

## Задание практикума в машинном зале №1 (2 семестр)

### «Электронная записная книга»

Необходимо реализовать «электронную записную книгу» – хранилище данных об электронных адресах и телефонах людей. Книга представляет собою хранилище списков, организованных по иерархическому принципу. Каждый список содержит множество записей, каждая из которых содержит следующие поля: фамилию, имя, отчество, телефон, электронный адрес (все поля представляют собой строки). Для хранения адресной книги используется директория, адресные списки хранятся в файлах, расположенных в данной директории. Предлагается использовать следующую **структуру хранения** (‘./’ - текущая директория):

- ./ab.cfg – файл конфигурации адресной книги
- ./list1,...,listN – файлы, в которых хранятся адресные списки (название файла списка не совпадает с именем, видимым для пользователя адресной книги).

Пользователь имеет возможность добавлять, удалять, редактировать и просматривать имеющиеся списки контактов и записи в отдельных списках контактов. Кроме того, у пользователя имеются возможности:

- поиска отдельной записи в списке контактов по фамилии, имени и отчеству,
  - копирования списков контактов,
  - переноса отдельных записей из одного списка контактов в другой
- и т.д. (см. п. 1).

### 1. Функционирование системы

#### Запуск системы

**Пользователь** запускает систему (программу), передавая в командной строке путь до директории, в которой хранятся файл конфигурации адресной книги и файлы – списки контактов. **Система** загружает файл конфигурации, проверяет, что данный файл соответствует тем спискам контактов, которые хранятся в указанной пользователем директории, и переходит в режим ввода команд (выводит приглашение пользователю для ввода дальнейших команд).

Если **система** не может открыть файл конфигурации, то она выдаёт предупреждение пользователю и заканчивает работу.

Если при проверке корректности конфигурации **система** обнаруживает, что файлы, указанные в конфигурации, не соответствуют тем файлам, которые находятся в директории:

- в директории имеются файлы, которые не указаны в конфигурации,
  - в конфигурации указаны файлы, которых нет в директории,
- то **система** выдаёт пользователю всю информацию об обнаруженных несоответствиях и переходит в режим ввода команд.

Если при проверке корректности конфигурации **система** обнаруживает, что имеются несколько списков с одинаковыми именами, то она выдаёт предупреждение пользователю и заканчивает работу.

#### Формат файла адресного списка

Файл содержит последовательность записей вида (каждая запись начинается с новой строки):

```
<Фамилия1>,<Имя1>,<Отчество1>,<Телефон1>,<e-mail1>  
<Фамилия2>,<Имя2>,<Отчество2>,<Телефон2>,<e-mail2>
```

...

Записи должны быть упорядочены по алфавиту.

Символ ‘,’ (запятая) и ‘”’ (двойная кавычка) являются зарезервированными и не могут встречаться внутри полей. Символ «запятая» является разделителем полей, символ «двойная кавычка» используется при работе с командами.

### Добавление нового списка контактов

В командной строке **пользователь** вводит команду `new_list`. **Система** запрашивает у пользователя имя списка, создаёт новый список контактов с указанным пользователем именем в адресной книге и возвращается в режим ввода команд.

Если **пользователь** указывает имя, которое не является уникальным, **система** выдаёт предупреждение и просит **пользователя** повторить ввод.

#### Формат команды:

`new_list <name>`

где `<name>` — имя нового списка.

### Добавление нового контакта в список контактов

В командной строке **пользователь** вводит команду `new_contact`. **Система** запрашивает у пользователя имя списка, в который необходимо добавить запись, а также все поля записи (фамилия, имя, отчество и т.д.). С использованием введённых данных **система** создаёт новую запись в указанном пользователем списке контактов и возвращается в режим ввода команд.

Если **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

#### Формат команды:

`new_contact <list> <last_name> <first_name> <middle_name> <phone_number> <email>`

где `<list>` — имя списка, `<last_name>` — фамилия, `<first_name>` — имя, `<middle_name>` — отчество, `<phone_number>` — номер телефона, `<email>` — адрес электронной почты.

### Удаление списка контактов

В командной строке **пользователь** вводит команду `delete_list`. **Система** запрашивает у пользователя имя списка, который необходимо удалить, и удаляет его из адресной книги. Далее **система** возвращается в режим ввода команд.

Если **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

#### Формат команды:

`delete_list <list>`

где `<list>` — имя списка.

### Удаление записи

В командной строке **пользователь** вводит команду `delete_contact`. **Система** запрашивает у пользователя имя списка, в котором необходимо удалить запись. **Пользователь** вводит имя списка контактов, **система** отображает пронумерованный список записей, входящих в указанный список контактов и запрашивает у пользователя номер записи, которую необходимо удалить. **Пользователь** вводит номер записи, **система** удаляет указанную запись из списка контактов и возвращается в режим ввода команд.

Если **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

Если **пользователь** вводит некорректный номер или номер несуществующей записи, то **система** выдаёт предупреждение и просит пользователя повторить ввод.

#### **Формат команды:**

*delete\_contact* <list> <record\_number>

где <list> — имя списка, <record\_number> — номер удаляемой записи.

### **Просмотр существующих списков контактов**

В командной строке **пользователь** вводит команду *show\_lists*. **Система** отображает имена для всех имеющихся в адресной книге списков контактов и имена файлов, в которых они хранятся. Далее **система** возвращается в режим ввода команд.

#### **Формат команды:**

*show\_lists*

### **Просмотр списка контактов**

В командной строке **пользователь** вводит команду *show\_contacts*. **Система** запрашивает у пользователя название списка контактов, отображает содержимое списка контактов, указанного пользователем и возвращается в режим ввода команд.

Если **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

#### **Формат команды:**

*show\_contacts* <list>

где <list> — имя списка.

### **Изменение имени списка контактов**

В командной строке **пользователь** вводит команду *rename\_list*. **Система** запрашивает имя исходного списка, затем новое имя, переименовывает список и возвращается в режим ввода команд.

Если в качестве исходного **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

Если новое имя списка не является уникальным, то **система** выдаёт предупреждение и повторно запрашивает новое имя у пользователя.

#### **Формат команды:**

*rename\_list* <previous\_name> <new\_name>

где <previous\_name> — исходное **полное** имя списка, <new\_name> — новое **полное** имя списка.

### **Редактирование записи в списке контактов**

В командной строке **пользователь** вводит команду *edit\_contact*. **Система** запрашивает имя списка контактов, в котором необходимо производить редактирование. **Пользователь** вводит имя списка контактов, после чего **система** выводит пронумерованный список записей, входящих в указанный список контактов. Далее **система** запрашивает у пользователя номер записи, которую необходимо отредактировать. **Пользователь** вводит номер записи, **система** последовательно запрашивает у пользователя новые значения для всех полей данной записи и производит их сохранение в списке.

Если **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

Если **пользователь** указывает некорректный номер или номер несуществующей записи, то **система** выдаёт предупреждение и просит повторить ввод.

Если в ходе редактирования **пользователь** вводит пустое поле, то **система** оставляет старое значение этого поля.

#### **Формат команды:**

*edit\_contact* <list> <record\_number> <last\_name> <first\_name> <middle\_name>  
<phone\_number> <email>

где <list> — имя списка, <record\_number> — номер удаляемой записи, <last\_name> — фамилия, <first\_name> — имя, <middle\_name> — отчество, <phone\_number> — номер телефона, <email> — адрес электронной почты.

### **Поиск записей в списке контактов**

В командной строке **пользователь** вводит команду *find\_contact*. **Система** запрашивает у пользователя имя списка, в котором необходимо произвести поиск. **Пользователь** вводит имя списка, **система** последовательно запрашивает фамилию, имя и отчество для записи, которую нужно найти, производит поиск и выводит все поля для всех записей, соответствующих указанным критериям поиска. Далее **система** возвращается в режим ввода команд.

Если **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

Если при вводе полей фамилия, имя, отчество **пользователь** пропускает одно из полей, то при поиске в списке контактов **система** не учитывает данное поле.

#### **Формат команды:**

*find\_contact* <list> <last\_name> <first\_name> <middle\_name>

где <list> — имя списка, <last\_name> — фамилия, <first\_name> — имя, <middle\_name> — отчество.

### **Создание копии списка контактов**

В командной строке **пользователь** вводит команду *make\_list\_copy*. **Система** запрашивает имя исходного списка контактов, имя копии и создаёт точную копию исходного списка под указанным пользователем именем копии. Далее **система** возвращается в режим ввода команд.

Если при запросе исходного списка **пользователь** указывает имя несуществующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

Если при запросе имени списка копии **пользователь** указывает имя существующего списка, то **система** выдаёт предупреждение и просит повторить ввод.

#### **Формат команды:**

*make\_list\_copy* <original\_name> <new\_name>

где <original\_name> — исходное имя списка, <new\_name> — имя копии списка.

### **Копирование записи**

В командной строке **пользователь** вводит команду *make\_contact\_copy*. **Система** запрашивает имя списка, в котором хранится копируемая запись. **Пользователь** вводит имя списка контактов, **система** отображает пронумерованный список всех записей, входящих в данный список контактов, и запрашивает у пользователя номер записи. **Пользователь** вводит номер записи, после чего **система** запрашивает имя списка контактов в котором система должна создать копию записи с данным номером.

Пользователь вводит имя списка контактов, система создаёт в нём копию выбранной пользователем записи и возвращается в режим ввода команд.

Если при запросе имён списков контактов **пользователь** указывает имя несуществующих списков, то **система** выдаёт предупреждение и просит повторить ввод.

Если **пользователь** указывает некорректный номер или номер несуществующей записи, то **система** выдаёт предупреждение и просит повторить ввод.

#### **Формат команды:**

*make\_contact\_copy* <source\_list> <record\_number> <dest\_list>

где <source\_list> — имя списка, откуда берется информация о записи, <record\_number> — номер копируемой записи, <dest\_list> — имя списка, куда копируется запись.

### **Перенос записи из одного списка в другой**

Команда *move\_contact* аналогична предыдущей, только **система** не копирует запись, а переносит её из одного списка в другой.

#### **Формат команды:**

*move\_contact* <source\_list> <record\_number> <dest\_list>

где <source\_list> — имя списка, откуда берется информация о записи, <record\_number> — номер перемещаемой записи, <dest\_list> — имя списка, куда переносится запись.

### **Выход из системы**

В командной строке **пользователь** набирает команду “exit”, **программа** завершает свою работу.

#### **Формат команды:**

*exit*

## **2. Замечания о реализации**

1. Каждый список контактов хранится в отдельном текстовом файле. Имена списков и соответствующие им имена файлов задаются в файле конфигурации с предопределенным именем «ab.cfg». Предполагается, что все файлы данных (т.е. файлы списков контактов и файл конфигурации) находятся в одной директории, которая задается в командной строке при запуске программы.
2. Имя списка контактов и имя файла, в котором он хранится, могут не совпадать. При добавлении нового списка контактов в адресную книгу уникальное имя файла для хранения нового списка генерируется системой автоматически.
3. Длина списков контактов и общее количество списков в адресной книге не ограничены.
4. Имена списков контактов в адресной книжке являются уникальными, и везде, где это необходимо, сравнение строк (названий списков, фамилий, телефонов, эл. адресов и т.п.) производится с точностью до регистра символов (буквы верхнего и нижнего регистров считаются одинаковыми).
5. Пользователь работает с программой через командную строку. В случае, если пользователь вводит неизвестную команду, программа должна выдать предупреждение об этом.
6. Система должна поддерживать как интерактивный режим работы (когда происходит диалог с пользователем в режиме реального времени), так и командный, когда система считывает команду со всеми необходимыми параметрами. Подразумевается, что если указано лишь имя команды, то **система** должна в диалоговом режиме запросить все необходимые для выполнения команды данные. Если же после имени команды следует хотя бы один параметр, то считается, что это командный режим и,

следовательно, вся командная строка должна соответствовать объявленному выше синтаксису (в противном случае **система** должна диагностировать ошибку).

- a. Если текущая команда подразумевает вывод пронумерованного списка контактов, то в командном режиме **система** этого не делает (но соглашения о нумерации остаются теми же).
7. Для команд, выводящих пронумерованный список записей, подразумевается, что записи выводятся в том порядке, в котором они хранятся внутри файла. Нумерация начинается с 1 (первая запись из файла появляется под номером 1).
8. В **системе** используются следующие зарезервированные символы:
  - a. Символ '-' (минус) для разделения имени списка контактов и имени файла в конфигурационном файле.
  - b. Символ ',' (запятая) для разделения полей записи внутри файла списка контактов.
  - c. Символ '"' (двойная кавычка) используется для того, чтобы указать, что заключенный в кавычки текст, воспринимался как один параметр. Например, "" обозначает, пустую строку, а "Petroff Jr." воспринимается как значение одного параметра, а не двух.
  - d. Одиночный символ пробела используется в качестве разделителя в командных строках.
9. Сохранение измененных данных в файлах может происходить по мере необходимости, но в обязательном порядке измененные данные должны быть сохранены к моменту выхода из программы.
10. При удалении списка контактов программа удаляет не только запись о нём в адресной книге, но и файл, соответствующий ему.
11. Все предупреждения, которые выводятся программой для пользователя, должны быть осмысленными и содержать информацию, достаточную, чтобы понять, что именно произошло.
12. Программа должна быть реализована в виде совокупности классов на языке C++. Необходимо уметь внятно обосновывать свой выбор при разработке иерархии классов и связей между ними.

### 3. Дополнительные сведения

Чтобы узнать содержимое папки, можно воспользоваться следующими функциями:

```
#include <sys/types.h>
#include <dirent.h>
...
DIR *dirp;
struct dirent *dp;
...
dirp = opendir("dirpath");
while ((dp = readdir(dirp)) != NULL)
    printf("%s", dp->d_name);
(void)closedir(dirp);
```