**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5.  
Курс «​Программирование на основе классов и шаблонов​»

Отчет по домашней работе.

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-21Б |  | преподаватель каф. ИУ5 |
| Карпов Даниил |  |  |
| Подпись и дата: |  | Подпись и дата: |

г. Москва, 2019 г.

**Задание.**

Написать программу моделирования работы автобусного парка.

Сведения о каждом автобусе содержат: номер автобуса, фамилию и инициалы водителя, номер маршрута. База хранится в текстовом файле, его размер может быть произвольным.

Программа должна обеспечивать выбор с помощью меню и выполнение следующих функций:

начальное формирование данных о всех автобусах в парке в виде списка;

имитация выезда автобуса из парка: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;

имитация въезда автобуса в парк: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;

вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте.

Для представления необходимых списков использовать класс list.

**Разработка интерфейса класса:**

**Bus.h**

#pragma once

#include <string>

using namespace std;

class Bus

{

private:

int number;

string surname, name;

int route;

bool condition;

public:

Bus();

Bus(string s);

bool check(int n);

void transform(int i);

friend ostream& operator << (ostream& out, Bus point);

};

**Bus.cpp**

#include "Bus.h"

#include <iostream>

Bus::Bus()

{

number = 69;

surname = "P";

name = "N";

route = 69;

condition = false;

}

Bus::Bus(string s)

{

surname = s.substr(s.find(' ') + 1, s.find(' ', s.find(' ') + 1));

// находим первый и второй пробел и все между ними в фамилию

name = s.substr(s.find(' ', s.find(' ') + 1) + 1, s.rfind(' '));

number = 0;

route = 0;

for (int i = 0; i < s.find(' '); i++)

{

number = 10 \* number + s[i] - '0';

}

for (int i = s.rfind(' ') + 1; i < s.size(); i++)

{

route = 10 \* route + s[i] - '0';

}

condition = false;

}

bool Bus::check(int n)

{

if (n != 0)

return number == n;

return condition;

}

void Bus::transform(int i)

{

if (i == 0)

condition = true;

else

condition = false;

}

ostream& operator << (ostream& out, Bus point)

{

cout << point.number << ' ' << point.surname << ' ' << point.name << ' ' << point.condition << endl;

return out;

}

**Menu.cpp**

#include "Bus.h"

#include <iostream>

#include <list>

#include <fstream>

using namespace std;

void menu(list<Bus>& array)

{

setlocale(LC\_ALL, "RUS");

cout << "1 – считываем текстовый файл\n"

<< "2 – выход на маршрут\n"

<< "3 – возвращение блудного сына\n"

<< "4 – те, кто там\n"

<< "5 – те, кто здесь\n"

<< "НУЖНО: ";

int choice = 0, n;

string s;

ifstream fin;

while (choice < 1 || choice > 5)

cin >> choice;

switch (choice)

{

default:

case 1:

cin >> s;

fin.open(s);

while (!fin.eof())

{

getline(fin, s);

array.push\_back(Bus(s));

}

break;

case 2:

cin >> n;

for (auto i : array)

{

if (i.check(n))

{

i.transform(0);

}

}

break;

case 3:

cin >> n;

for (auto i : array)

{

if (i.check(n))

{

i.transform(1);

}

}

break;

case 4:

for (auto i : array)

if (i.check(0))

cout << i;

break;

case 5:

for (auto i : array)

if (!i.check(0))

cout << i;

break;

}

}

**Main.cpp**

#include "Bus.h"

#include <iostream>

#include <list>

using namespace std;

void menu(list<Bus>& array);

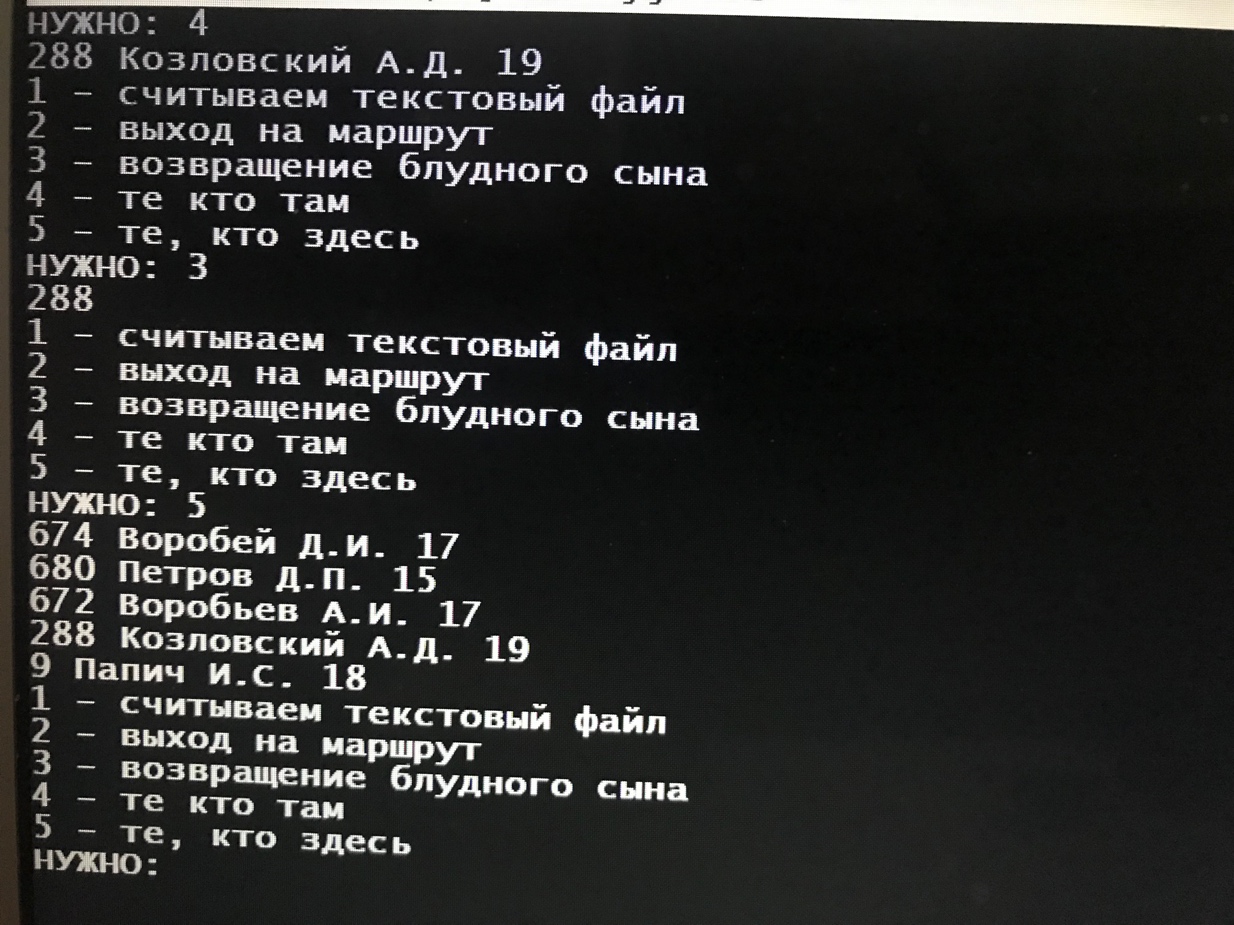
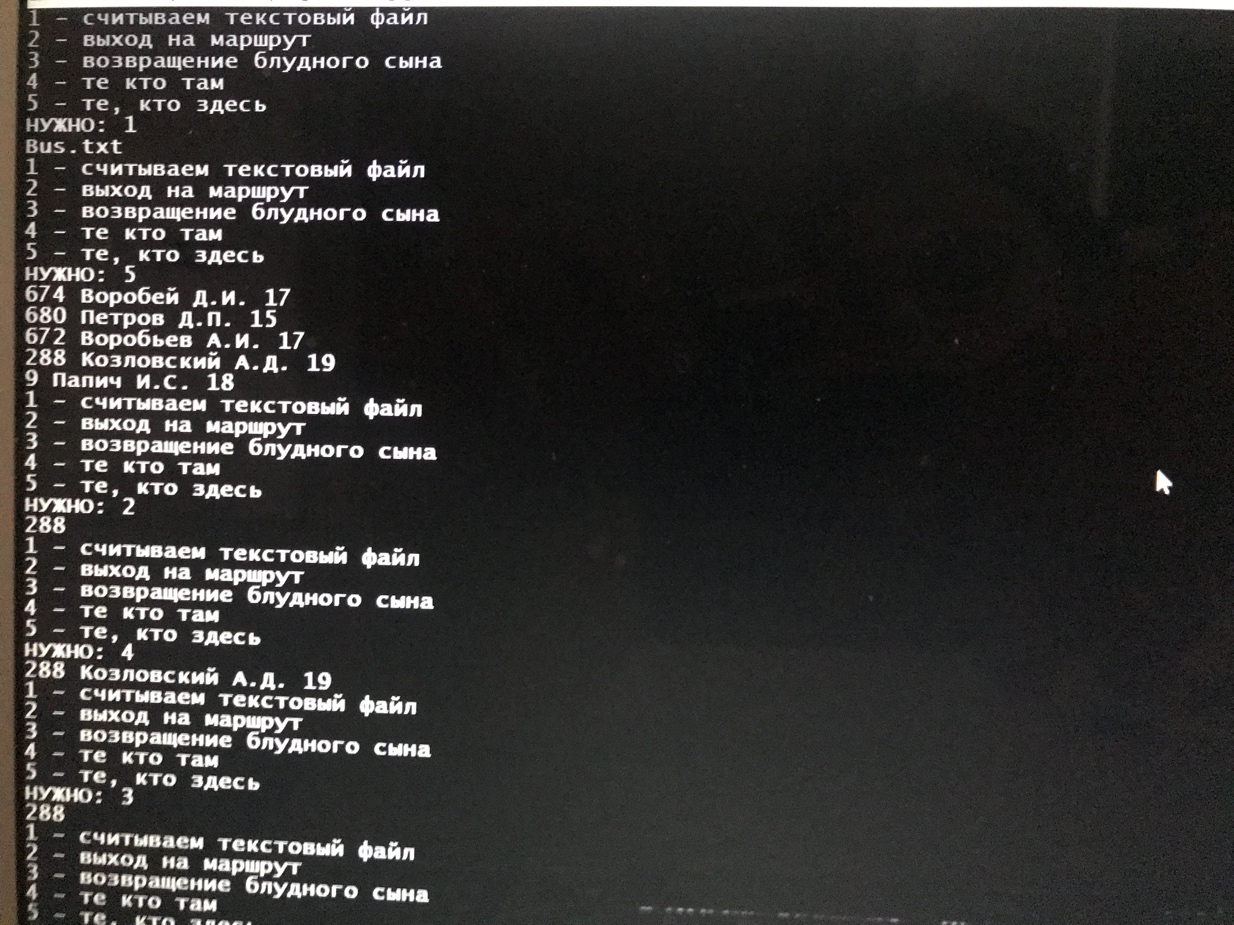
int main()

{

list <Bus> array;

while (1) { menu(array); }

}

**Вывод данных:**