МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«РОССИЙСКИЙ БИОТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ (РОСБИОТЕХ)»

Институт промышленной инженерии, информационных технологий и мехатроники

Кафедра «Информатика и вычислительная техника пищевых производств»

Направление: 09.03.01 Информатика и вычислительная техника

**ЛАБОРАТОРНАЯ РАБОТА №**2

*на тему:*

«ПРОСТЕЙШИЕ КЛАССЫ И ОБЪЕКТЫ В С++»

Вариант № 14

|  |  |  |
| --- | --- | --- |
| Выполнил:  ФИО |  | Студент 1 курса, гр. 24o-090301/БА-2  Орлов Григорий Артемович |
| Проверил: |  | Ящун Т.В. |

Москва, 2025

Цель работы: Научиться работать с простейшими классами и объектами в C++.

**Задания**

1. Создать класс Phone: Фамилия, Имя, Отчество, Адрес, Номер, Время внутригородских разговоров, Время междугородних разговоров.

2. Создать динамический массив (вектор) объектов класса.

3. Вывести:

а) сведения об абонентах, время внутригородских разговоров которых превышает заданное;

б) сведения об абонентах, воспользовавшихся междугородней связью;

в) сведения об абонентах, выведенные в алфавитном порядке.

**Решение**

1. Файл Orlov\_LR2\_Phone.h
2. #ifndef PHONE\_H
3. #define PHONE\_H
4. #include <string>
5. #include <vector>
6. using namespace std;
7. extern const string DATA\_FILE;
8. class Phone {
9. private:
10. string lastName;
11. string firstName;
12. string middleName;
13. string address;
14. string phoneNumber;
15. int vgorodCallTime;
16. int mgorodCallTime;
17. public:
18. Phone();
19. ~Phone();
21. string getLastName() const { return lastName; }
22. string getFirstName() const { return firstName; }
23. string getMiddleName() const { return middleName; }
24. string getAddress() const { return address; }
25. string getPhoneNumber() const { return phoneNumber; }
26. int getLocalCallTime() const { return vgorodCallTime; }
27. int getInternationalCallTime() const { return mgorodCallTime; }
29. void setLastName(const string &ln) { lastName = ln; }
30. void setFirstName(const string &fn) { firstName = fn; }
31. void setMiddleName(const string &mn) { middleName = mn; }
32. void setAddress(const string &addr) { address = addr; }
33. void setPhoneNumber(const string &num) { phoneNumber = num; }
34. void setLocalCallTime(int time) { vgorodCallTime = time; }
35. void setInternationalCallTime(int time) { mgorodCallTime = time; }
37. void display() const;
39. static Phone\* inputFromKeyboard();
41. bool exceedsLocalCallTime(int porog) const;
42. bool usedInternationalCalls() const;
44. bool operator<(const Phone &other) const;
45. };
46. void loadDataFromFile(const string &filename, vector<Phone> &phones);
47. void addPhoneToVector(vector<Phone> &phones);
48. void displayAllPhones(const vector<Phone> &phones);
49. void displayPhonesExceedingLocalTime(const vector<Phone> &phones, int porog);
50. void displayPhonesWithInternationalCalls(const vector<Phone> &phones);
51. void displayPhonesInAlphabeticalOrder(vector<Phone> &phones);
52. #endif

2. Файл Orlov\_LR2\_Phone.cpp

#include "Orlov\_LR2\_Phone.h"

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <limits>

using namespace std;

const string DATA\_FILE = "phones\_data.txt";

vector<Phone> phoneDatabase;

Phone::Phone() : lastName(""), firstName(""), middleName(""), address(""),

                phoneNumber(""), vgorodCallTime(0), mgorodCallTime(0) {}

Phone::~Phone() {}

void Phone::display() const {

    cout << "ФИО: " << lastName << " " << firstName << " " << middleName << "\n"

         << "Адрес: " << address << "\n"

         << "Номер телефона: " << phoneNumber << "\n"

         << "Время внутригородских разговоров: " << vgorodCallTime << " минут\n"

         << "Время междугородних разговоров: " << mgorodCallTime << " минут\n"

         << "------------------------------------\n";

}

Phone\* Phone::inputFromKeyboard() {

    Phone\* newPhone = new Phone();

    string input;

    int time;

    cout << "Введите фамилию: ";

    getline(cin, input);

    newPhone->setLastName(input);

    cout << "Введите имя: ";

    getline(cin, input);

    newPhone->setFirstName(input);

    cout << "Введите отчество: ";

    getline(cin, input);

    newPhone->setMiddleName(input);

    cout << "Введите адрес: ";

    getline(cin, input);

    newPhone->setAddress(input);

    cout << "Введите номер телефона: ";

    getline(cin, input);

    newPhone->setPhoneNumber(input);

    cout << "Введите время внутригородских разговоров (в минутах): ";

    cin >> time;

    newPhone->setLocalCallTime(time);

    cout << "Введите время междугородних разговоров (в минутах): ";

    cin >> time;

    newPhone->setInternationalCallTime(time);

    cin.ignore(numeric\_limits<streamsize>::max(), '\n');

    return newPhone;

}

bool Phone::exceedsLocalCallTime(int porog) const {

    return vgorodCallTime > porog;

}

bool Phone::usedInternationalCalls() const {

    return mgorodCallTime > 0;

}

bool Phone::operator<(const Phone &other) const {

    if (lastName != other.lastName) return lastName < other.lastName;

    if (firstName != other.firstName) return firstName < other.firstName;

    return middleName < other.middleName;

}

bool isValidName(const string &name) {

    return !name.empty() && name.find\_first\_not\_of("абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ- ") == string::npos;

}

bool isValidPhoneNumber(const string &number) {

    return !number.empty() && number.find\_first\_not\_of("0123456789-+() ") == string::npos;

}

bool isValidTime(int time) {

    return time >= 0;

}

string inputString(const string &prompt, bool (\*validator)(const string&)) {

    string input;

    do {

        cout << prompt;

        getline(cin, input);

        if (!validator(input)) {

            cout << "Ошибочный ввод. Попробуйте снова.\n";

        }

    } while (!validator(input));

    return input;

}

int inputInt(const string &prompt, bool (\*validator)(int)) {

    int value;

    do {

        cout << prompt;

        cin >> value;

        if (!validator(value)) {

            cout << "Ошибочный ввод. Попробуйте снова.\n";

        }

        cin.ignore(numeric\_limits<streamsize>::max(), '\n');

    } while (!validator(value));

    return value;

}

void loadDataFromFile(const string &filename, vector<Phone> &phones) {

    ifstream file("phones\_data.txt");

    if (!file.is\_open()) {

        cerr << "Ошибка запуска файла: " << "phones\_data.txt" << endl;

        return;

    }

    Phone phone;

    string line;

    while (getline(file, line)) {

        phone.setLastName(line);

        getline(file, line);

        phone.setFirstName(line);

        getline(file, line);

        phone.setMiddleName(line);

        getline(file, line);

        phone.setAddress(line);

        getline(file, line);

        phone.setPhoneNumber(line);

        int time;

        file >> time;

        phone.setLocalCallTime(time);

        file >> time;

        phone.setInternationalCallTime(time);

        file.ignore(numeric\_limits<streamsize>::max(), '\n');

        phones.push\_back(phone);

    }

    file.close();

}

void addPhoneToVector(vector<Phone> &phones) {

    Phone\* newPhone = Phone::inputFromKeyboard();

    phones.push\_back(\*newPhone);

    delete newPhone;

}

void displayAllPhones(const vector<Phone> &phones) {

    if (phones.empty()) {

        cout << "Нет телефона в базе данных.\n";

        return;

    }

    for (const auto &phone : phones) {

        phone.display();

    }

}

void displayPhonesExceedingLocalTime(const vector<Phone> &phones, int porog) {

    bool found = false;

    for (const auto &phone : phones) {

        if (phone.exceedsLocalCallTime(porog)) {

            phone.display();

            found = true;

        }

    }

    if (!found) {

        cout << "Нет номеров телефонов, время внутригородских разговоров которых превышает лимит в " << porog << " минут(ы).\n";

    }

}

void displayPhonesWithInternationalCalls(const vector<Phone> &phones) {

    bool found = false;

    for (const auto &phone : phones) {

        if (phone.usedInternationalCalls()) {

            phone.display();

            found = true;

        }

    }

    if (!found) {

        cout << "Нет номеров, использовавших время междугородных разговоров.\n";

    }

}

void displayPhonesInAlphabeticalOrder(vector<Phone> &phones) {

    if (phones.empty()) {

        cout << "Нет телефона в базе данных.\n";

        return;

    }

    sort(phones.begin(), phones.end());

    displayAllPhones(phones);

}

3. Файл Orlov\_LR2\_MAin.cpp

#include "Orlov\_LR2\_Phone.h"

#include "Orlov\_LR2\_Phone.cpp"

#include <iostream>

#include <map>

#include <functional>

#include <limits>

using namespace std;

void showMenu() {

    cout << "\nМеню базы данных Phone:\n";

    cout << "1. Загрузить данные из файла\n";

    cout << "2. Добавить новую запись\n";

    cout << "3. Показать все записи\n";

    cout << "4. Показать записи, где был превышен порог времени внутригородских разговоров\n";

    cout << "5. Показать записи, где использовалось время междугородних разговоров\n";

    cout << "6. Показать записи в алфавитном порядке\n";

    cout << "7. Выход\n";

    cout << "Введите свой выбор: ";

}

int main() {

    vector<Phone> phones;

    map<int, function<void()>> menuActions;

    menuActions[1] = [&]() {

        loadDataFromFile(DATA\_FILE, phones);

        cout << "Данные загружены из файла.\n";

    };

    menuActions[2] = [&]() {

        addPhoneToVector(phones);

        cout << "Новая запись добавлена.\n";

    };

    menuActions[3] = [&]() {

        cout << "\nВсе записи:\n";

        displayAllPhones(phones);

    };

    menuActions[4] = [&]() {

        int porog;

        cout << "Введите порог времени внутригородских разговоров (в минутах): ";

        cin >> porog;

        cin.ignore(numeric\_limits<streamsize>::max(), '\n');

        cout << "\nЗаписи, где время внутригородиских разговоров превысело " << porog << " минут(ы):\n";

        displayPhonesExceedingLocalTime(phones, porog);

    };

    menuActions[5] = [&]() {

        cout << "\nЗаписи, где использовалось время междугородних разговоров:\n";

        displayPhonesWithInternationalCalls(phones);

    };

    menuActions[6] = [&]() {

        cout << "\nЗаписи в алфавитном порядке:\n";

        displayPhonesInAlphabeticalOrder(phones);

    };

    menuActions[7] = []() {

        cout << "Выход из программы.\n";

        exit(0);

    };

    while (true) {

        showMenu();

        int choice;

        cin >> choice;

        cin.ignore(numeric\_limits<streamsize>::max(), '\n');

        try {

            auto action = menuActions.at(choice);

            action();

        } catch (const out\_of\_range&) {

            cout << "Ошибочный выбор. Попробуйте вновь.\n";

        }

    }

    return 0;

}

4. Файл phones\_data.txt

Иванов

Сергей

Петрович

Тверская улица, 15

+79123456789

45

10

Смирнов

Дмитрий

Александрович

Арбат, 23

+79876543210

30

0

Кузнецов

Анлрей

Владимирович

Ленинский проспект, 42

+79261234567

60

15

Попов

Алексей

Игоревич

Кутузовский проспект, 10

+79031112233

25

5

Васильев

Михаил

Сергеевич

Новый Арбат, 8

+79655554466

50

0

Соколова

Анна

Дмитриевна

Садовое кольцо, 34

+74951234567

40

8

Морозова

Екатерина

Павловна

Проспект Мира, 55

+74999876543

35

12

Волкова

Ольга

Викторовна

Улица Покровка, 12

+74955551234

20

0

Лебедева

Мария

Андреевна

Рублевское шоссе, 18

+74996789012

55

7

Козлова

Наталья

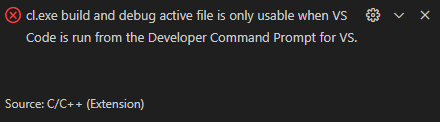
Игоревна

Улица Вавилова, 7

+74957890123

15

3

Вывод: Не получается посмотреть вывод программы, так как выдает ошибку, решить которую я не в силах

После одного Debug’а всё сломалось и не получается запустить.