

# 소프트웨어공학



AI소프트웨어학과

202124311

김경렬

## ■ 목차

### ☐ 요구분석

- 전화번호부 프로그램 설명

### ☐ 프로그램 설계 및 구현

- 플로우 차트 설명
- 함수 설명

### ☐ 테스트 결과

- 결과 분석
- 원인 분석



## 1, 전화번호부 프로그램 설명

전화번호부를 생각할 때 본인이 필수로 가지고 있어야 된다고 생각하는 기능은 전화번호를 저장하는 기능과 전화번호를 검색하는 기능, 그리고 삭제하는 기능이다. 전화번호를 검색할 때 동명이인이 있는 경우 같은 이름의 전화번호들을 전부 출력할 수 있어야 되고 마찬가지로 전화번호를 삭제할 때 동명이인이 있는 경우 같은 이름의 전화번호들을 전부 출력하고 그중에서 이용자가 원하는 전화번호를 삭제할 수 있도록 만들어야 된다.

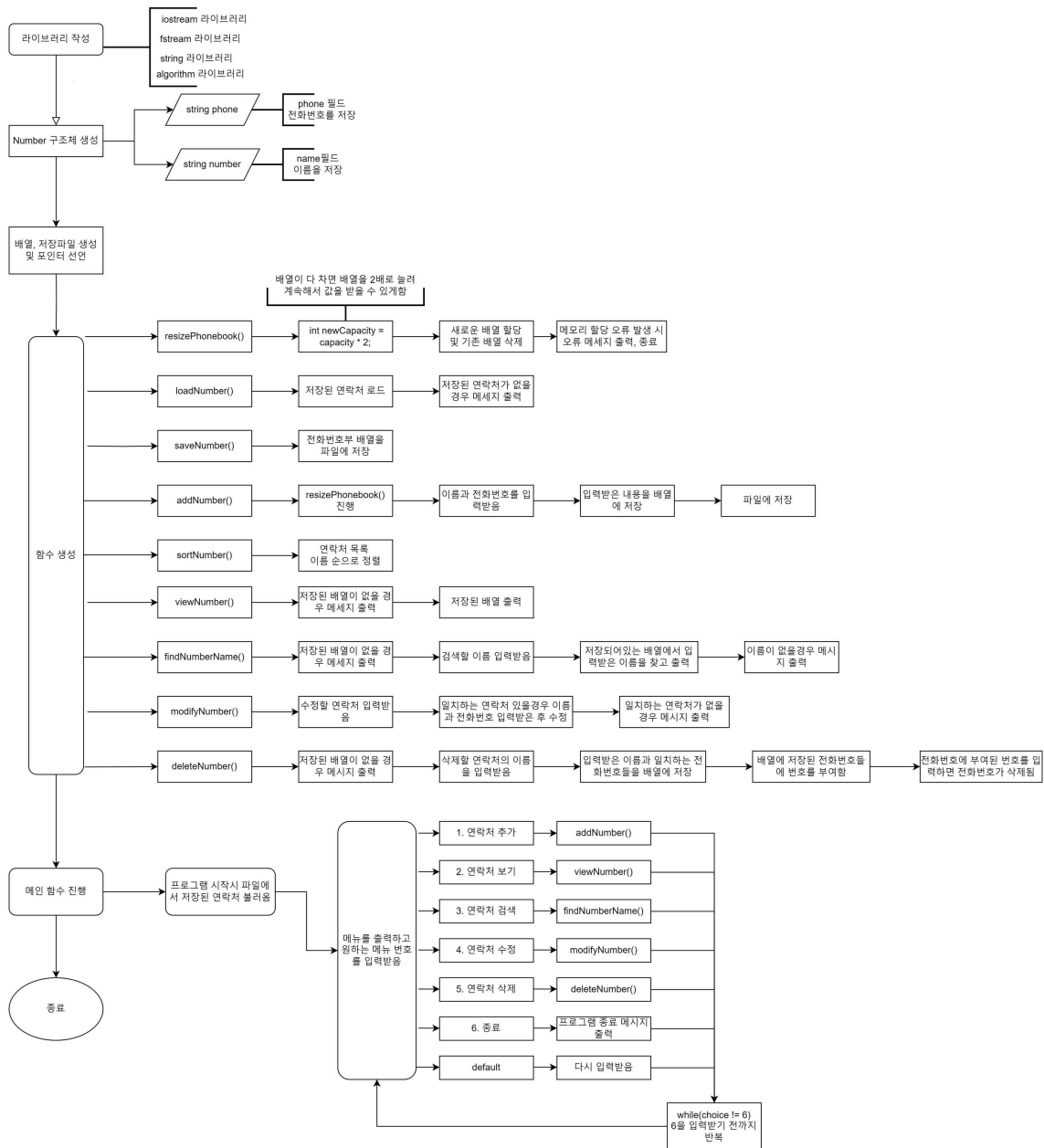
전화번호를 검색할 때 동명이인이 있는 경우 먼저 입력된 사람의 번호만 출력되게 나오면 이용자들이 급할 때 원하는 전화번호를 찾기에 어려움이 생겨 사용하지 않게 된다. 추가로 이용자들이 전화번호를 목록을 확인할 때 저장 순서대로 출력될 경우 원하는 전화번호를 찾는 데에 있어 불편함이 생기기에 이름순으로 전화번호 목록이 출력되도록 하여야된다.

마찬가지로 전화번호를 삭제할 때 이용자들이 전화번호를 삭제하고 싶은 사람의 이름을 검색할 때 다른 동명이인의 번호가 삭제된다면 이용자들은 해당 이름의 전화번호를 모두 삭제하고 다시 전화번호들을 추가하는 불편함이 생긴다.

또한 전화번호부 프로그램을 사용하는 이용자들의 기기엔 메모리가 무한이 아니기 때문에 메모리 할당에 있어 오류가 생길 수 있다. 메모리 할당에 오류가 생길 경우 무엇 때문에 오류가 생기는 것인지 알 수 없는 이용자들은 이용에 있어 어려움을 겪게 된다. 이를 해결하기 위해선 명확하게 메모리 할당에 오류가 생겼음을 알리는 메시지가 출력되어야 된다.

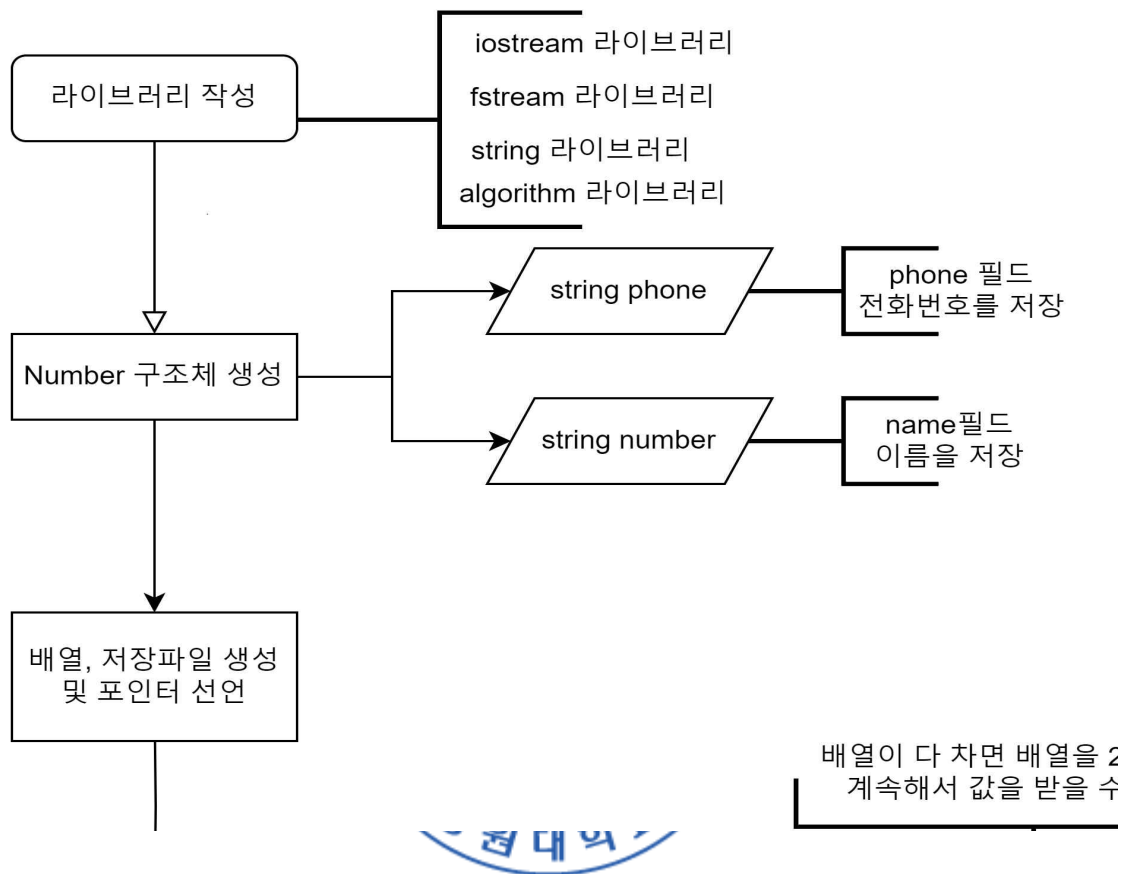
위의 기능들도 중요하지만 가장 중요한 것은 프로그램을 종료하고 다시 실행했을 때, 이전에 입력해둔 전화번호들이 사라질 경우 전화번호부의 역할을 못하여 이용자들은 사용하지 않을 것이다. 그래서 본인은 프로그램이 종료되더라도 다음에 실행 할 때 이전에 입력한 전화번호들이 저장되도록 하는 기능이 있어야 된다고 생각한다.





## 2. 프로그램 설계 및 구현

위의 플로우 차트는 이 프로그램의 전체적인 진행 순서도를 나타낸다.

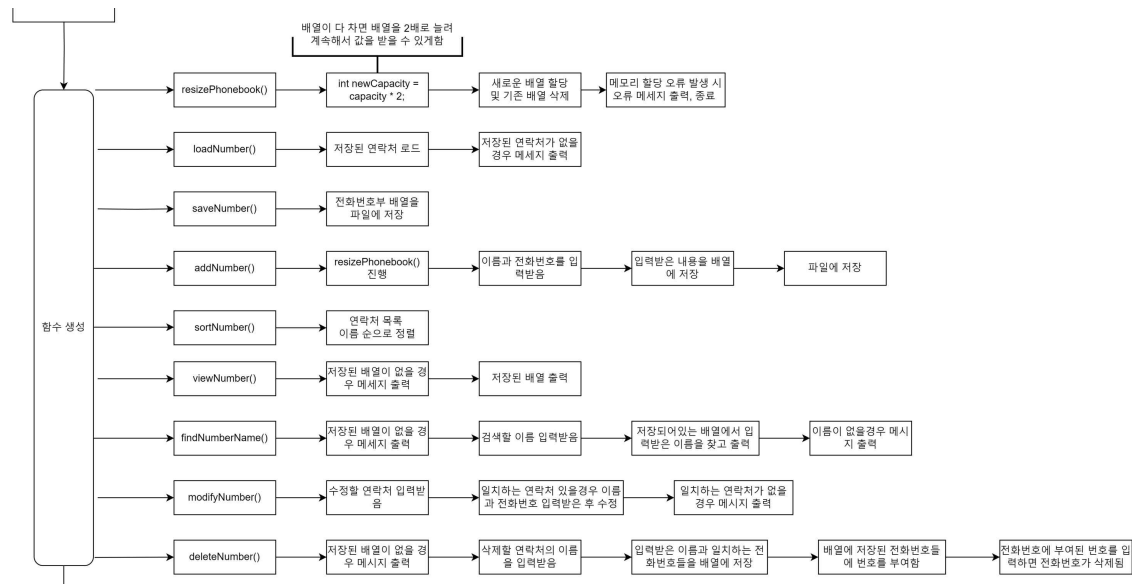


먼저 플로우 차트의 초반 부분이다. cin과 cout를 사용하기 위해 iostream 라이브러리를 사용하였고, 입력된 전화번호들을 프로그램을 종료하고 다시 실행할 때까지 저장할 수 있도록 하기 위해 fstream 라이브러리를 사용하였다. string 라이브러리는 아래에서 전화번호와 이름을 저장할 때 문자열로 입력 받기 때문에 string 라이브러리를 사용하였다.

algorithm 라이브러리는 전화번호를 이름순으로 나열할 것이기에 sort() 함수를 사용을 위해 사용하였다.

Number 구조체를 생성하고 phone 필드와 number 필드를 생성하였는데 전화번호를 입력받을 때 공백도 받을 수 있도록 하기 위해 string을 사용하였다.

이 프로그램은 배열을 사용 할 것이기에 배열 생성과 포인터 선언을 하고, 프로그램 종료 후 실행 시에 이전에 입력한 전화번호와 이름을 불러올 수 있도록 저장파일을 생성했다.



위의 플로우 차트는 함수 생성 부분을 나타낸다. `resizePhonebook()` 함수에선 `int newCapacity = capacity * 2;`를 선언해 배열이 다 찼을 경우 배열을 2배로 늘려주도록 하였다.

`loadNumber()` 함수는 이전 프로그램 실행시에 파일에 저장된 전화번호와 이름 불러온다.

`saveNumber()` 함수는 이전 프로그램 실행시에 입력된 전화번호와 이름을 파일에 저장한다.

`addNumber()` 함수는 배열이 다 찼을 경우 `resizePhonebook()` 함수를 실행하여 배열을 늘리고 이름과 전화번호를 입력받아 배열에 저장하고, `saveNumber()` 함수를 실행하여 파일에 저장한다.

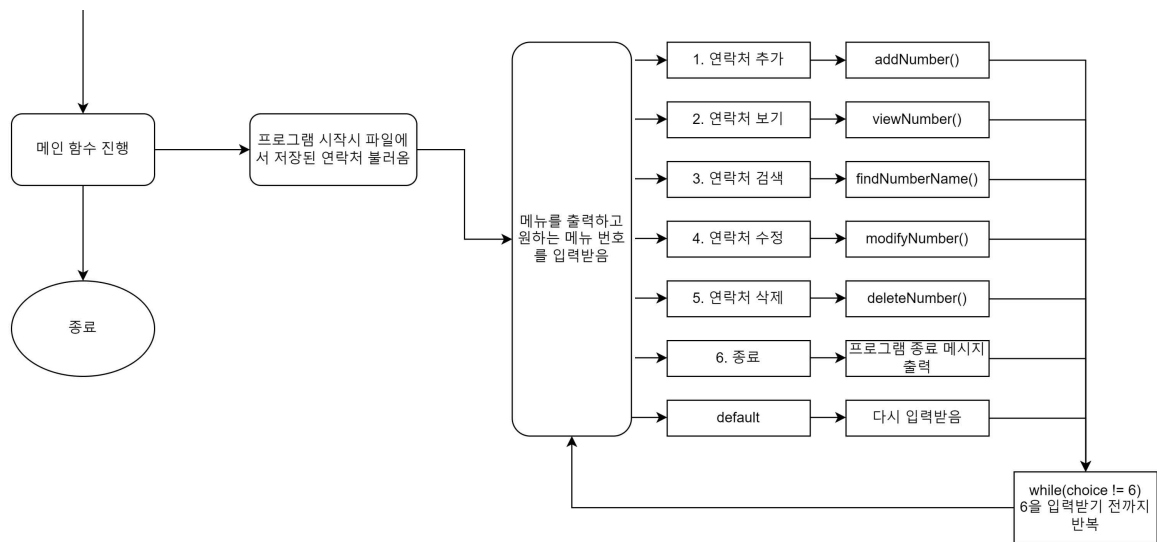
`sortNumber()` 함수는 연락처 목록을 이름 순으로 정렬시킨다.

`viewNumber()` 함수는 저장된 배열이 없을 경우 저장된 전화번호가 없다는 메시지를 출력하고 저장된 배열이 있을 경우 출력한다.

`findNumberName()` 함수는 저장된 배열이 없을 경우 저장된 전화번호가 없다는 메시지를 출력하고 저장된 배열이 있을 경우 이용자로부터 이름을 입력받은 다음 저장된 연락처에서 입력된 이름과 같은 전화번호를 출력한다.

`modifyNumber()` 함수는 입력된 전화번호와 이름을 수정하는 함수로 동명이인이 있을 경우 한 명을 선택하여 수정할 수 있고, 입력받은 이름과 일치하는 전화번호가 없을 경우 메시지를 출력한다.

`deleteNumber()` 함수는 입력된 전화번호와 이름을 삭제하는 함수로 동명이인이 있을 경우 한 명을 선택하여 삭제할 수 있고, 입력받은 이름과 일치하는 전화번호가 없을 경우 메시지를 출력한다.



함수를 생성한 다음 메인 함수가 진행되는데 먼저 프로그램 시작 시 파일에서 저장된 연락처들을 불러온다. 그 다음 메뉴를 출력하고 원하는 메뉴 번호를 입력받은 뒤 입력된 번호와 맞는 메뉴가 실행이 된다. 1을 입력할시 연락처를 추가하는 메뉴가 실행되고 2를 입력시엔 연락처 목록을 전부 확인할 수 있는 메뉴가 실행이 되고 3을 입력할때 연락처를 검색하여 원하는 전화번호를 확인할 수 있는데 메뉴가 실행이 된다. 4를 입력하면 수정하고 싶은 연락처를 입력받아 원하는 전화번호로 수정할 수 있는 메뉴가 실행이 되며 5를 입력 받을 시엔 원하는 연락처를 삭제하는 메뉴가 실행된다. 6을 입력 받을 시엔 프로그램을 종료한다는 메시지를 출력하도록 하였고 이때 단순 메시지만 출력되는 것이 아닌 프로그램을 실제로 종료 시키기 위해 while(choice != 6)을 사용하여 6을 입력받으면 프로그램이 종료되고 6을 입력받기 전까진 1~5번의 메뉴들을 반복하여 사용할 수 있도록 만들었다.

### 3. 테스트 결과

전화번호부

1. 연락처 추가
  2. 연락처 보기
  3. 연락처 검색
  4. 연락처 수정
  5. 연락처 삭제
  6. 종료
- 선택 : |

먼저 실행 창이다. 초기에 본인의 의도대로 연락처 추가, 연락처 보기, 연락처 검색, 연락처 수정, 연락처 삭제, 종료의 메뉴들이 나타났다. 여기서 메뉴들이 잘 작동하는지 확인하기 위해 먼저 1을 입력하여 연락처를 추가 해보도록 하겠다.



선택 : 1  
이름을 입력하세요 : 김경렬  
전화번호를 입력하세요 : 010-1111-1111  
연락처가 추가되었습니다.

1. 연락처 추가
  2. 연락처 보기
  3. 연락처 검색
  4. 연락처 수정
  5. 연락처 삭제
  6. 종료
- 선택 : |

1을 입력하자 이름과 전화번호를 입력할 수 있는 창이 나왔다. 이름과 전화번호를 입력하자 연락처가 추가되었다는 메시지가 출력되었는데 연락처가 제대로 추가되었는지 확인하기 위해 2를 입력해 연락처 목록을 확인해보겠다.

6. 종료  
선택 : 2  
  
전화번호부 목록 (이름순):  
이름 : 김경렬 , 전화번호 : 010-1111-1111  
  
1. 연락처 추가  
2. 연락처 보기

전화번호부 목록에 입력한 이름과 전화번호가 제대로 입력되어 출력되었다. 이제 3을 입력하여 연락처 검색을 해보겠다.

6. 종료  
선택 : 3  
검색할 이름을 입력하세요 : 김경렬  
이름 : 김경렬 , 전화번호 : 010-1111-1111  
이름 : 김경렬 , 전화번호 : 010-2222-2222  
  
1. 연락처 추가

3을 입력하니 이전에 입력해둔 같은 이름의 전화번호가 모두 출력이 되는 것을 볼 수 있다. 이제 4를 입력하여 연락처 수정을 해보겠다.

선택 : 4  
수정할 연락처의 이름을 입력하세요 : 김경렬  
1. 이름 : 김경렬, 전화번호 : 010-1111-1111  
2. 이름 : 김경렬, 전화번호 : 010-2222-2222  
수정할 연락처의 번호를 입력하세요 (1 ~ 2): 2  
새로운 이름을 입력하세요 (현재 : 김경렬): 임형근  
새로운 전화번호를 입력하세요 (현재 : 010-2222-2222): 010-3333-3333  
연락처가 수정되었습니다.

1. 연락처 추가
  2. 연락처 보기
  3. 연락처 검색
  4. 연락처 수정
  5. 연락처 삭제
  6. 종료
- 선택 : 2

전화번호부 목록 (이름순):  
이름 : 김경렬, 전화번호 : 010-1111-1111  
이름 : 임형근, 전화번호 : 010-3333-3333

처음 의도 했던 대로 동명이인인 전화번호가 여러개 있을 경우 하나를 골라 수정하는 기능이 문제 없이 실행되는 모습이다.

이제 5를 입력해 연락처 삭제를 해보겠다.

선택 : 5  
삭제할 연락처의 이름을 입력하세요 : 임형근  
1. 이름 : 임형근, 전화번호 : 010-3333-3333  
삭제할 연락처의 번호를 입력하세요 (1 ~ 1): 1  
연락처가 삭제되었습니다.

1. 연락처 추가
  2. 연락처 보기
  3. 연락처 검색
  4. 연락처 수정
  5. 연락처 삭제
  6. 종료
- 선택 : 2

전화번호부 목록 (이름순):  
이름 : 김경렬, 전화번호 : 010-1111-1111

1. 연락처 추가

삭제할 연락처의 이름을 입력받고 동명이인이 있을 경우 이용자가 원하는 연락처를 삭제할 수 있도록 하나를 골라서 삭제하는 기능이 제대로 작동하는 모습이다. 이제 마지막으로 6을 입력하고 프로그램을 종료해보겠다.

```
3. 연락처 검색
4. 연락처 수정
5. 연락처 삭제
6. 종료
선택 : 6
프로그램을 종료합니다.

C:\Users\82105\source\repos\ConsoleApplication1\x64\
계 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요 ...|
```

6을 입력하자 ‘프로그램을 종료합니다.’라는 메시지가 출력되고 프로그램이 정상적으로 종료되었다.

## -문제 발생 및 해결

초기에 코드를 짤 때 배열의 크기를 1로 생성하고 배열을 \*2를 하여 늘리도록 코드를 작성 하였지만 코드를 진행하다 보니 메모리 할당 오류가 발생하였다. 이 문제를 해결하기 위해 메모리 할당 오류 발생 시 메모리 할당 오류가 발생하였다는 메시지가 출력 되도록 코드를 수정하고 다시 실행을 해보았지만 메모리 할당 오류가 발생했다는 메시지는 잘 출력이 되 지만 코드가 실행이 되진 않았다. 이를 해결하기 위해 문제점을 찾던 도중 배열의 초기 값 이 문제였다는 사실을 알게 되었다. 배열의 초기 값을 1로 잡은 뒤 계속해서 \*2를 해주니 배열이 늘어나는 빈도가 너무 많았고 때문에 메모리 할당 오류가 발생하였던 것이었다.

이를 해결 하기 위해 배열이 늘어나는 빈도를 줄여야 됐고 배열의 초기 값을 10으로 수정하 고 코드를 진행하니 오류가 해결이 되었다.

또한 메뉴를 선택할 때 숫자가 아닌 문자를 입력하는 경우 “다시 입력하세요.” 라는 메시 지가 무한으로 출력되는 오류가 있었다.

해결방안을 찾던 도중 cin에서 입력 받을 때 원치 않는 값을 입력받더라도 입력 버퍼에 해 당 값이 남아 있다는 사실을 알게 되었다.

이를 해결하기 위해 cin.fail를 사용해 원치 않는 값을 받은 상황에서

cin.clear을 사용해 cin이 다시 작동할 수 있도록 해줬고

cin.ignore((numeric\_limits<streamsize>>::max())를 사용해 입력 버퍼에 남아있던 값을 모두

비워 오류가 더 이상 진행되지 않도록 처리를 하였다.

