СОДЕРЖАНИЕ

1. ЗАДАНИЕ 3

2. ОПИСАНИЕ ПРИМЕНЕНИЯ 3

2.1. Запуск программы 3

2.2. Входные данные 3

2.3. Выходные данные 3

2.4. Сообщения программы 4

3. ОПИСАНИЕ ПРОГРАММЫ 4

3.1. Метод решения задачи 4

3.2. Структура программы 5

3.3. Описание функций 5

3.3.1. main – главная функция 5

3.3.2. fio\_nomer – вывод владельцев автомобилей и номера их автомобилей. 7

3.3.3. kolvo\_marki - вывод списка автомобилей каждой марки

и их количества.  8

ЛИТЕРАТУРА 11

Приложение 1. Текст программы 12

Приложение 2. Пример входного файла vokzal.txt 17

Приложение 3. Результаты тестирования программы 17

**1. ЗАДАНИЕ**

Дан файл, содержащий сведения об отправлении поездов дальнего следования с Казанского вокзала. Структура записи файла: номер поезда, станция назначения, время отправления, время в пути, наличие билетов.

Написать программу, которая позволяет получить:

- информацию о поездах, отправляющихся до указанного времени;

- наличие билетов на поезд с номером ХХХ.

**2. ОПИСАНИЕ ПРИМЕНЕНИЯ**

**2.1 Запуск программы**

Запуск программы осуществляется в среде программирования CodeBlocks.

**2.2 Входные данные**

Входные данные находятся в файле “vokzal.txt”. Число строк в файле - произвольное. Каждая строка содержит номер поезда (4 символа), станция назначения (15 символов), время отправления (5 символов), время в пути (5 символов), наличие билетов (4 символа). Пример строки файла:

094,Красноусольск 0830 0231 230.

| 4 | 15 | 5 | 5 | 4 |

Пример входного файла приведен в приложении 2.

По запросу программы с клавиатуры необходимо ввести номер пункта меню (см. раздел 2.3).

**2.3. Выходные данные**

Программа выводит на экран меню:

===========================================================

**Выберите номер пункта меню:**

**1 - информация о поездах, уходящих до заданного времени**

**2 - количество билетов на заданные поезда**

**3 - выход**

При выборе пункта 1 на экран выводится запрос в виде:

**укажите время отправки в виде ХХХХ:**

После указания времени выводится список в виде:

**№ Станция вр.отправки вр.пути кол-во билетов**

**-------------------------------------------------------------------**

**(номер поезда) (станция отправки) (время отбытия) (время в пути) (кол-во билетов)**

**Вывод завершен**

При отсутствии файла выводится сообщение:

Файл не может быть открыт!

Примеры выходных данных для остальных пунктов меню см. в приложении 3.

**2.4. Сообщения программы**

Ниже приводится перечень возможных сообщений программы:

**1.Файл не может быть открыт!**

**2.Выберите номер пункта меню:**

**3.Нужно вводить номер пункта от 1 до 4**

**4.Для продолжения нажмите любую клавишу**

**5.Файл пуст!**

**6.** **№ Станция вр.отправки вр.пути кол-во билетов**

**7.Вывод завершен**

**8.** **укажите время отправки в виде XXXX:**

**9.** **---------------------------------------------**

**10.** **Введите номер поезда в виде XXX:**

**11.** **Кол-во билетов на заданный поезд =**

**3. ОПИСАНИЕ ПРОГРАММЫ**

**3.1. Метод решения задачи**

Задачу можно разбить на две отдельные подзадачи:

1. Вывод информации о поездах, отправляющихся до заданного времени.

2. Подсчет и вывод количества билетов на поезд с заданным номером.

Каждая подзадача решается методом линейного поиска (последовательного просмотра элементов таблицы).

Чтобы пользователь мог выбирать, какие подзадачи решать и в каком порядке, программа выводит на экран меню (см. раздел 2.3).

**3.2. Структура программы**

bileti\_number

otprvka\_do

main

Рис.1. Функциональная структура программы

Программа состоит из трех функций: главной функции main и двухподпрограмм.

otprvka\_do - выводит информацию о поездах, отправляющихся до заданного времени.

bileti\_number - подсчитывает и выводит количество билетов на поезд с заданным номером.

**3.3. Описание функций**

**3.3.1. main – главная функция**

Заголовок функции:

int main ()

Значение функции:

0 – в случае успешного завершения;

1 – если входной файл не найден.

Рабочие данные:

f – объект класса ifstream, с помощью которого получают доступ к указанному файлу;

n – номер пункта меню.

n

чтение номера пункта меню n

вывод меню

f=NULL

ifstream f(”vokzal.txt”)

да нет

вывод “Файл не может быть открыт!”

otprvka\_do

'1'

bileti\_number

'2'

n!='4'

да

нет

закрытие файла

Рис.1 Блок-схема функции main()

**3.3.2. otprvka\_do - выводит информацию о поездах, отправляющихся до заданного времени.**

Заголовок функции: Структура записи файла:

void otprvka\_do (int kolvozn, vokzal tz[])

poezd – номер поезда station – станция отправки

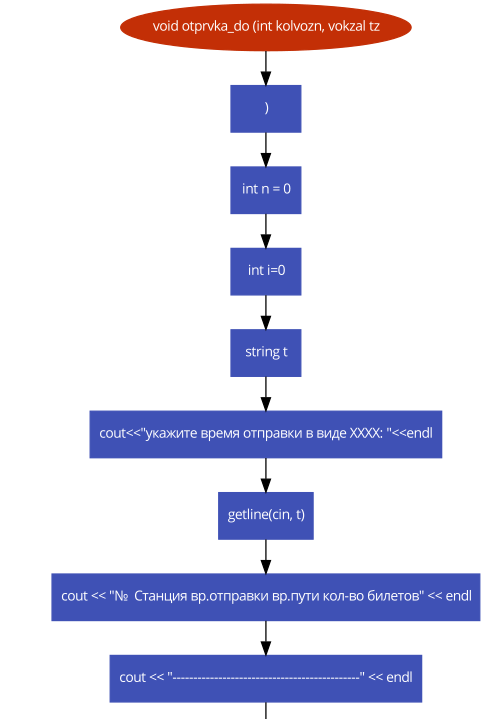
otpravka – время отправки

put – время в пути

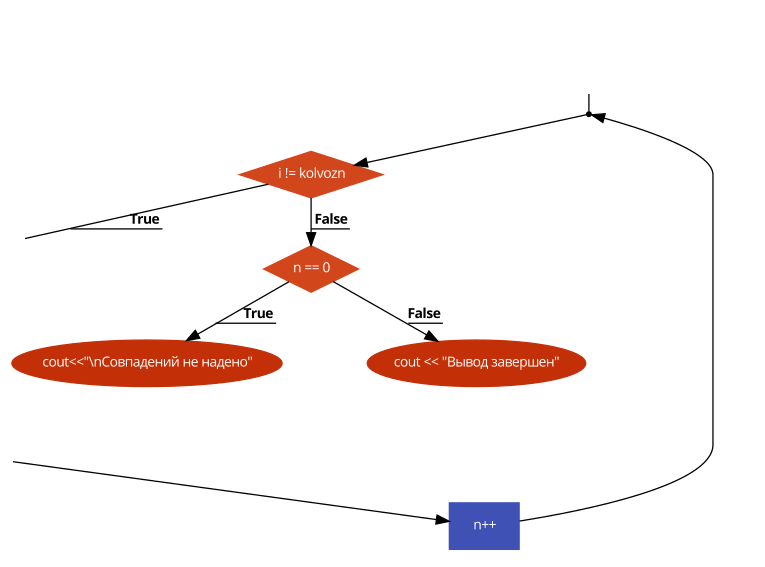
bilet – кол-во оставшихся билетов

Входные данные:

f – объект класса ifstream, с помощью которого получают доступ к указанному файлу;



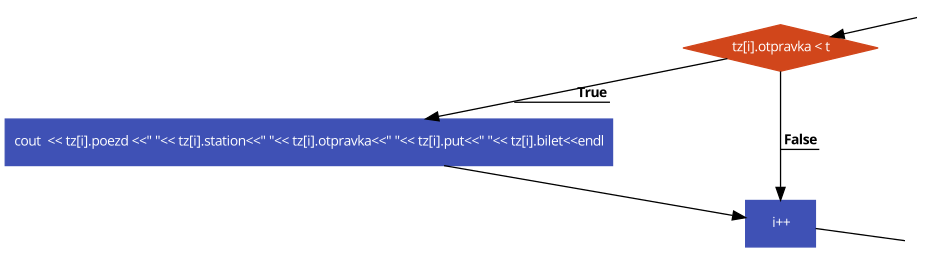
1



2

3

1



3

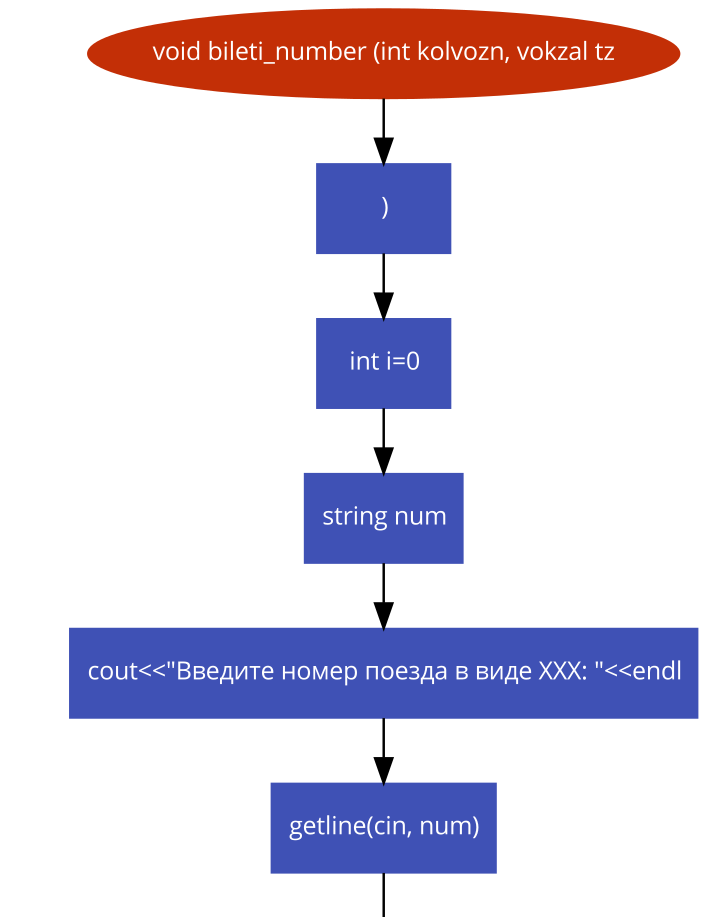
2

Рис.2. Схема функции вывода информации о поездах, отправляющихся до заданного времени.

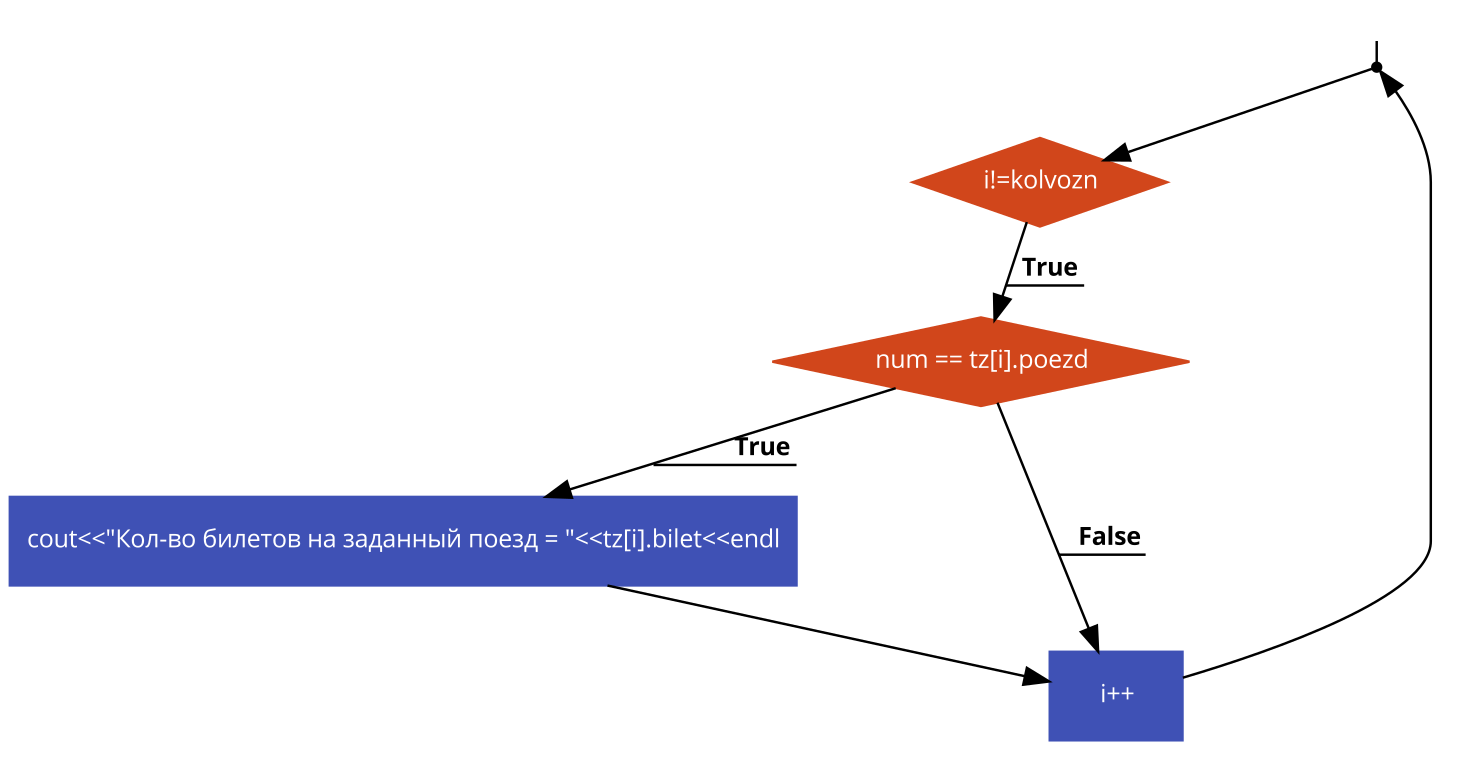
**3.3.3. bileti\_number - подсчитывает и выводит количество билетов на поезд с заданным номером.**

Заголовок функции:

void bileti\_number (int kolvozn, vokzal tz[])



1



1

Рис.3. Блок-схема функции подсчета и вывода количества билетов на поезд с заданным номером.

**Литература.**

1. Задание к курсовой работе.

2. Пример титульного листа.

3. Пример отчета по курсовой работе.

4. Пример кода программы.

**Приложение 1. Текст программы**

#include <fstream>

#include <string>

#include <iostream>

#include <sstream>

#include <windows.h>

using namespace std;

struct vokzal

{

string poezd;

string station;

string otpravka;

string put;

string bilet;

};

void otprvka\_do (int kolvozn, vokzal tz[]);

void bileti\_number (int kolvozn, vokzal tz[]);

int main()

{

ifstream f; /\* ссылка на входной файл \*/

char n; /\* номер пункта меню \*/

int i;

char sl;

string sl1;

SetConsoleCP(1251);// установка кодовой страницы win-cp 1251 в поток ввода

SetConsoleOutputCP(1251); // установка кодовой страницы win-cp 1251 в поток вывода

f.open("vokzal.txt");

if (!f.is\_open())

{

cout << "Файл не может быть открыт!\n";

cin.get();

return 1;

}

else if (!f.eof())

{

vokzal tz[88];

for (i = 0; f.good();i++)

{

if ((sl=f.get())=='\n'&&i!=0)

getline(f, tz[i].poezd,',');

else

{

tz[i].poezd += sl;

getline(f, sl1, ',');

tz[i].poezd += sl1;

}

f >> tz[i].station>>tz[i].otpravka>>tz[i].put;

if ((sl = f.get()) == ' ')

getline(f, tz[i].bilet,'.');

cout << tz[i].poezd<<" "<< tz[i].station <<" " << tz[i].otpravka<<" "<<tz[i].put<<" "<<tz[i].bilet<<" "<<endl;

}

int kolvozn = i - 1;

do

{

cout << "========================================================"<<endl;

cout << "Выберите номер пункта меню:"<<endl;

cout << "1 - информация о поездах, уходящих до заданного времени" << endl;

cout << "2 - количество билетов на заданные поезда"<<endl;

cout << "3 - выход"<<endl;

cout << "------------------------------------------------------"<<endl;

n=cin.get();

cin.sync();

switch (n)

{

case '1': otprvka\_do(kolvozn,tz); break;

case '2': bileti\_number(kolvozn,tz); break;

case '3': break;

default: cout << "\nНужно вводить номер пункта от 1 до 4";

}

if (n != '3')

{

cout << "\nДля продолжения нажмите любую клавишу";

cin.get();

}

} while (n != '3');

f.close();

}

else

cout << "Файл пуст!\n";

//cin.get();

return 0;

}

void otprvka\_do (int kolvozn, vokzal tz[])

{

int n = 0;

int i=0;

string t;

cout<<"укажите время отправки в виде XXXX: "<<endl;

getline(cin, t);

cout << "№ Станция вр.отправки вр.пути кол-во билетов" << endl;

cout << "---------------------------------------------" << endl;

while (i != kolvozn)

{

if (tz[i].otpravka < t)

cout << ++n << tz[i].poezd <<" "<< tz[i].station<<" "<< tz[i].otpravka<<" "<< tz[i].put<<" "<< tz[i].bilet<<endl;

i++;

}

if (n == 0) cout<<"\nСовпадений не надено";

else cout << "Вывод завершен";

}

void bileti\_number (int kolvozn, vokzal tz[])

{

int i=0, k=0;

string num;

cout<<"Введите номер поезда в виде XXX: "<<endl;

getline(cin, num);

while (i!=kolvozn)

{

if(num == tz[i].poezd)

{

cout<<"Кол-во билетов на заданный поезд = "<<tz[i].bilet<<endl;

k++;

}

i++;

}

if (k==0)

cout<<"Поезд с таким номером не найден!"<<endl;

}

**Приложение 2. Пример входного файла vokzal.txt**

094,Красноусольск 0830 0231 230.

102,Камышова 1015 0245 12.

131,Аметьева 1136 0035 0.

142,Красная поляна 2034 0404 137.

169,Комарова 1805 0307 0.

**Приложение 3. Результаты тестирования программы**

**Тест 1.** Входного файла нет в текущем каталоге.

Результат:

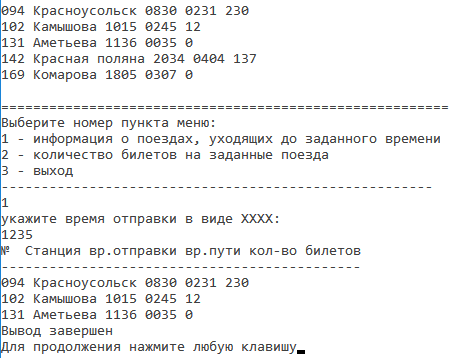


В следующих тестах используется файл из приложения 2.

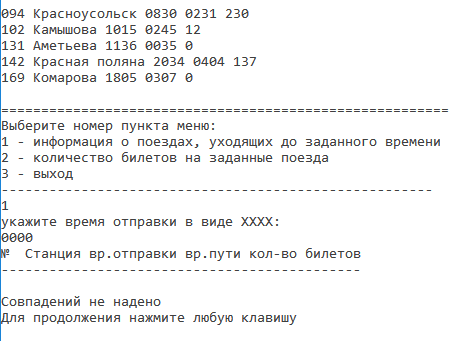
**Тест 2.**

****

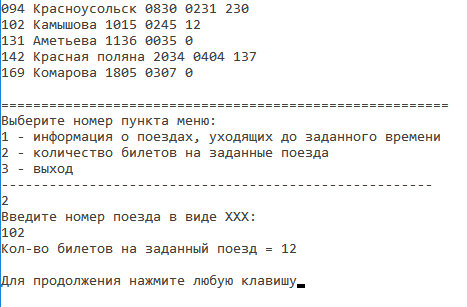
**Тест 3.**

****

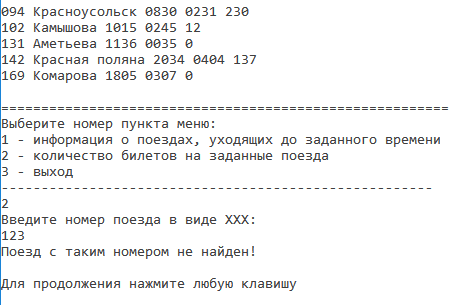
**Тест 4.**

****

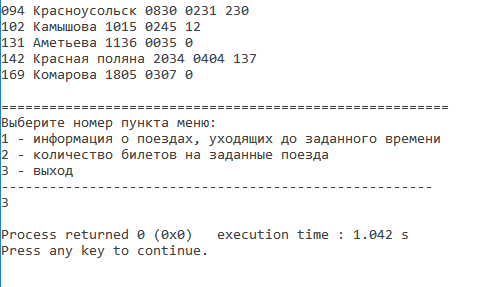
**Текст 5.**

****

**Тест 6.**

****

**Тест 7.**

****