**노래관리프로그램(가제)**

**A03 팀**

**초기 기획서 원판**

**201911573 이원준**

**202111310 손원희**

**각자 학번 이름 추가해주세요.**

프로그램 설명 추가.

**목 차**

**1. 용어**   **2**

**2. 기본사항**   **3**

2.1 작동 환경 3

2.2 프로그램 구성 3

2.3 프로그램 실행 3

**3. 프로그램 사용 흐름도**  **4**

**4. 데이터 요소** (수정)  **4**

4.1 노래 데이터 파일

4.1.1 제목

4.1.2 가수

4.1.3 재생시간

4.1.4 작곡가

4.1.5 작사자

4.1.6 장르

4.1.7 앨범명

4.1.8 앨범발매연도

4.2 4

4.3 5

**5. 무결성 확인 및 처리**   **5**

**6. 주 프롬프트**  **5**

6.1 초기메뉴 5

6.2 노래 리스트 관리 6

6.2.1 노래 추가

6.2.1.1 제목

6.2.1.2 가수

6.2.1.2.1 가수 - 동명이인

6.2.1.3 재생시간

6.2.1.4 작곡가  
 6.2.1.2.1 작곡가 - 동명이인

6.2.1.5 작사가

6.2.1.2.1 작사가 - 동명이인

6.2.1.6 장르

6.2.1.7 앨범명

6.2.1.8 앨범발매연도 6

6.2.2 노래 리스트 출력 6

6.2.3 노래 삭제 7

6.3검색 기능 9

6.3.1 통합검색 9

6.3.2 태그검색 9

6.4 플레이리스트 10

6.4.1 현재 플레이리스트 출력 11

6.4.2 플레이리스트 추가 11

6.4.3 플레이리스트 삭제 11

6.5 종료  **12**

**1. 용어**

이 문서에서 사용할 용어들의 의미를 약속하고, 아울러 일부 용어들과 관련하여 이 문서에서 사용할 표기법도 함께 약속해둡니다.

**프로그램** : 이 문서를 통해 기획, 명시하고 있는 대상 프로그램인 <노래관리프로그램>

**주 프롬프트** : 컴퓨터 운영 체제에서 사용되는 인터페이스 중 하나로, 사용자가 컴퓨터에 명령어를 입력하고 실행할 수 있는 환경을 제공합니다. 일반적으로 명령 프롬프트(command prompt) 또는 콘솔(console)이라고도 불림

**데이터 파일**(혹은 **파일**) : <노래관리프로그램>이 메뉴 정보를 저장하고 읽어 들이는 텍스트 형식의 파일

**숫자** : (서)아라비아 숫자들 중 표준 키보드로 직접 입력할 수 있는 10개(U+0030 ‘0’ ~ U+0039’9’)만을 뜻합니다. 즉, 이 문서에서 말하는 “숫자”에는 아라비아 숫자가 아닌 로마 숫자나 각 언어별 고유 숫자 기호들은 포함되지 않으며, 아라비아 숫자들 중에서도 전각 숫자, 원⦁괄호로 둘러싸인 숫자, 위⦁아래 첨자용 숫자, 수식용 글꼴별 숫자 등은 역시 모두 제외됩니다.

**탭** : 이 문서에서의 탭은 표준 키보드의 Tab키로 입력할 수 있는 문자(U+0009 Character Tabulation)만을 의미합니다. 이 문서에서 탭을 시각적으로 명확히 표시해야할 때에는 ‘→’ 기호로 표시합니다. 이 기호는 위치에 따라 화살대의 길이가 다를 수 있지만, 화살대의 길이가 아무리 길더라도 화살촉(과 그 끝의 세로 선)이 한 개면 무조건 한 개의 탭 문자를 나타냅니다.

**표준공백** : 표준 키보드의 Space bar키로 직접 입력할 수 있는 문자(U+0020’ ‘Space). 이 문서에서 표준공백을 시각적으로 명확히 구분해야할 때에는 그 자리에 ’⊔‘ 기호를 써서 표시합니다.

**공백** : 이 문서에서의 공백은 표준공백, 탭만을 의미합니다.

**공백열0** : 0개 이상의 공백으로 이루어진 것을 의미합니다.

**공백열1**: 1개 이상의 공백으로 이루어진 것을 의미합니다.

**문자** : 이 문서에서의 문자는 아스키코드 48번~57번(0~9), 65번~90번(A~Z), 97번~122번(a~z)만을 의미합니다. 즉, 특수기호 및 공백, 한글은 문자로 인정하지 않습니다.

**문자열 :** 1개 이상의 문자로 이루어진 것을 의미합니다.

**(문자열의)** **길이** : 문자열의 길이는 그 속에 들어있는 문자들의 실제 개수로만 셉니다. 예를 들어, “12452”와 “Hello”의 길이는 똑같이 5입니다.

**명령어**(혹은 **명령**) : 사용자가 주 프롬프트에 입력한 단어가 특정한 위치에 등장하면, 프로그램이 이를 특정한 의미로 간주하도록 예약된 문자열들.

**2. 기본사항**

**2.1 작동 환경**

본 프로그램은 C로 구현한 프로그램으로서 추가로 설치해야 할 인터프리터나 가상머신과 같은 프로그램을 필요로 하지 않습니다.

본 프로그램은 데이터 파일을 UTF-8로 인코딩하였기 때문에 UTF-8을 출력할 수 있는 터미널을 필요로 합니다.

MS Windows 10은 본 프로그램의 실행요건을 충족시키고, 그 외의 버전에서 어쩌면 이 요건을 충족시킬 수도 있지만 보장되지는 않습니다. macOS와 Linux는 exe파일을 실행시킬 수 있는 별도의 방법을 필요로 할 수 있습니다. 다만, 그 방법을 통해 작동이 될 것이라고 확실히 보장되지는 않습니다.

**2.2 프로그램 구성 (수정)**

프로그램이 배포될 때는 주 실행 파일인 kiosk.exe 파일과 데이터 파일인 menu.txt 파일이 같이 배포됩니다.

menu.txt 파일은 메뉴의 정보가 저장되어 있습니다.

menu.txt 파일의 경우 사용자가 삭제하더라도 프로그램상에서 회원가입 과정에서 자동으로 다시 생성합니다. 다만, 그러면 유저 정보가 사라지거나 프로그램상 오류가 발생할 수 있으므로 추천하지는 않습니다.

주어진 데이터 파일을 사용자가 임의로 수정하거나 삭제할 시 프로그램을 구동하는 데 오류가 발생할 수 있습니다.

**2.3 프로그램 실행 및 주의사항**

본 프로그램은 별도의 설치과정 없이 프로그램을 실행할 수 있습니다. 다만 Windows환경을 제외한 Linux와 macOS의 경우 운영체제 특성상 실행이 불가능할 수 있습니다.

**3. 프로그램 사용 흐름도**

프로그램 사용 흐름, 즉, 프로그램 사용자 관점에서 프로그램을 사용하는 단계별 경로는 다음과 같습니다.

(흐름도 추가)

**그림1. 사용 흐름도1**

**4. 데이터 요소**

데이터 요소에는 크게 카테고리, 메뉴명, 금액이 있습니다.

각 요소마다 동치 비교 규칙도 서로 다르고 검색 시의 검색어 합치 규칙도 서로 다르므로, 아래의 각 요소별 소절에서 해당 요소의 동치 비교 규칙, 검색어 합치 규칙을 각각 명시하겠습니다. 단, 정렬 시의 순서 비교 방법은 모든 데이터 요소들이 공통적으로 ‘파일에 저장된 문자열 그대로, 사전식 순서로’ 비교됩니다.

**4.1. 노래 데이터 파일**

노래 데이터 파일에는 크게 제목, 가수, 재생시간, 작곡가, 작사자, 장르, 앨범명, 앨범 발매 연도가 있습니다.

**4.1.1 제목**

제목은 노래 리스트에 들어갈 노래 제목입니다.

문법 형식: 문법적으로 올바른 제목은 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 1이상
* 첫 문자와 마지막 문자는 실상 문자
* 탭도 개행도 전혀 들어있지 않음

의미 규칙: 제목에는 아무런 추가적인 의미규칙도 없습니다.

**[노래 제목 내에 띄어쓰기 규칙]**

동치 비교: 두 제목 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 제목으로 간주합니다.

**4.1.2 가수**

가수는 노래 리스트에 들어갈 가수 이름입니다.

문법 형식: 문법적으로 올바른 가수는 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 1이상
* 첫 문자와 마지막 문자는 실상 문자
* 탭도 개행도 전혀 들어있지 않음

의미 규칙: 가수에는 아무런 추가적인 의미규칙도 없습니다.

**[가수 이름 내에 띄어쓰기 규칙]**

동치 비교: 두 가수 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 가수로 간주합니다.

**4.1.3 재생시간**

재생시간은 노래 리스트에 들어갈 노래의 총 재생시간입니다.

문법 형식: 문법적으로 올바른 재생시간은 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 1이상
* 공백류는 전혀 들어있지 않음
* 문자열 내에 “분” 또는 “초” 또는 “분”과 ”초”가 포함되어 있어야 함

즉, 올바른 문법 형식은 다음과 같습니다.

* <숫자열> <분>
* <숫자열> <초>
* <숫자열> <분> <숫자열> <초>

예를 들어, “3분”이나 “3초”, “3분3초”는 모두 올바른 재생시간이지만, “3분␣3초’”나 “3␣3초”, “3␣3초”, “3␣3”은 올바른 재생시간이 아닙니다.

의미 규칙: <숫자열>의 선행 0을 제거한 수가 양의 정수가 아닌 경우 의미적으로 틀린 재생시간입니다.

또한, “초” 바로 앞에 있는 <숫자열>의 선행 0을 제거했을 때 60 이상인 경우 틀린 재생시간입니다.

**4.1.4 작곡가**

**작곡가는 노래 리스트에 들어갈 작곡가의 이름입니다.**

문법 형식: 문법적으로 올바른 작곡가는 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 1이상
* 첫 문자와 마지막 문자는 실상 문자
* 탭도 개행도 전혀 들어있지 않음

의미 규칙: 작곡가에는 아무런 추가적인 의미규칙도 없습니다.

**[작곡가 이름 내에 띄어쓰기 규칙]**

동치 비교: 두 작곡가 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 작곡가로 간주합니다.

**4.1.5 작사자**

작사자는 노래 리스트에 들어갈 작사자의 이름입니다.

문법 형식: 문법적으로 올바른 작사자는 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 1이상
* 첫 문자와 마지막 문자는 실상 문자
* 탭도 개행도 전혀 들어있지 않음

의미 규칙: 작사자에는 아무런 추가적인 의미규칙도 없습니다.

**[작사자 이름 내에 띄어쓰기 규칙]**

동치 비교: 두 작사자 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 작사자로 간주합니다.

**4.1.6 장르**

장르는 노래 리스트에 들어갈 노래의 장르입니다.

문법 형식: 문법적으로 올바른 장르는 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 0이상(즉, 빈 문자열도 가능)
* (길이가 1이상일 경우에 한해)첫 문자와 마지막 문자는 실상 문자
* 탭도 개행도 전혀 들어있지 않음

의미 규칙: 장르에는 아무런 추가적인 의미규칙도 없습니다.

**[장르 내에 띄어쓰기 규칙]**

동치 비교: 두 장르 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 장르로 간주합니다.

**4.1.7 앨범명**

앨범명은 노래 리스트에 들어갈 앨범의 이름입니다.

문법 형식: 문법적으로 올바른 앨범명은 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 0이상(즉, 빈 문자열도 가능)
* (길이가 1이상일 경우에 한해)첫 문자와 마지막 문자는 실상 문자
* 탭도 개행도 전혀 들어있지 않음

의미 규칙: 앨범명에는 아무런 추가적인 의미규칙도 없습니다.

**[앨범명 내에 띄어쓰기 규칙]**

동치 비교: 두 앨범명 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 앨범명으로 간주합니다.

**4.1.8 앨범 발매 연도**

앨범 발매 연도는 노래 리스트에 들어갈 앨범의 발매연도입니다.

문법 형식: 문법적으로 올바른 앨범발행연도는 아래 세 조건을 모두 만족시키는 문자열입니다.

* 길이가 0이상(즉, 빈 문자열도 가능)
* (길이가 1이상일 경우에 한해) 문자열 내에 <숫자열>과 “년”이 포함되어 있어야 함
* 탭도 개행도 전혀 들어있지 않음

즉, 올바른 형식은 다음과 같습니다.

* <빈 문자열>
* <숫자열> <년>

예를 들어, “2025년”이나 “1998년”은 올바른 앨범발매연도이지만, “2025␣년”, “19␣98년”은 올바른 앨범발매연도가 아닙니다.

의미 규칙: <숫자열>의 선행 0이 포함된 경우 선행0을 제거한 수가 1990보다 작거나 2024보다 크면 의미적으로 틀린 앨범발행연도입니다. 즉, “2025년”은 의미적으로 틀린 앨범발행연도입니다.

**[발행연도 내에 규칙]**

동치 비교: 두 앨범발행연도 간의 동치 비교를 할 때에는 두 문자열 전체가 일치해야만 같은 앨범발행연도로 간주합니다.

**4.2. 메뉴명**

메뉴명은 각 메뉴의 고유한 이름입니다.

**문법 형식:** 길이가 2~16인 문자열.

**의미 규칙:** 동일한 메뉴명은 같은 메뉴라는 의미이므로 중복될 수 없습니다.

**동치 비교:** 두 메뉴 간의 동치 비교를 할 때에는, 두 문자열 전체가 (영어의 대소문자 포함) 서로 완전히 일치해야만 같은 이름으로 간주합니다.

**4.3. 금액**

금액은 각 메뉴를 구매할 때 발생하는 비용입니다.

**문법 형식:** 1000 이상 1000000 이하인 정수

**의미 규칙:** 금액은 각 메뉴마다 갖고 있으나 고유한 값은 아니므로 중복될 수 있습니다.

**동치 비교:** 두 금액 간의 동치 비교는 한 금액에서 다른 한 금액을 뺐을 때의 값이 0일 때 같은 금액으로 간주합니다.

**5. 무결성 확인 및 처리**

프로그램이 실행될 때 다음과 같이 데이터 무결성을 확인하고 자동 수정 가능한 사항들을 처리하거나 치명적 오류를 알리고 프로그램이 종료됩니다.

**1.** 프로그램과 같은 경로에 메뉴 정보가 담긴 menu.txt 파일이 있는지를 확인합니다. menu.txt 파일이 같은 경로에 존재하면 문제가 없다는 뜻이므로 아무 출력 없이 다음 단계로 넘어가고, 파일이 존재하지 않으면 비어있는 menu.txt 파일을 프로그램과 같은 경로에 새롭게 생성하면서 다음과 같은 메시지를 출력하고 다음 단계로 넘어갑니다.



**2.** 이 단계까지 오면 무결성 확인 및 처리가 완료된 것이고, 6.1절의 초기 메뉴 주 프롬프트 화면을 출력하고 대기합니다.

**6. 주 프롬프트**

**6.1 초기메뉴**

프로그램 실행 시 처음으로 출력되는 주 프롬프트의 화면입니다.

**[초기 메뉴 선택 그림]**

사용자는 노래 리스트 관리, 검색 기능, 플레이리스트 3개의 메뉴 중 하나를 택할 수 있습니다.

문법 형식: 메뉴를 선택하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <명령어> <공백열0>

<명령어>에 올 수 있는 올바른 단어는 표1과 같습니다.

|  |  |  |
| --- | --- | --- |
| ① 노래 리스트 관리 | ② 검색 기능 | ③ 플레이리스트 |
| 1 | 2 | 3 |

표1: 올바른 명령어

명령어로 “1”,“2”,“3”중 하나의 숫자를 입력 받습니다. 사용자가 명령어에 해당하지 않는 입력을 할 경우 다음과 같이 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[오류가 발생한 경우 재출력된 선택 그림]**

의미 규칙 : 정해진 문법 형식 이외의 의미 규칙은 따로 존재하지 않습니다.

정상 결과: “1”을 입력할 시 6.2절, “2”를 입력할 시 6.3절, “3”을 입력할 시 6.4절로 넘어갑니다.

**6.2 노래 리스트 관리**

6.1 초기메뉴에서 ① 노래 리스트 관리를 선택하면 출력되는 화면입니다.

**[노래 리스트 관리 기능 선택 그림]**

사용자는 노래 추가, 노래 리스트 출력, 노래 삭제 3개의 메뉴 중 하나를 택할 수 있습니다.

문법 형식: 메뉴를 선택하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <명령어> <공백열0>

<명령어>에 올 수 있는 올바른 단어는 표2와 같습니다.

|  |  |  |
| --- | --- | --- |
| ① 노래 추가 | ② 노래 리스트 출력 | ③ 노래 삭제 |
| 1 | 2 | 3 |

표2: 올바른 명령어

명령어로 “1”,“2”,“3”중 하나의 숫자를 입력 받습니다. 사용자가 명령어에 해당하지 않는 입력을 할 경우 다음과 같이 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[오류가 발생한 경우 재출력된 선택 그림]**

의미 규칙 : 정해진 문법 형식 이외의 의미 규칙은 따로 존재하지 않습니다.

정상 결과: “1”을 입력할 시 6.2.1절, “2”를 입력할 시 6.2.2절, “3”을 입력할 시 6.2.3절로 넘어갑니다.

**6.2.1 노래 추가**

사용자에게 앞으로 나올 6.2.1.1절부터 6.2.1.8절까지의 입력을 받아 모두 오류가 없다면 노래 데이터 파일을 추가합니다.

**6.2.1.1 제목**

**[“제목을 입력하세요.”]**

추가하길 원하는 노래 제목을 입력 받습니다.

문법 형식: 제목을 입력하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <제목> <공백열0>

제목은 4.1.1절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“제목 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 제목에는 아무런 추가적인 의미규칙도 없습니다.

정상 결과: 문법 형식에 어긋나지 않는 입력을 받을 경우 6.2.1.2절로 넘어갑니다.

**6.2.1.2 가수**

**[“가수를 입력하세요.”]**

추가하길 원하는 노래의 가수를 입력 받습니다.

문법 형식: 제목을 입력하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <가수> <공백열0>

이때 만약 가수가 한 명이 아니라 여러 명일 경우 표준 키보드에 있는 반점인 ‘,’(U+002C)을 이용하여 구분하여

여러 명의 가수를 입력 받습니다. 예를 들어, 가수가 3명일 경우 문법 형식은 다음과 같습니다.

* <공백열0> <가수> <공백열0> <,> <공백열0> <가수> <공백열0> <,> <공백열0> <가수> <공백열0>

가수는 4.1.2절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“가수 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 만약 기존의 노래 데이터 파일에 동일한 가수가 있다면 6.2.1.2.1절로 넘어갑니다.

정상 결과: 문법 형식과 의미 규칙에 어긋나지 않는 입력을 받을 경우 6.2.1.3절로 넘어갑니다.

**6.2.1.2.1 가수 - 동명이인 가수**

다음과 같은 문구를 출력한다.

**[“동일한 가수가 이미 존재합니다. 기존의 가수와 동일합니까?”+’기존의 노래 제목과 가수’ + 선택 그림]**

사용자는 예와 아니요 중 하나를 택할 수 있습니다.

문법 형식: 메뉴를 선택하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <명령어> <공백열0>

<명령어>에 올 수 있는 올바른 단어는 표3과 같습니다.

|  |  |
| --- | --- |
| ① 예 | ② 아니요 |
| 1 | 2 |

표3: 올바른 명령어

명령어로 “1”과 “2” 중 하나의 숫자를 입력 받습니다. 사용자가 명령어에 해당하지 않는 입력을 할 경우 다음과 같이 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“해당 선택지는 잘못되었습니다. 정확히 입력해주세요.” + “1. 예 2. 아니요”]**

의미 규칙: 정해진 문법 형식 이외의 의미 규칙은 따로 존재하지 않습니다.

정상 결과: 문법 형식에 올바른 입력을 하면 6.2.1.3절로 넘어갑니다.

**6.2.1.3 재생시간**

**[“재생시간을 입력하세요.”]**

추가하길 원하는 노래의 재생시간을 입력 받습니다.

문법 형식: 재생시간을 입력하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <재생시간> <공백열0>

제목은 4.1.3절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“재생시간 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 재생시간에는 아무런 추가적인 의미규칙도 없습니다.

정상 결과: 문법 형식에 어긋나지 않는 입력을 받을 경우 6.2.1.4절로 넘어갑니다.

**6.2.1.4 작곡가**

**[“작곡가를 입력하세요.”]**

추가하길 원하는 노래의 작곡가를 입력 받습니다.

문법 형식: 작곡가를 입력하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <작곡가> <공백열0>

이때 만약 작곡가가 한 명이 아니라 여러 명일 경우 표준 키보드에 있는 반점인 ‘,’(U+002C)을 이용하여 구분하여

여러 명의 작곡가를 입력 받습니다. 예를 들어, 작곡가가 3명일 경우 문법 형식은 다음과 같습니다.

* <공백열0> <작곡가> <공백열0> <,> <공백열0> <작곡가> <공백열0> <,> <공백열0> <작곡가> <공백열0>

작곡가는 4.1.4절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“작곡가 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 만약 기존의 노래 데이터 파일에 동일한 작곡가가 있다면 6.2.1.4.1절로 넘어갑니다.

정상 결과: 문법 형식과 의미 규칙에 어긋나지 않는 입력을 받을 경우 6.2.1.5절로 넘어갑니다.

**6.2.1.4.1 작곡가 - 동명이인 작곡가**

다음과 같은 문구를 출력한다.

**[“동일한 작곡가가 이미 존재합니다. 기존의 작곡가와 동일합니까?”+’기존의 노래 제목과 작곡가’ + 선택 그림]**

사용자는 예와 아니요 중 하나를 택할 수 있습니다.

문법 형식: 메뉴를 선택하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <명령어> <공백열0>

<명령어>에 올 수 있는 올바른 단어는 표4와 같습니다.

|  |  |
| --- | --- |
| ① 예 | ② 아니요 |
| 1 | 2 |

표4: 올바른 명령어

명령어로 “1”과 “2” 중 하나의 숫자를 입력 받습니다. 사용자가 명령어에 해당하지 않는 입력을 할 경우 다음과 같이 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“해당 선택지는 잘못되었습니다. 정확히 입력해주세요.” + “1. 예 2. 아니요”]**

의미 규칙: 정해진 문법 형식 이외의 의미 규칙은 따로 존재하지 않습니다.

정상 결과: 문법 형식에 올바른 입력을 하면 6.2.1.5절로 넘어갑니다.

**6.2.1.5 작사가**

**[“작사가를 입력하세요.”]**

추가하길 원하는 노래의 작사가를 입력 받습니다.

문법 형식: 작사가를 입력하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <작사가> <공백열0>

이때 만약 작사가가 한 명이 아니라 여러 명일 경우 표준 키보드에 있는 반점인 ‘,’(U+002C)을 이용하여 구분하여

여러 명의 작사가를 입력 받습니다. 예를 들어, 작사가가 3명일 경우 문법 형식은 다음과 같습니다.

* <공백열0> <작사가> <공백열0> <,> <공백열0> <작사가> <공백열0> <,> <공백열0> <작사가> <공백열0>

작사가는 4.1.5절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“작사가 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 만약 기존의 노래 데이터 파일에 동일한 작사가가 있다면 6.2.1.5.1절로 넘어갑니다.

정상 결과: 문법 형식과 의미 규칙에 어긋나지 않는 입력을 받을 경우 6.2.1.6절로 넘어갑니다.

**6.2.1.5.1 작사가 - 동명이인 작사가**

다음과 같은 문구를 출력한다.

**[“동일한 작사가가 이미 존재합니다. 기존의 작사가와 동일합니까?”+’기존의 노래 제목과 작사가’ + 선택 그림]**

사용자는 예와 아니요 중 하나를 택할 수 있습니다.

문법 형식: 메뉴를 선택하는 올바른 문법 형식은 다음과 같습니다.

* <공백열0> <명령어> <공백열0>

<명령어>에 올 수 있는 올바른 단어는 표5와 같습니다.

|  |  |
| --- | --- |
| ① 예 | ② 아니요 |
| 1 | 2 |

표5: 올바른 명령어

명령어로 “1”과 “2” 중 하나의 숫자를 입력 받습니다. 사용자가 명령어에 해당하지 않는 입력을 할 경우 다음과 같이 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“해당 선택지는 잘못되었습니다. 정확히 입력해주세요.” + “1. 예 2. 아니요”]**

의미 규칙: 정해진 문법 형식 이외의 의미 규칙은 따로 존재하지 않습니다.

정상 결과: 문법 형식에 올바른 입력을 하면 6.2.1.5절로 넘어갑니다.

**6.2.1.6 장르**

**[“장르를 입력하세요.”]**

추가하길 원하는 노래의 장르를 입력 받습니다.

문법 형식: 장르을 입력하는 올바른 문법 형식은 다음 중 하나와 같습니다.

* (<장르>의 길이가 0일 때) <공백열0>
* <공백열0> <장르> <공백열0>

제목은 4.1.6절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“장르 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 재생시간에는 아무런 추가적인 의미규칙도 없습니다.

정상 결과: 문법 형식에 어긋나지 않는 입력을 받을 경우 6.2.1.7절로 넘어갑니다.

**6.2.1.7 앨범명**

**[“앨범명를 입력하세요.”]**

추가하길 원하는 노래의 앨범명을 입력 받습니다.

문법 형식: 앨범명을 입력하는 올바른 문법 형식은 다음 중 하나와 같습니다.

* (<앨범명>의 길이가 0일 때) <공백열0>
* <공백열0> <앨범명> <공백열0>

앨범명은 4.1.7절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“앨범명 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 앨범명에는 아무런 추가적인 의미규칙도 없습니다.

정상 결과: 문법 형식에 어긋나지 않는 입력을 받을 경우 6.2.1.8절로 넘어갑니다.

**6.2.1.8 앨범발행연도**

**[“앨범발행연도를 입력하세요.”]**

추가하길 원하는 노래의 앨범발행연도를 입력 받습니다.

문법 형식: 앨범발행연도를 입력하는 올바른 문법 형식은 다음 중 하나와 같습니다.

* (<앨범발행연도>의 길이가 0일 때) <공백열0>
* <공백열0> <앨범발행연도> <공백열0>

앨범발행연도은 4.1.8절의 문법 형식에 맞게 문자열로만 입력 받습니다.

문법 형식에 어긋나는 입력을 할 경우 다음과 같은 오류 문구를 출력한 후 올바른 입력을 받을 때까지 사용자에게 재입력을 받습니다.

**[“앨범발행연도 입력이 잘못되었습니다. 정확히 입력해주세요.” + 올바른 예시]**

의미 규칙: 앨범발행연도에는 아무런 추가적인 의미규칙도 없습니다.

정상 결과: 문법 형식에 어긋나지 않는 입력을 받을 경우 초기 메뉴로 넘어갑니다.

**6.2.2 노래 리스트 출력**

**6.2.3 노래 삭제**

**6.3 검색 기능**

**6.3.1 통합 검색**

**6.3.2 태그 검색**

**6.4 플레이리스트**

**6.4.1 현재 플레이리스트 출력**

**6.4.2 플레이리스트 추가**

**6.4.3 플레이리스트 삭제**

**6.5 종료**

프로그램을 종료합니다. 6.1절에서 종료를 선택하거나 무결성 검사에서 결함이 발견되면 아래 문구를 출력하고 프로그램을 종료합니다.

