

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Вычислительной техники**

**Курсовая работа**  
**по дисциплине "Программирование"**

**Тема: «Разработка электронной картотеки»**

Студент гр. 3311

\_\_\_\_\_

Шарпинский Д. А.

Преподаватель

\_\_\_\_\_

Хахаев И. А.

Санкт-Петербург

2024

## **Введение**

Электронная картотека пользователей — это система для хранения и управления данными пользователей социальной сети.

Цель работы:

Разработка программы, которая будет обеспечивать эффективное взаимодействие с электронной картотекой пользователей, хранящейся на диске.

Программа должна выполнять следующие действия:

- занесение данных в электронную картотеку;
- внесение изменений (исключение, корректировка, добавление);
- поиск данных по различным признакам;
- сортировку по различным признакам;
- вывод результатов на экран и сохранение на диске.
- Выбор подлежащих выполнению команд должен быть реализован с помощью основного меню и вложенных меню.

Задача должна быть структурирована и отдельные части должны быть оформлены как функции.

## **Постановка задачи и описание решения**

**ПРЕДМЕТНАЯ ОБЛАСТЬ:**

Пользователи социальной сети

**Постановка задачи:**

Цель данной курсовой работы — разработка электронной картотеки, которая будет храниться на диске и управляться с помощью программы.

Данные должны первоначально считываться из файлов, а в процессе работы данные должны вводиться с клавиатуры. Картотека должна храниться в памяти компьютера в виде списков и массивов структур, связанных указателями. В программе должно быть реализовано простейшее меню для выбора команд.

**Описание решения:**

Для реализации данной задачи был разработан проект на языке программирования С, в котором использована архитектура, базирующаяся на двухсвязных списках для хранения данных. Программа структурирована таким образом, чтобы каждая часть задачи была оформлена как отдельная функция, обеспечивая модульность и удобство в поддержке кода.

Основные компоненты программы включают:

- Структуры данных:

User и UserHead для хранения информации о пользователях.

Profession и ProfessionHead для хранения информации о профессиях.

- Функции для работы с данными:

1. Создание, добавление и удаление узлов: makeProfessionNode, pushBackProfessionNode, deleteProfessionNode, makeUserNode, pushBackUserNode, deleteUserNode.

2. Чтение и запись данных из/в файл: readProfessions, writeProfessionsToFile, readUsers, writeUsersToFile.

3. Поиск и фильтрация данных: findProfessionById, findProfessionByName, findUserById, filterUsersByPublicRating, filterUsersByFriendsRating, filterUsersByAge, filterUsersByFriendsCount, filterUsersByProfessionName, filterUsersByName.

4. Сортировка данных: sortUsersByField.

5. Функции для взаимодействия с пользователем:

Основное меню: printMenu, appGUI, appOption.

Функции для добавления, редактирования и удаления данных: addProfessionGUI, addUserGUI, updateUserDataGUI, deleteProfessionGUI, deleteUserGUI, clearProfessionListGUI, clearUserListGUI.

6. Ввод данных с клавиатуры: specifyUserNameGUI, specifyUserAgeGUI, specifyUserFriendsRatingGUI, specifyUserPublicRatingGUI, specifyUserProfessionGUI, specifyUserFriendsGUI.

## 7. Вспомогательные функции:

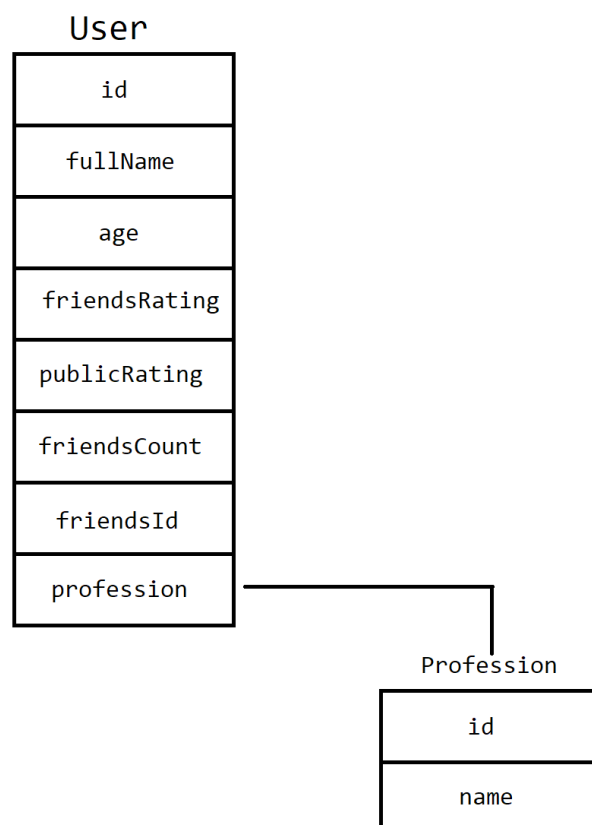
Обработка строк и ввод-вывод: `trim`, `trimForDisplay`, `clearStdin`, `pressEnterToContinue`, `clearConsole`.

Логирование и обработка ошибок: `makeLog`.

Программа начинается с инициализации списков пользователей и профессий, которые считываются из файлов CSV. После этого пользователю предоставляется меню, через которое он может управлять картотекой. Основной цикл программы управляется функцией `appGUI`, которая вызывает соответствующие функции в зависимости от выбора пользователя

Таким образом, данная структура кода обеспечивает четкое разделение логики, управления данными и взаимодействия с пользователем, что упрощает процесс разработки, тестирования и поддержки программы.

Сущности и их назначение:



## Описание переменных

Структура User:

| Название поля | Тип         | Описание                              |
|---------------|-------------|---------------------------------------|
| id            | int         | Уникальный идентификатор пользователя |
| fullName      | char*       | Полное имя пользователя               |
| age           | int         | Возраст пользователя                  |
| profession    | Profession* | Указатель на элемент списка профессий |
| friendsRating | float       | Рейтинг среди друзей                  |
| publicRating  | float       | Общественный рейтинг                  |
| friendsCount  | int         | Количество друзей                     |
| friendsId     | int*        | Массив идентификаторов друзей         |
| prev          | User*       | Указатель на предыдущего пользователя |
| next          | User*       | Указатель на следующего пользователя  |

Структура UserHead:

| Название поля | Тип   | Описание                                      |
|---------------|-------|---|
| count         | int   | Количество пользователей                      |
| first         | User* | Указатель на первого пользователя в списке    |
| last          | User* | Указатель на последнего пользователя в списке |

Структура Profession:

| Название поля | Тип         | Описание                          |
|---------------|-------------|-----------------------------------|
| name          | char*       | Название профессии                |
| prev          | Profession* | Указатель на предыдущую профессию |
| next          | Profession* | Указатель на следующую профессию  |

Структура ProfessionHead:

| Название поля | Тип         | Описание                                  |
|---------------|-------------|---|
| count         | int         | Количество пользователей                  |
| first         | Profession* | Указатель на первую профессию в списке    |
| last          | Profession* | Указатель на последнюю профессию в списке |

Функция main()

| № | Имя переменной | Тип             | Назначение                      |
|---|----------------|-----------------|---------------------------------|
| 1 | professionHead | ProfessionHead* | Метаданные списка профессий     |
| 2 | userHead       | UserHead*       | Метаданные списка пользователей |

Функция printAllProfessions()

| № | Имя переменной | Тип             | Назначение                  |
|---|----------------|-----------------|-----------------------------|
| 1 | head           | ProfessionHead* | Метаданные списка профессий |

Функция printAllUsers()

| № | Имя переменной    | Тип         | Назначение                           |
|---|-------------------|-------------|--------------------------------------|
| 1 | optionDescription | const char* | Строка с описанием текущего действия |

Функция trimForDisplay()

| № | Имя переменной | Тип         | Назначение   |
|---|----------------|-------------|--|
| 1 | output         | char*       | Указатель на строку, куда будет сохранен результат |
| 2 | input          | const char* | Входная строка, которую нужно обрезать             |
| 3 | maxLength      | int         | Максимальная длина строки                          |

### Функция printUser()

| № | Имя переменной | Тип   | Назначение                                    |
|---|----------------|-------|---|
| 1 | user           | User* | Указатель на структуру с данными пользователя |

### Функция printProfession(Profession \*profession)

| № | Имя переменной | Тип         | Назначение                                 |
|---|----------------|-------------|--|
| 1 | profession     | Profession* | Указатель на структуру с данными профессии |

### Функция Profession\* makeProfessionNode()

| № | Имя переменной | Тип   | Назначение                                |
|---|----------------|-------|---|
| 1 | name           | char* | Имя профессии, которую необходимо создать |

### Функция void pushBackProfessionNode()

| № | Имя переменной | Тип             | Назначение  |
|---|----------------|-----------------|---|
| 1 | head           | ProfessionHead* | Метаданные списка профессий                         |
| 2 | profession     | Profession*     | Указатель на узел профессии, который нужно добавить |

### void deleteProfessionNode()

| № | Имя переменной | Тип             | Назначение   |
|---|----------------|-----------------|--|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий                        |
| 2 | uHead          | UserHead*       | Метаданные списка пользователей                    |
| 3 | profession     | Profession*     | Указатель на узел профессии, который нужно удалить |

Функция void freeProfessionList()

| № | Имя переменной | Тип             | Назначение                  |
|---|----------------|-----------------|-----------------------------|
| 1 | head           | ProfessionHead* | Метаданные списка профессий |

Функция void readProfessions()

| № | Имя переменной | Тип             | Назначение                             |
|---|----------------|-----------------|--|
| 1 | filename       | char*           | Имя файла, из которого читаются данные |
| 2 | head           | ProfessionHead* | Метаданные списка профессий            |

Функция Profession\* findProfessionById()

| № | Имя переменной | Тип             | Назначение                                   |
|---|----------------|-----------------|--|
| 1 | head           | ProfessionHead* | Метаданные списка профессий                  |
| 2 | id             | int             | Идентификатор профессии, которую нужно найти |

Функция Profession\* findProfessionByName()

| № | Имя переменной | Тип             | Назначение                         |
|---|----------------|-----------------|------------------------------------|
| 1 | head           | ProfessionHead* | Метаданные списка профессий        |
| 2 | name           | char[MAXLEN]    | Имя профессии, которую нужно найти |

Функция void writeProfessionsToFile()

| № | Имя переменной | Тип             | Назначение                               |
|---|----------------|-----------------|--|
| 1 | head           | ProfessionHead* | Метаданные списка профессий              |
| 2 | filename       | const char*     | Имя файла, в который записываются данные |



Функция void fillUserNode()

| № | Имя переменной | Тип             | Назначение  |
|---|----------------|-----------------|---|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий                             |
| 2 | uHead          | UserHead*       | Метаданные списка пользователей                         |
| 3 | user           | User*           | Указатель на узел пользователя, который нужно заполнить |
| 4 | str            | char**          | Массив строк с данными для заполнения узла пользователя |

Функция void pushBackUserNode()

| № | Имя переменной | Тип       | Назначение   |
|---|----------------|-----------|--|
| 1 | head           | UserHead* | Метаданные списка пользователей                        |
| 2 | user           | User*     | Указатель на узел пользователя, который нужно добавить |

Функция void freeUserStruct()

| № | Имя переменной | Тип   | Назначение   |
|---|----------------|-------|--|
| 1 | user           | User* | Указатель на узел пользователя, который нужно освободить |

Функция void freeUserList()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | head           | UserHead* | Метаданные списка пользователей |

Функция void clearUsersProfessionById()

| № | Имя переменной | Тип       | Назначение                                      |
|---|----------------|-----------|---|
| 1 | head           | UserHead* | Метаданные списка пользователей                 |
| 2 | id             | int       | Идентификатор профессии, которую нужно очистить |

Функция void readUsers()

| № | Имя переменной | Тип             | Назначение                             |
|---|----------------|-----------------|--|
| 1 | filename       | char*           | Имя файла, из которого читаются данные |
| 2 | head           | UserHead*       | Метаданные списка пользователей        |
| 3 | pHead          | ProfessionHead* | Метаданные списка профессий            |

Функция User\* findUserById()

| № | Имя переменной | Тип       | Назначение                                       |
|---|----------------|-----------|--|
| 1 | head           | UserHead* | Метаданные списка пользователей                  |
| 2 | id             | int       | Идентификатор пользователя, которого нужно найти |

Функция void filterUsersByPublicRating()

| № | Имя переменной | Тип       | Назначение                        |
|---|----------------|-----------|-----------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей   |
| 2 | minRating      | float     | Минимальный общественный рейтинг  |
| 3 | maxRating      | float     | Максимальный общественный рейтинг |

Функция void filterUsersByFriendsRating()

| № | Имя переменной | Тип       | Назначение                        |
|---|----------------|-----------|-----------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей   |
| 2 | minRating      | float     | Минимальный рейтинг среди друзей  |
| 3 | maxRating      | float     | Максимальный рейтинг среди друзей |

Функция void filterUsersByAge()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |
| 2 | minAge         | int       | Минимальный возраст             |
| 3 | maxAge         | int       | Максимальный возраст            |

Функция void filterUsersByFriendsCount()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |
| 2 | minCount       | int       | Минимальное количество друзей   |
| 3 | maxCount       | int       | Максимальное количество друзей  |

Функция void filterUsersByProfessionName()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |
| 2 | professionName | char*     | Имя профессии                   |

Функция void filterUsersByName()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |
| 2 | name           | char*     | Имя                             |

Функция void deleteUserNode()

| № | Имя переменной | Тип       | Назначение  |
|---|----------------|-----------|---|
| 1 | head           | UserHead* | Метаданные списка пользователей                       |
| 2 | user           | User*     | Указатель на узел пользователя, который нужно удалить |

Функция int compareUsers()

| № | Имя переменной | Тип   | Назначение                                 |
|---|----------------|-------|--|
| 1 | a              | User* | Указатель на первого пользователя          |
| 2 | b              | User* | Указатель на второго пользователя          |
| 3 | option         | int   | Параметр, по которому происходит сравнение |

Функция void sortUsersByField()

| № | Имя переменной | Тип       | Назначение                                  |
|---|----------------|-----------|---|
| 1 | uHead          | UserHead* | Метаданные списка пользователей             |
| 2 | option         | int       | Параметр, по которому происходит сортировка |

Функция void writeUsersToFile()

| № | Имя переменной | Тип         | Назначение                               |
|---|----------------|-------------|--|
| 1 | head           | UserHead*   | Метаданные списка пользователей          |
| 2 | filename       | const char* | Имя файла, в который записываются данные |

Функция nullString()

| № | Имя переменной | Тип   | Назначение                                    |
|---|----------------|-------|---|
| 1 | str            | char* | Строка, которую нужно инициализировать нулями |

### Функция trim()

| № | Имя переменной | Тип   | Назначение   |
|---|----------------|-------|--|
| 1 | str            | char* | Строка, которую нужно очистить от пробелов и символов новой строки |

### Функция split()

| № | Имя переменной | Тип   | Назначение                      |
|---|----------------|-------|---------------------------------|
| 1 | str            | char* | Строка, которую нужно разделить |
| 2 | length         | int   | Длина строки                    |
| 3 | sep            | char  | Символ-разделитель              |

### Функция inputIntArray()

| № | Имя переменной | Тип       | Назначение                       |
|---|----------------|-----------|----------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей  |
| 2 | user           | User*     | Указатель на узел пользователя   |
| 3 | str            | char*     | Строка с идентификаторами друзей |
| 4 | sep            | char      | Символ-разделитель               |
| 5 | isManual       | int       | Флаг, указывающий на ручной ввод |

### Функция getUsersIdList()

| № | Имя переменной | Тип       | Назначение  |
|---|----------------|-----------|---|
| 1 | uHead          | UserHead* | Метаданные списка пользователей                   |
| 2 | dest           | int*      | Массив для хранения идентификаторов пользователей |

### Функция strcmp()

| № | Имя переменной | Тип         | Назначение                                |
|---|----------------|-------------|---|
| 1 | a              | const void* | Указатель на первый элемент для сравнения |
| 2 | b              | const void* | Указатель на второй элемент для сравнения |

### Функция binarySearch()

| № | Имя переменной | Тип        | Назначение                          |
|---|----------------|------------|-------------------------------------|
| 1 | arr            | const int* | Массив, в котором выполняется поиск |
| 2 | start          | int        | Начальный индекс для поиска         |
| 3 | end            | int        | Конечный индекс для поиска          |
| 4 | target         | int        | Целевое значение для поиска         |

### Функция startsWithIgnoreCase()

| № | Имя переменной | Тип         | Назначение                       |
|---|----------------|-------------|----------------------------------|
| 1 | str            | const char* | Строка, которую нужно проверить  |
| 2 | prefix         | const char* | Префикс, который ищется в строке |

### Функция makeLog()

| № | Имя переменной | Тип         | Назначение                                     |
|---|----------------|-------------|--|
| 1 | title          | const char* | Заголовок сообщения                            |
| 2 | funcName       | const char* | Имя функции, из которой вызывается логирование |
| 3 | log            | const char* | Текст сообщения для логирования                |

### Функция appGUI()

| № | Имя переменной | Тип             | Назначение                      |
|---|----------------|-----------------|---------------------------------|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий     |
| 2 | uHead          | UserHead*       | Метаданные списка пользователей |

### Функция appOption()

| № | Имя переменной | Тип             | Назначение                                  |
|---|----------------|-----------------|---|
| 1 | professionHead | ProfessionHead* | Метаданные списка профессий                 |
| 2 | userHead       | UserHead*       | Метаданные списка пользователей             |
| 3 | option         | int             | Выбранный пользователем номер опции из меню |

### Функция deleteProfessionGUI()

| № | Имя переменной | Тип             | Назначение                      |
|---|----------------|-----------------|---------------------------------|
| 1 | head           | ProfessionHead* | Метаданные списка профессий     |
| 2 | userHead       | UserHead*       | Метаданные списка пользователей |

### Функция addProfessionGUI()

| № | Имя переменной | Тип             | Назначение                  |
|---|----------------|-----------------|-----------------------------|
| 1 | head           | ProfessionHead* | Метаданные списка профессий |

### Функция specifyUserNameGUI()

| № | Имя переменной | Тип   | Назначение                     |
|---|----------------|-------|--------------------------------|
| 1 | user           | User* | Указатель на узел пользователя |

### Функция specifyUserAgeGUI()

| № | Имя переменной | Тип   | Назначение                     |
|---|----------------|-------|--------------------------------|
| 1 | user           | User* | Указатель на узел пользователя |

### Функция specifyUserFriendsRatingGUI()

| № | Имя переменной | Тип   | Назначение                     |
|---|----------------|-------|--------------------------------|
| 1 | user           | User* | Указатель на узел пользователя |

Функция specifyUserPublicRatingGUI()

| № | Имя переменной | Тип   | Назначение                     |
|---|----------------|-------|--------------------------------|
| 1 | user           | User* | Указатель на узел пользователя |

Функция specifyUserProfessionGUI()

| № | Имя переменной | Тип             | Назначение                     |
|---|----------------|-----------------|--------------------------------|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий    |
| 2 | user           | User*           | Указатель на узел пользователя |

Функция specifyUserFriendsGUI()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |
| 2 | user           | User*     | Указатель на узел пользователя  |

Функция updateUserDataGUI()

| № | Имя переменной | Тип             | Назначение                      |
|---|----------------|-----------------|---------------------------------|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий     |
| 2 | uHead          | UserHead*       | Метаданные списка пользователей |

Функция addUserGUI()

| № | Имя переменной | Тип             | Назначение                      |
|---|----------------|-----------------|---------------------------------|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий     |
| 2 | uHead          | UserHead*       | Метаданные списка пользователей |

Функция filterUsersByFieldGUI()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |



Функция deleteUserGUI()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | head           | UserHead* | Метаданные списка пользователей |

Функция clearProfessionListGUI()

| № | Имя переменной | Тип             | Назначение                      |
|---|----------------|-----------------|---------------------------------|
| 1 | pHead          | ProfessionHead* | Метаданные списка профессий     |
| 2 | uHead          | UserHead*       | Метаданные списка пользователей |

Функция sortUsersByFieldGUI()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | uHead          | UserHead* | Метаданные списка пользователей |

Функция clearUserListGUI()

| № | Имя переменной | Тип       | Назначение                      |
|---|----------------|-----------|---------------------------------|
| 1 | head           | UserHead* | Метаданные списка пользователей |

## Структура вызовов функций

