МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**“Сибирский государственный университет науки и технологий**

**Имени академика М. Ф. Решетнева”**

**Лабораторная работа №1**  
**По дисциплине “Программирование”**

**Работу выполнил:**  
Козловский Данил Иванович.  
**Работу проверил:**  
Доцент каф.ИВТ Богданов К.В

**Красноярск 2019**

**Вариант 15**

**Задача:**

Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

**Программа:**

#include <iostream>

**using** std::string;

**class** Container{

**public**:

**class** NumberBook{

**public**:

string name, lastName;

**int** number;

string birthDay;

NumberBook(string name, string lastName, **int** number, string birthDay){

setName(name);

setLastName(lastName);

setNumber(number);

setBirthDay(birthDay);

}

NumberBook(){}

**void** setName(string name){

**this**->name = name;

}

**void** setLastName(string lastName){

**this**->lastName = lastName;

}

**void** setNumber(**int** number) {

**this**->number = number;

}

**void** setBirthDay(string birthDay){

**this**->birthDay = birthDay;

}

string getName(){

**return** name;

}

string getLastName(){

**return** lastName;

}

**int** getNumber() {

**return** number;

}

string getBirthDay(){

**return** birthDay;

}

};

**private**:

NumberBook nb[100000];

**public**:

NumberBook\* getNB(){

**return** nb;

}

**void** sort(**int** realSize, **int** s){

**for** (**int** startIndex = 0; startIndex < realSize - 1; ++startIndex){

**int** smallestIndex = startIndex;

**for** (**int** currentIndex = startIndex + 1; currentIndex < realSize; ++currentIndex){

**if**(s == 1)

**if** (nb[currentIndex].getName() < nb[smallestIndex].getName())

smallestIndex = currentIndex;

**if**(s == 2)

**if** (nb[currentIndex].getLastName() < nb[smallestIndex].getLastName())

smallestIndex = currentIndex;

**if**(s == 3)

**if** (nb[currentIndex].getNumber() < nb[smallestIndex].getNumber())

smallestIndex = currentIndex;

**if**(s == 4)

**if** (nb[currentIndex].getBirthDay() < nb[smallestIndex].getBirthDay())

smallestIndex = currentIndex;

}

std::swap(nb[startIndex], nb[smallestIndex]);

}

}

**void** search(**int** realSize, **int** s){

**int** i = 0;

**if**(s == 1){

string name;

std::cout << "Введите имя" << std::endl;

std::cin >> name;

**for**(i = 0; i < realSize; i++){

**if**(nb[i].getName() == name)

**break**;

}

}

**if**(s == 2){

string lastName;

std::cout << "Введите фамилию" << std::endl;

std::cin >> lastName;

**for**(i = 0; i < realSize; i++){

**if**(nb[i].getLastName() == lastName)

**break**;

}

}

**if**(s == 3){

**int** number;

std::cout << "Введите номер" << std::endl;

std::cin >> number;

**for**(i = 0; i < realSize; i++){

**if**(nb[i].getNumber() == number)

**break**;

}

}

**if**(s == 4){

string birthDay;

std::cout << "Введите дату рождения (дд.мм.гггг)" << std::endl;

std::cin >> birthDay;

**for**(i = 0; i < realSize; i++){

**if**(nb[i].getBirthDay() == birthDay)

**break**;

}

}

**if**(i == realSize)

std::cout << "В записной книжке нет такой записи!" << std::endl;

**else**{

std::cout << "Имя: Фамилия: Номер телефона: Дата рождения:" << std::endl;

std::cout << nb[i].getName() << " " << nb[i].getLastName() << " " << nb[i].getNumber() << " " << nb[i].getBirthDay() << std::endl;

}

}

};

**int** main() {

Container cont;

Container::NumberBook\* nb = cont.getNB();

**int** realSize = 0;

**int** k = 10;

**while**(k != 0){

std::cout<< "Меню:\n"

"0. Закрыть программу\n"

"1. Добавить значение\n"

"2. Отобразить записную книжку\n"

"3. Удалить значение\n"

"4. Отсортировать по полю\n"

"5. Поиск\n";

std::cin>>k;

**switch**(k){

**case** 0:

exit(0);

**case** 2:

std::cout << "Имя: Фамилия: Номер телефона: Дата рождения:" << std::endl;

**for**(**int** i = 0; i < realSize; i++){

std::cout << nb[i].getName() << " " << nb[i].getLastName() << " " << nb[i].getNumber() << " " << nb[i].getBirthDay() << std::endl;

}

**break**;

**case** 3:

**int** num;

std::cout << "Введите номер телефона:" << std::endl;

std::cin >> num;

**for**(**int** i = 0; i < realSize; i++){

**if**(nb[i].getNumber() == num){

**for**(**int** j = i; j < realSize; j++){

**if**(j != realSize)

nb[j] = nb[j + 1];

}

realSize--;

}

}

**break**;

**case** 4:

std::cout << "Выберите поле сортировки: 1-имя 2-фамилия 3-номер телефона 4-дата рождения" << std::endl;

**int** s;

std::cin >> s;

cont.sort(realSize, s);

std::cout << "Сортировка завершена" << std::endl;

**break**;

**case** 5:

std::cout << "Выберите поле для поиска: 1-имя 2-фамилия 3-номер телефона 4-дата рождения" << std::endl;

**int** b;

std::cin >> b;

cont.search(realSize, b);

std::cout << "Поиск завершен!" << std::endl;

**break**;

**case** 1:

string name, lastName;

**int** number;

string birthDay;

std::cout<<"Введите: Имя Фамилию Номер Дату рождения(ДД.ММ.ГГГГ)"<<std::endl;

std::cin>>name>>lastName>>number>>birthDay;

std::cout<<std::endl;

nb[realSize++] = Container::NumberBook(name,lastName,number,birthDay);

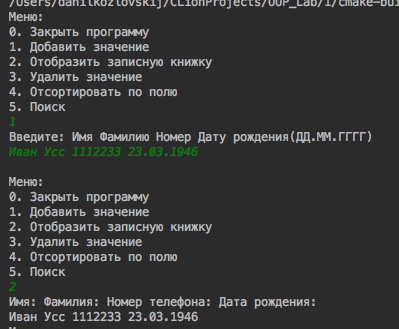
**break**;

}

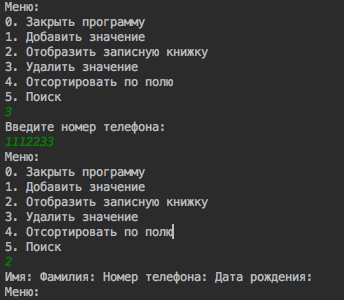
}

}

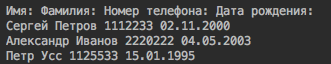
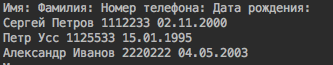
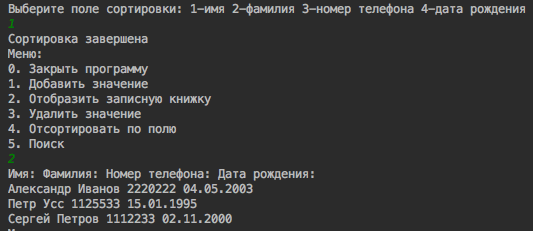
**Тестирование:**

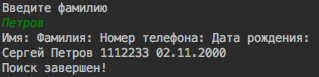
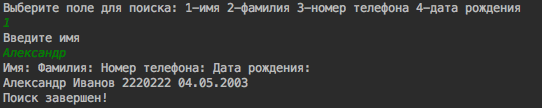
1. **Добавление и вывод записей**

Как видно из скриншота программа работает корректно, так как запись была успешно добавлена и выведена.

1. **Удаление записей**

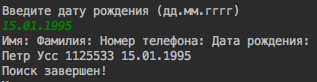
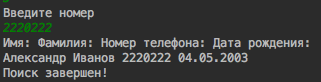
Как видно из скриншота строка удалилась успешно, а значит программа работает правильно.

1. **Сортировка записейСнимок экрана 2019-02-27 в 19.38.31.png**

Соответсвенно скриншоты расставлены в порядке номеров сортировки в меню.

Как видно из скриншотов сортировка по всем полям происходит корректно, а значит программа работает правильно

1. **Поиск по записям**

Приведены примеры поиска разных людей из списка по разным параметрам, а так-же вариант ввода неизвестного списку Снимок экрана 2019-02-27 в 19.43.41.pngзначения.

Как видно из скриншотов поиск по всем полям происходит успешно, а значит программа работаете правильно.