Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «вычислительная техника»

Лабораторная работа №11

«Списки»

Выполнил студент

Группы ИВТАСбд-11

Зюзин Г.А.

Ульяновск, 2022

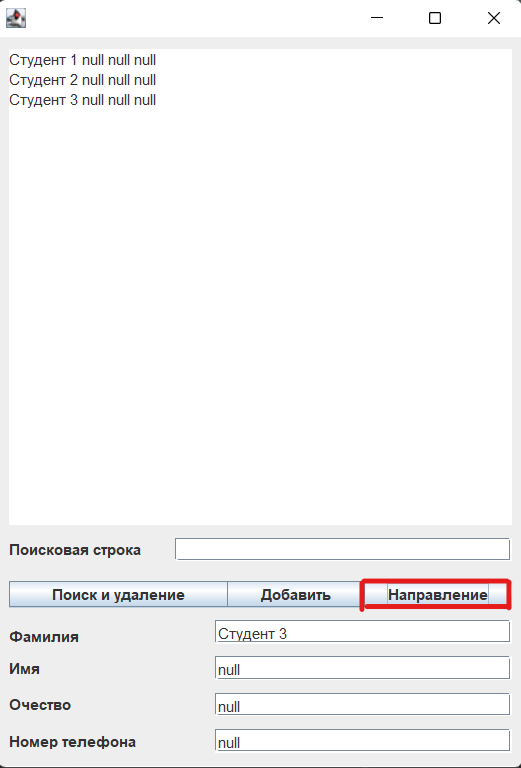
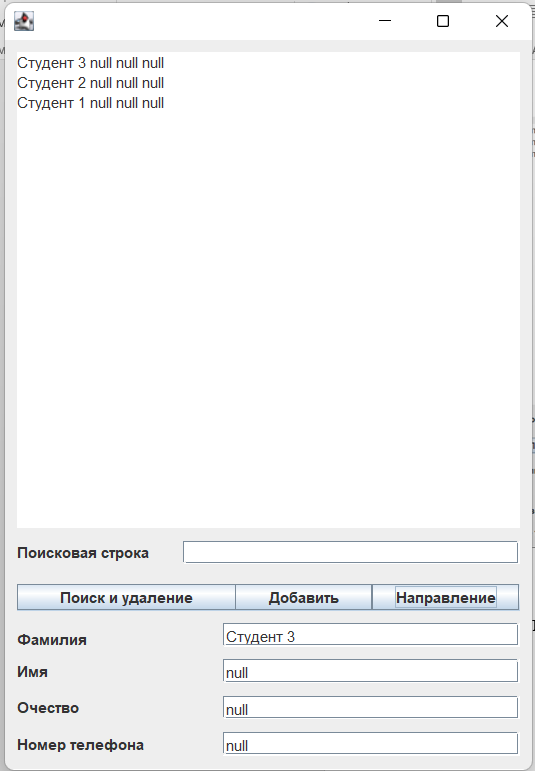
1. **Постановка задачи**

Требуется написать программу, позволяющую выполнять следующие действия над списком: просмотр списка в прямом и обратном направлении, добавление элемента, удаление элемента по значению.

Для программы реализовать графический пользовательский интерфейс.  
В данном варианте необходимо работать с двухсвязным списком фамилий, имен, отчеств и номеров телефонов. Отображение списка осуществлять в прямом или обратном направлении по выбору пользователя. Поиск осуществлять по фамилии, при удалении удалять только первый встретившийся элемент.

1. **Особенности реализации**

*Просмотр списка в прямом и в обратном направление.* Данная

возможность реализована в графическом интерфейсе следующим образом:

По нажатию кнопки «Направление» список начинает прочитываться в противоположном направлении. Направление определяется флаговой переменной и список перерисовывается:

StringBuilder sb = new StringBuilder();

if(flagDirection) {

Element search = list.tail;

while(search != null) {

sb.append(search.nameF).append(' ').append(search.nameI).append(' ').append(search.nameO).append(' ').append(search.number).append('\n');

search = search.prev;

}

}else {

Element search = list.head;

while(search != null) {

sb.append(search.nameF).append(' ').append(search.nameI).append(' ').append(search.nameO).append(' ').append(search.number).append('\n');

search = search.next;

}

}

textArea.setText(sb.toString());

flagDirection = !flagDirection;

}

Для работы со списком были написаны следующие функции: addHead(String nameF, String nameI, String nameO, String number), addTail(<...>), deleteHead(), deleteTail(), deleteElementByNameF(<…>).

Функции add добавляют элементы в список в двух разных направления в зависимости от того, по какому направлению захочет работать пользователь.

public void addHead(String nameF, String nameI, String nameO, String number) {

Element element = new Element(nameF, nameI, nameO, number);

if(this.checkList()) {

tail = element;

} else {

head.prev = element; //Предыдущая голова

}

element.next = head;

head = element;

}

Функции delete удаляют элементы из списка в двух разных направления в зависимости от того, по какому направлению захочет работать пользователь.

public void deleteTail() {

if(tail.prev == null) {

head = null;

} else {

tail.prev.next = null;

}

}

И функция, которую необходимо реализовать для варианта – deleteElementByNameF. Это функция, которая будет удалять элемент из списка по первой фамилии.

public void deleteElementByNameF(String lastName, boolean flag) {

Element search = head;

while (!search.nameF.equals(lastName)) {

search = search.next;

if (search == null) {

return;

}

}

if (search == head) {

deleteHead();

} else {

search.prev.next = search.next;

}

if (search == tail) {

deleteTail();

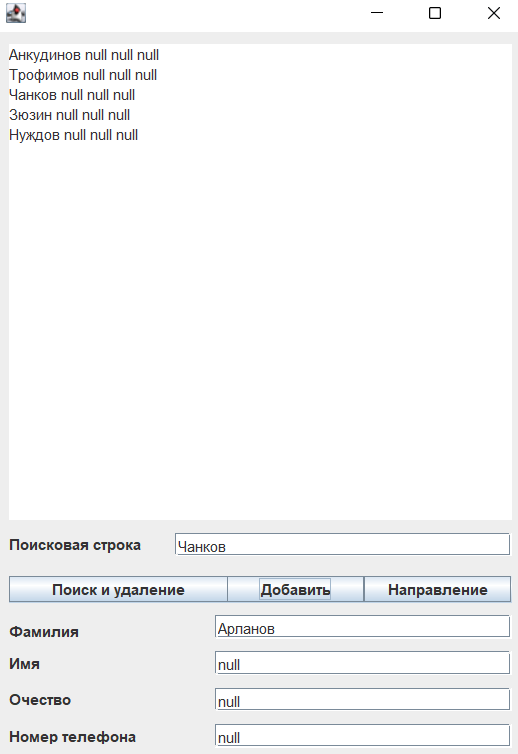
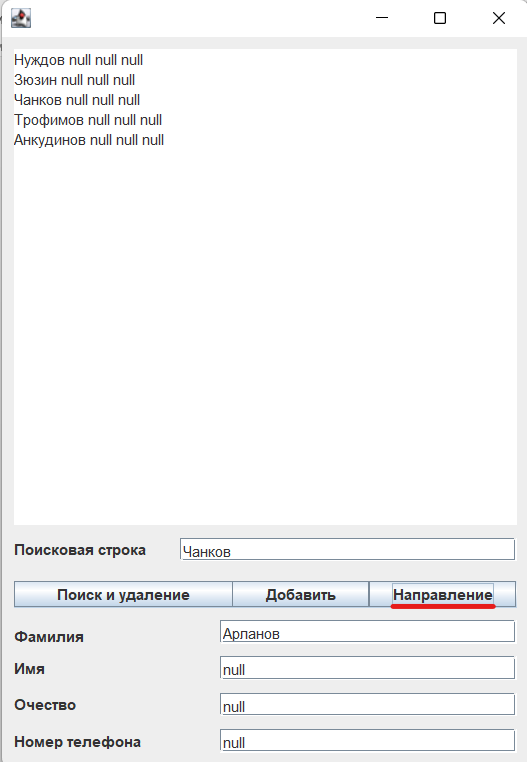
} else {

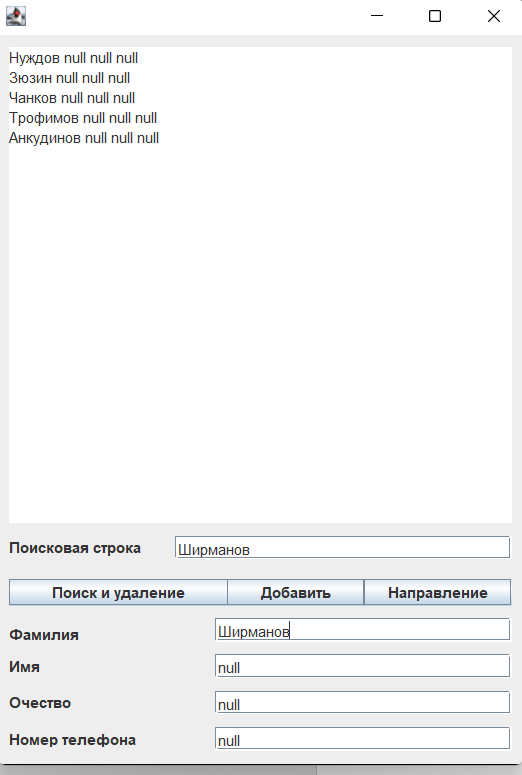
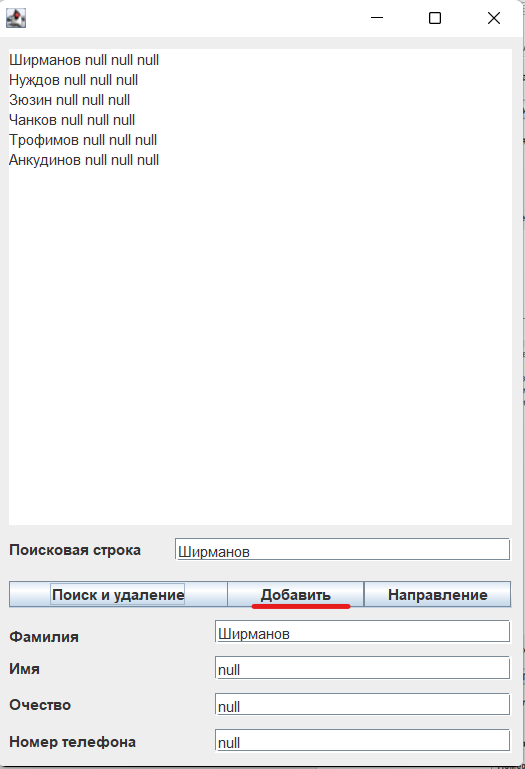
search.next.prev = search.prev;

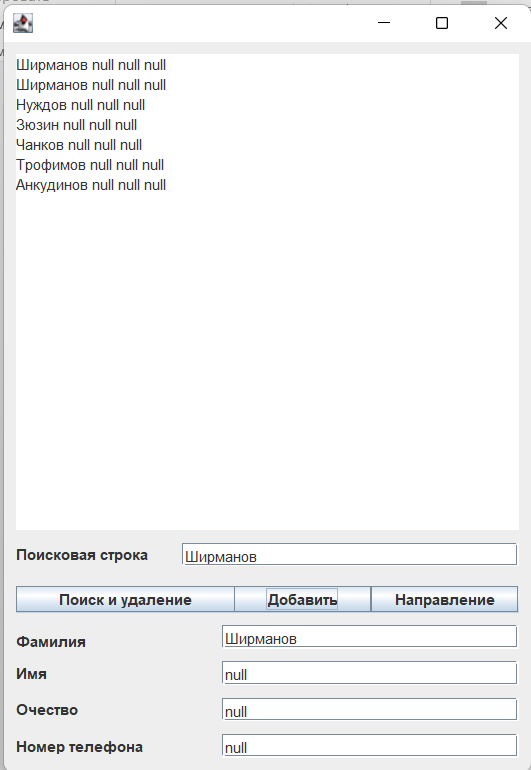
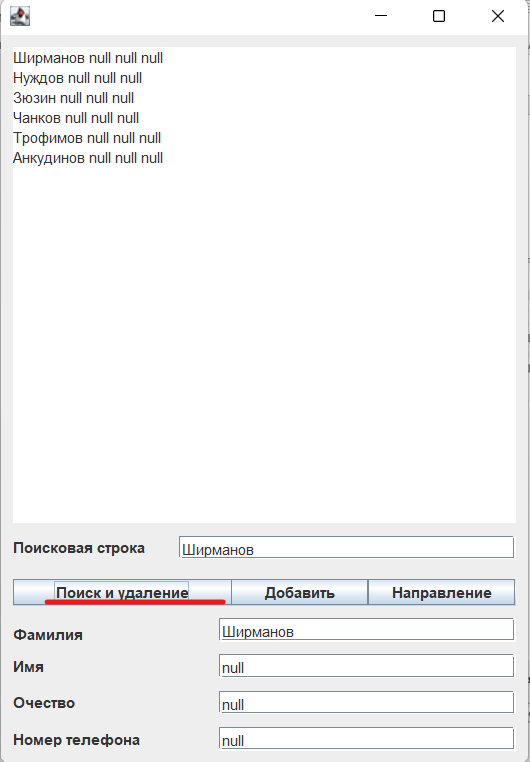
}

}

1. **Тестирование**

Смена направления списка

Добавление элемента списка

И удаление элемента

1. **Выводы о проделанной работе**

Программа протестирована и работает успешно. Программа позволяет проводить работу с двусвязным списком благодаря интерактивному интерфейсу. Можно просматривать список в прямом и обратном направлении, добавлять и удалять элементы.