C++ 프로그래밍및실습(3)

주소록

최종 보고서

제출일자: 2023.11.26
제출자명: 이규민
제출자학번: 215118

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

현대 사회에서는 다양한 사람들과의 소통이 필수적이며, 이를 위해 효과적인 연락처 관리가 필요합니다. 이 프로젝트는 사용자가 간편하게 연락처를 추가, 검색, 수정, 삭제하고 관리할 수 있는

연락처 관리 시스템을 제공합니다.

전통적인 연락처 관리 프로그램과의 차별점은 다양한 검색 옵션과 그룹 관리 기능, 사용자 친화적인 인터페이스로 효율적이고 편리한 사용성을 제공한다는 것입니다.

2) 프로젝트 목표

사용자가 연락처를 추가, 검색, 수정, 삭제하고 관리할 수 있는 간편하면서도 다양한 기능을 제공하는 연락처 관리 시스템을 구현합니다.

다양한 검색 옵션을 통해 빠르게 원하는 연락처를 찾을 수 있도록 합니다.

연락처를 그룹화하여 관리할 수 있는 기능을 추가하여 더 효과적인 연락처 관리를 지원합니다.

3) 차별점

그룹 관리 기능 추가 : 기존 연락처 관리 프로그램에서는 그룹화 기능이 제한적이었으나, 이 프로젝트는 연락처를 그룹으로 나누어 관리할 수 있도록 지원합니다.

다양한 검색 옵션 : 이름, 전화번호, 이메일, 주소, 그룹 등 다양한 기준으로 연락처를 검색할 수 있습니다.

친화적인 사용자 인터페이스 : 사용자가 직관적으로 기능을 이해하고 사용할 수 있도록 친화적인 명령어와 설명을 제공합니다.

2. 기능 계획

1) 연락처 추가

- 사용자가 연락처를 추가할 때 필수 정보인 이름과 전화번호를 입력하도록 요구하며, 추가 정보로는 이메일, 주소, 그룹을 선택적으로 입력받습니다. 연락처 추가 시, 입력 받은 정보를 파일에 저장하도록 구현합니다.

(1) 입출력

- 사용자로부터 이름, 전화번호, 이메일, 주소, 그룹을 입력받음.

- 적용된 배운 내용:

    - 구조체 사용

    - 벡터 사용

    - 반복문과 조건문 활용

    - 함수(addContact) 정의와 호출

    - Getline 함수를 사용한 문자열 입력

    - 파일 입출력

2) 연락처 검색

- 설명 : 사용자는 이름, 전화번호, 이메일 주소, 주소, 그룹 등의 기준을 사용하여 연락처를 검색할 수 있습니다. 검색된 결과를 콘솔창에 출력하는 것 뿐만 아니라, 검색 결과를 파일에 저장하도록 구현합니다.

(1) 입출력

- 사용자로부터 검색 기준과 검색어를 입력받음.

- 적용된 배운 내용:

    - Switch 문을 사용한 다양한 검색 옵션 제공
    - 함수 (searchContact, partialSearchByName) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력

3) 연락처 수정

- 사용자는 저장된 연락처의 정보를 수정할 수 있습니다. 연락처를 수정하면 수정된 내용을 파일에 업데이트하도록 구현합니다.

(1) 입출력

- 사용자로부터 수정할 연락처의 이름과 새로운 정보를 입력받음.

- 적용된 배운 내용:

    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수(modifyContact) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력

4) 연락처 삭제

- 사용자는 더 이상 필요하지 않는 연락처를 삭제할 수 있습니다. 연락처를 삭제하면 삭제된 연락처를 파일에서도 제거하도록 구현합니다.

(1) 입출력

- 사용자로부터 삭제할 연락처의 이름을 입력받음.

- 적용된 배운 내용:
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수(deleteContact) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력

3. 기능 구현

(1) 연락처 추가

- 입출력
    - 사용자로부터 이름, 전화번호, 이메일, 주소, 그룹을 입력받음.
- 설명
    - 사용자로부터 입력받은 정보를 새로운 Contact 객체로 만들어 전역 연락처 목록에 추가.
    - 추가된 연락처 정보를 성공 메시지와 함께 출력
- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수(addContact) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력
- 코드 스크린샷

(2) 전체 연락처 조회

- 입출력
    - 사용자에게 연락처의 전체 목록을 출력
- 설명
    - 전역 연락처 목록에 있는 각 연락처의 정보를 출력
    - 각 연락처는 이름, 전화번호, 이메일, 주소, 그룹을 포함한 정보를 출력.
- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)
    - Switch 문을 사용한 다양한 검색 옵션 제공
    - 함수 (searchContact, partialSearchByName) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력
- 코드 스크린샷

(3) 연락처 수정

- 입출력
    - 사용자에게 수정할 연락처의 이름과 새로운 정보를 입력받음.
- 설명
    - 입력받은 이름으로 연락처를 찾아 해당 연락처의 정보를 사용자에게서 다시 입력받아 업데이트.
    - 수정된 연락처 정보를 성공 메시지와 함께 출력
- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수(modifyContact) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력
- 코드 스크린샷

(4) 연락처 삭제

- 입출력
    - 사용자로부터 삭제할 연락처의 이름을 입력받음
- 설명
    - 입력받은 이름으로 연락처를 찾아 해당 연락처를 목록에서 삭제.
    - 삭제된 연락처 정보를 성공 메시지와 함께 출력.
- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수(deleteContact) 정의와 호출
    - Getline 함수를 사용한 문자열 입력
    - 파일 입출력
- 코드 스크린샷

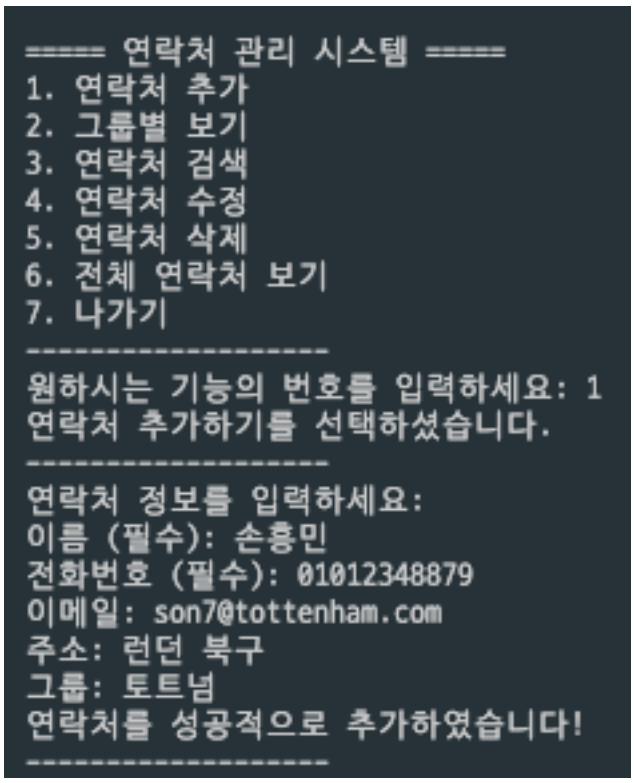
4. 테스트 결과

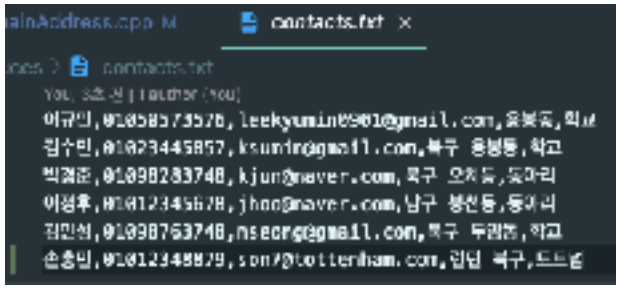
(1) 연락처 추가 테스트

- 사용자로부터 연락처의 정보를 입력받습니다

- 입력받은 연락처가 contacts.txt에 올바르게 추가되는지 확인합니다.

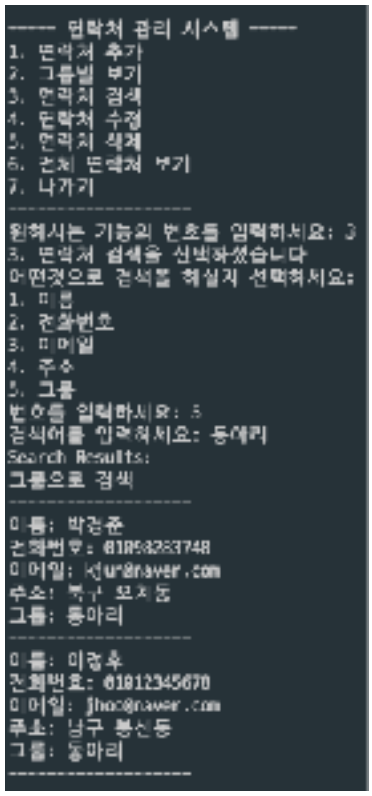
- 테스트 결과 스크린샷

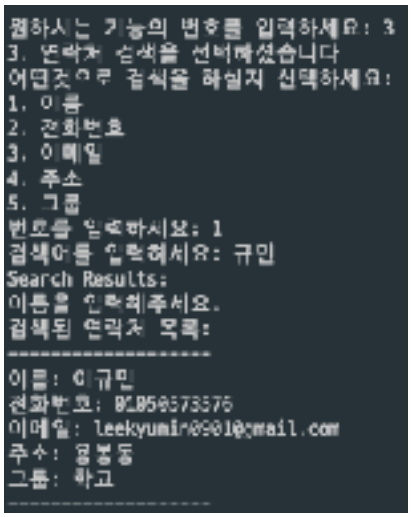




(2) 전체 연락처 조회 테스트

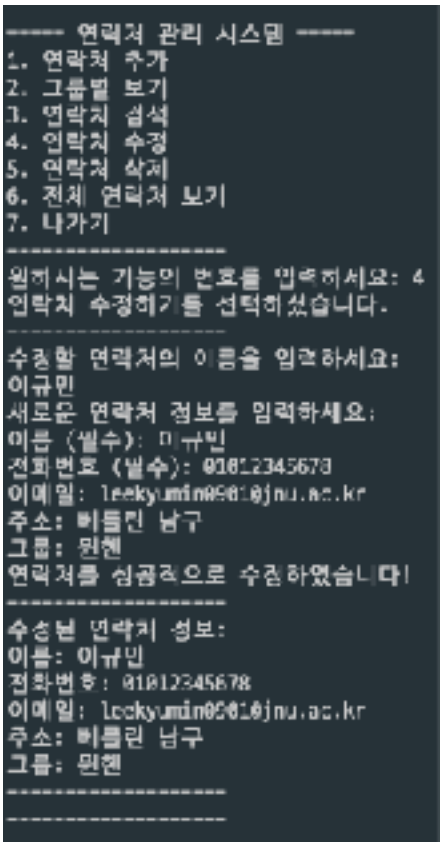
- 전체 연락처를 조회할 수 있습니다.
- 또한, 그룹, 전화번호, 이름, 주소, 이메일로 연락처 각각 조회가 가능합니다.
    - 전체 연락처 보기를 통해, 콘솔창에 연락처가 올바르게 출력되는지 확인합니다.
    - 상단의 각 분야별로 조회가 가능한지 확인합니다.
- 테스트 결과 스크린샷

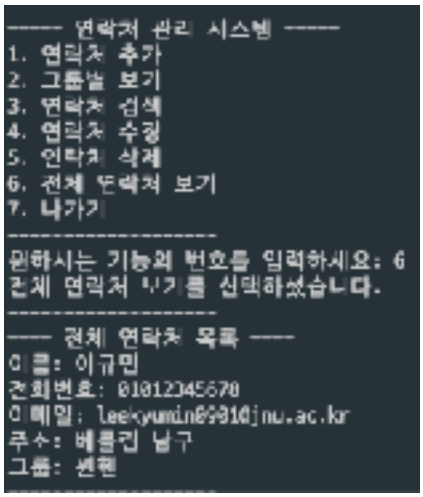


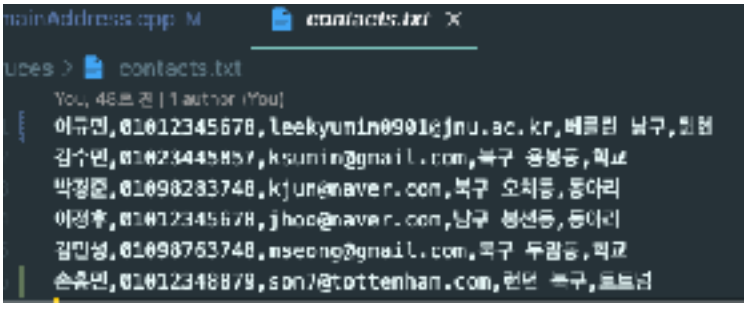


(3) 연락처 수정 테스트

- 입력받은 이름으로 원하는 연락처를 사용자가 찾습니다.
- 찾은 뒤, 해당 연락처의 정보를 사용자로부터 다시 입력받아 업데이트합니다.
    - 이름으로 연락처를 올바르게 찾을 수 있는지 테스트합니다
    - 새로 입력받아 업데이트 된 값이 콘솔창에 출력되는것을 확인합니다.
    - 연락처.txt 파일에 제대로 수정 내용이 반영되는것을 확인합니다.
- 테스트 결과 스크린샷

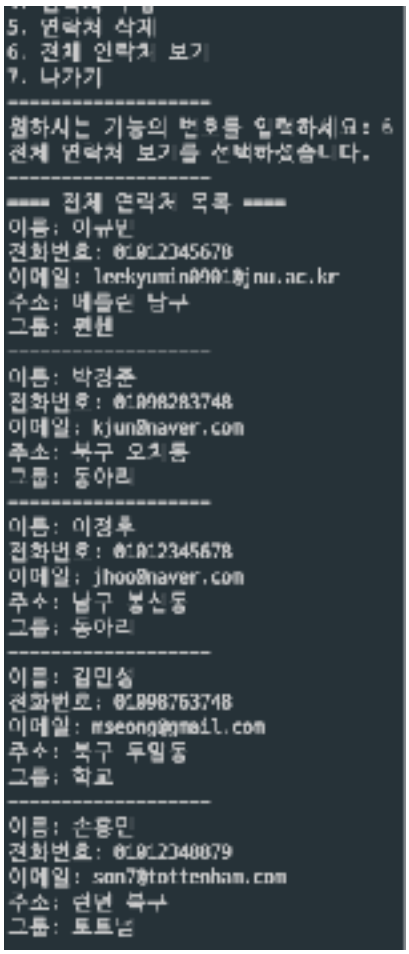


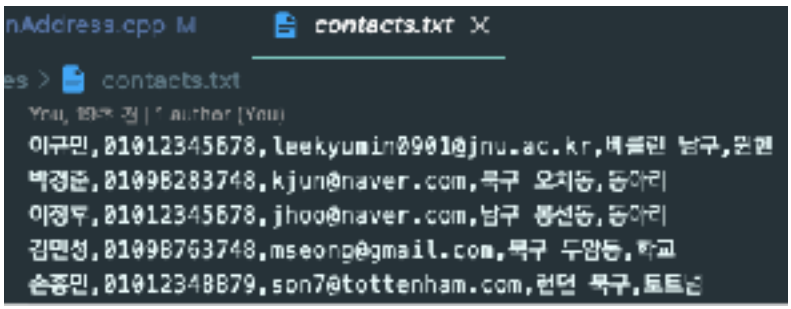


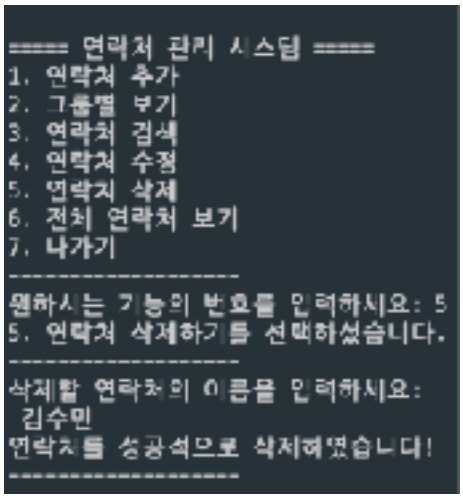


(3) 연락처 삭제 테스트

- 입력받은 이름으로 원하는 연락처를 사용자가 찾습니다.
- 찾은 뒤, 해당 연락처의 정보를 삭제합니다.
    - 이름으로 연락처를 올바르게 찾을 수 있는지 테스트합니다
    - 새로 삭제된 값이 콘솔창에 출력되는것을 확인합니다.
    - 연락처.txt 파일에 제대로 삭제 내용이 반영되는것을 확인합니다.
- 테스트 결과 스크린샷







5. 계획 대비 변경 사항

1) 검색기능 강화

- 이전
    - 단순 CRUD 주소록 구현
- 이후
    - 검색기능 강화
    - 이름, 주소, 연락처, 그룹 등 키워드 검색으로 해당 연락처가 검색이 가능하게 함
- 사유
    - 교수님 issue 반영

2) 파일 입출력

- 이전
    - 1차적으로 단순 콘솔창 출력 구현
    - mockData 를 이용하여 출력 구현
- 이후
    - 파일 입출력 기능을 이용하여 contact.txt 라는 텍스트 파일을 생성하여, 해당 파일 내에서 콘솔창에서 사용자의 입력에 맞게 추가, 업데이트 및 수정, 삭제가 반영되도록 함
- 사유
    - 1차적으로 콘솔창 출력 구현시
        - 너무 복잡하고, 계속 프로그램 종료 이후에 계속 데이터를 쌓고, 기존의 데이터(연락처)가 삭제됨. 이는 '연락처'라는 기능을 하지 못하는 것과 다름없다 생각
    - 2차적으로 mockData를 이용하여 출력 구현
        - 그럼 mockData를 다른 파일에 생성하고, 해당 파일을 수정하면 어떨까? 라는 생각을 함
        - 하지만, 상수로 지정해놓은 mockData는 수정이 불가능하였고, 이는 그냥 콘솔창에서만 반영이 가능하고, 결국 mockData를 작성해놓은 파일에 가면 이전과 동일한 데이터만 남아있는 것을 볼 수 있었음
        - 이 또한 '연락처'라는 기능을 핮 못하는 것과 다름없다 생각

6. 느낀점

사실 하고싶은건 정말 많았으나, 능력 부족과 그에 따른 시간의 한계로 파일 입출력, CRUD를 구현하는 것에 그쳤습니다.

객체지향 프로그래밍, 함수화를 통해서 조금더 가독성 좋은 프로그램을 만들어 보고 싶었는데, 그보다 구조체, 벡터 등을 이용하는 것에 급급했고, 그조차 버거웠습니다.

사용자 입장에서 더 좋은 프로젝트를 만들고자 하여, 파일 입출력을 구현한 부분은 괜찮았으나, 아예 csv 파일로 작성하고, '정렬' 기능까지 구현하고자 하였습니다.

하지만 계속 csv 파일로 변환하는 과정에서 너무나 많은 오류가 발생하였고, 결국 현재 파일을 제출하게 되었습니다.

추가로, 그룹화에 의한 정렬 및 해당 이름만 나오게 출력하는 기능을 구현하였습니다. 하지만 그러한 경우에 파일 입출력에 문제가 생겨, 결국 해당 기능은 커밋에만 남아있고, 현재 코드로 최종본이 완성되었습니다.

이번 프로젝트를 통해 코드 작성 전 '기능 요구사항 정리' 와 '개발 기간을 보수적으로 잡는 것' 그리고 구현했다고 끝이 아니라, 이 구현한 기능이 다른 프로그램의 기능에 어떤 영향을 미칠지까지 생각하는 것이 중요하다는 것을 수많은 에러를 통해 느꼈습니다.

아직도 어려운 코딩이지만, 이제 조금은 더 가까워진 것 같습니다. 더 나은 프로그램을 방학때 만들어 볼 수 있도록 하겠습니다. 감사합니다.