

cout << "Введите путь и название файла для открытия: ";

cin >> Fname;

Fbegin.open(Fname);

while (!Fbegin.is\_open())

{

cout << "Не удалось открыть файл " << Fname << endl;

cout << "Введите путь и название повторно: ";

cin >> Fname;

Fbegin.open(Fname);

}

Fansw.open(Foutput);

LineSeparation(Fbegin, Fansw);

Fbegin.close();

Fansw.close();

}

II Способ

#include <iostream>

#include <Windows.h>

#include <cstdio>

using namespace std;

void LineSeparation(FILE \*Fbegin, FILE \*Fansw)

{

const int n = 1000;

char line[n]= "";

int num\_len = 1;

while (fgets(line, n, Fbegin)) {

for (int i = 0; i < strlen(line); i++)

{

if (isdigit(line[i]) && isdigit(line[i + 1]))

{

num\_len += 1;

}

if (line[i + 1] == ' ' || i == strlen(line) - 1)

{

if (num\_len == 2)

{

fputs(line, Fansw);

break;

}

num\_len = 1;

}

}

num\_len = 1;

}

}

int main() {

setlocale(LC\_ALL, "ru");

const int n = 1000;

char Fname[n], Foutput[n] = "otvet.txt";

FILE \*Fbegin;

FILE \*Fansw;

cout << "Введите путь и название файла для открытия: ";

cin >> Fname;

errno\_t erro = fopen\_s(&Fbegin, Fname, "r");

while (erro!=0)

{

cout << "Не удалось открыть файл " << Fname << endl;

cout << "Введите путь и название повторно: ";

cin >> Fname;

errno\_t erro = fopen\_s(&Fbegin, Fname, "r");

}

erro = fopen\_s(&Fansw, Foutput, "w");

LineSeparation(Fbegin, Fansw);

fclose(Fbegin);

fclose(Fansw);

}

1. **Пример работы программы**

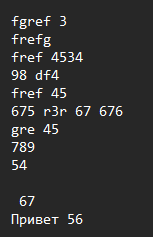


Рисунок 3 – Исходный файл

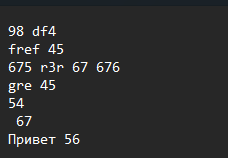


Рисунок 4 – Файл с ответом

Полученные данные совпадают с действительными

1. **Анализ результатов и выводы**

В процессе лабораторной работы были изучены файловые способы обработки ввода-вывода.

Достоинства программы:

* Возможность использования подпрограмм в других разработках
* Проверка на существование файла