|  |
| --- |
| **C++프로그래밍및실습**  **리소스 관리 모듈**  **진척 보고서 #2** |

|  |  |
| --- | --- |
|  | 제출일자: 2023.12.03  제출자명: 김기태  제출자학번: 185482 |

**1. 프로젝트 목표**

**1) 배경 및 필요성**

개인적으로 렌파이라는 툴을 이용한 비주얼 노벨 개발 작업을 진행 중에 있다. 작업을 진행할 때 관리해야 할 리소스가 너무나도 많아 불편한 점이 많이 있었기 때문에 이를 보완하기 위한 **실사용 목적으로 개발**한다. 실제로 현재 개발중인 버전 기준으로 리소스 정의 코드 + 시나리오 스크립트가 함께 들어 있어서 만 줄 가량을 뛰어넘는 스크립트 문서의 분량 때문에 그때그때 필요한 부분을 제때 찾기에 불편하고 시간이 소요되었기 때문에 스크립트 문서를 분할하여 각 리소스 관리 부분을 담당하는 텍스트 파일을 따로 만들 필요성이 생겼기 때문이다. 또한 개발 중인 리소스의 용량이 4GB를 넘는 문제 때문에 완성 후 배포 시에 필요 없는 더미 데이터들을 식별하여 삭제하여서 용량을 최소화시켜야만 할 필요가 생겼기 때문이기도 하다.

**2) 프로젝트 목표**

첫째로는 외부 파일에 리소스 정의에 사용된 코드들을 자동적으로 저장해내는 것이다. 렌파이의 스크립트 기반인 ‘rpy’ 확장자의 텍스트 파일로 저장하도록 한다. 따라서 보유 리소스의 내용이 변동되어도 단 한 번의 툴 사용만으로 모든 리소스의 정의 내역이 현재 파일과 정확히 연동되도록 할 것이다.

둘째로는 외부 텍스트 파일을 읽어 들여서 해당 텍스트 파일에서 사용된 파일과 사용되지 않은 파일을 찾아내서 명단화하는 것이다. 여기서 읽기에 사용할 파일은 script.rpy로 이 파일의 텍스트를 읽어 들여서 만일 사용한 적이 있는 리소스라면 사용된 리소스의 목록을 담은 텍스트 파일에 분류하고, 사용한 적이 없는 리소스면 사용되지 않은 리소스 목록을 담은 텍스트 파일로 분류한다.

**3) 차별점**

기존에 리소스 정의 및 정리 용도로 사용하던 파이썬 기반 코드들은 사용자가 직접 스크립트 파일에 복사 붙여넣기를 해 주어야만 한다는 단점이 있었다. 따라서 기존 코드와는 달리 **C++로 재설계**함과 동시에 외부 파일에 결과값을 저장할 수 있도록 하며, 어떤 환경에서도 실행되도록 절대 경로를 상대경로로 변경하고, 기존 코드들이 배경음악, 사운드 이펙트, 스탠딩 그래픽, 배경 그래픽 등으로 나뉘어져 있었던 것을 한 파일 안에 통합환경으로 구현할 계획이다.

**2. 기능 계획**

**1) 리소스 정의 및 외부 파일에 자동 저장**

- 각자 서식에 맞는 리소스 정의를 개별 파일에 저장하고, 리소스 정의 파일을 각자 파일에 맞는 audio.rpy, scg.rpy, bg.rpy등의 파일에 저장한다. 종류별로 따로따로 저장함으로써 스크립트의 가독성을 높이고 관리를 더 쉽게 할 수 있다.

**2) 사용되지 않은 리소스 식별 기능**

사용되지 않은 리소스를 스크립트 파일에서 동일 문자열의 존재 여부를 확인하여 단 하나라도 존재하면 존재하는 리소스 목록에 등재하고, 존재하지 않으면 존재하지 않는 리소스 목록에 등재하여 txt파일로 내보내는 것이다. 이는 테스트 버전 내보내기 및 개발 완료 단계에서 더미 데이터의 최소화를 통한 저장공간 절약에 도움이 될 것이다.

**3. 진척사항**

**1) 기능 구현**

**(1) 사용되지 않은 리소스 식별 기능**

- 입출력

입력값은 game폴더 안에 있는 script.rpy이며, 출력값은 현재 상태로는 콘솔에 출력한다. 후일 콘솔뿐만 아니라 txt파일로 출력하도록 할 예정이다.

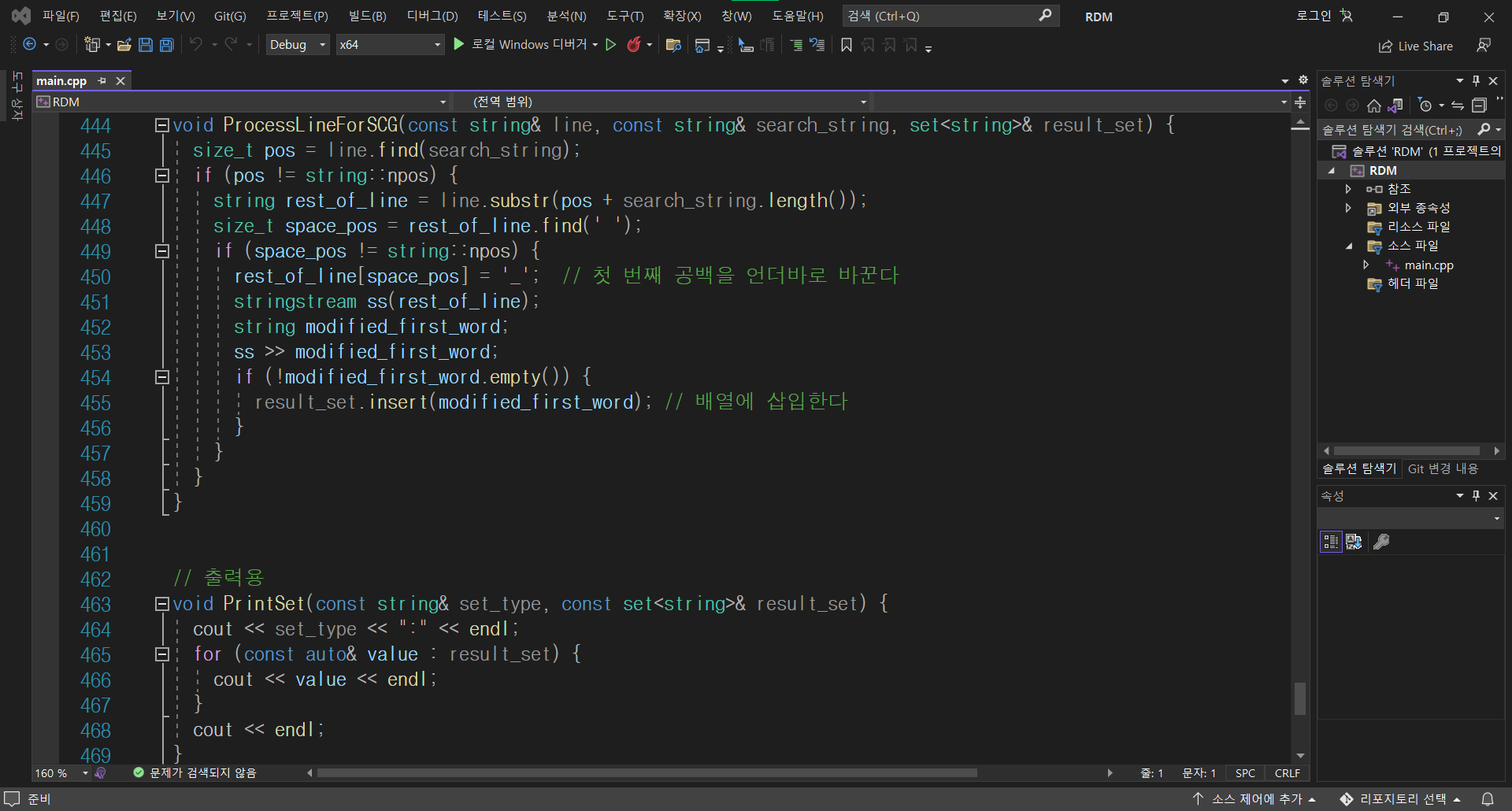
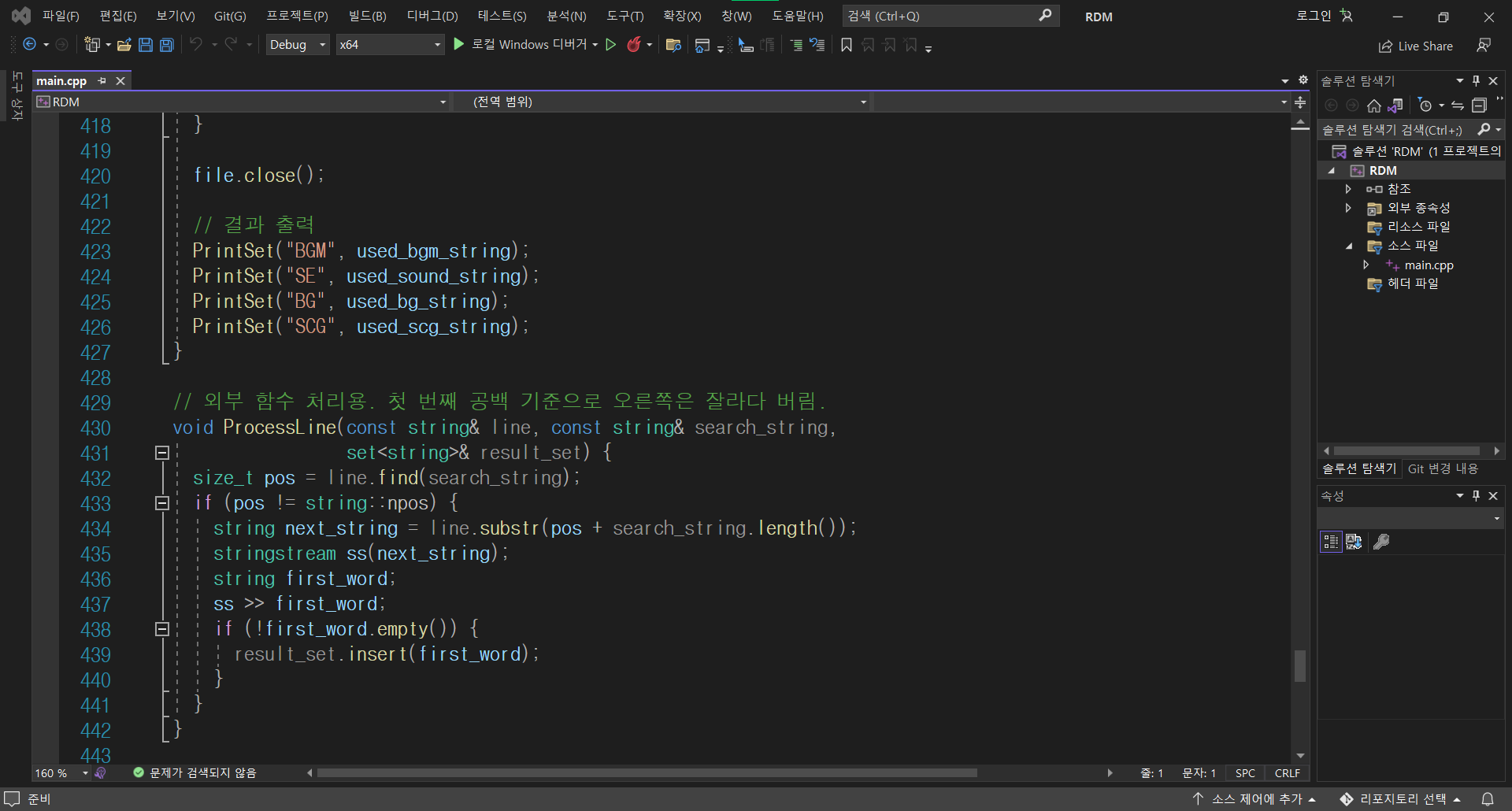
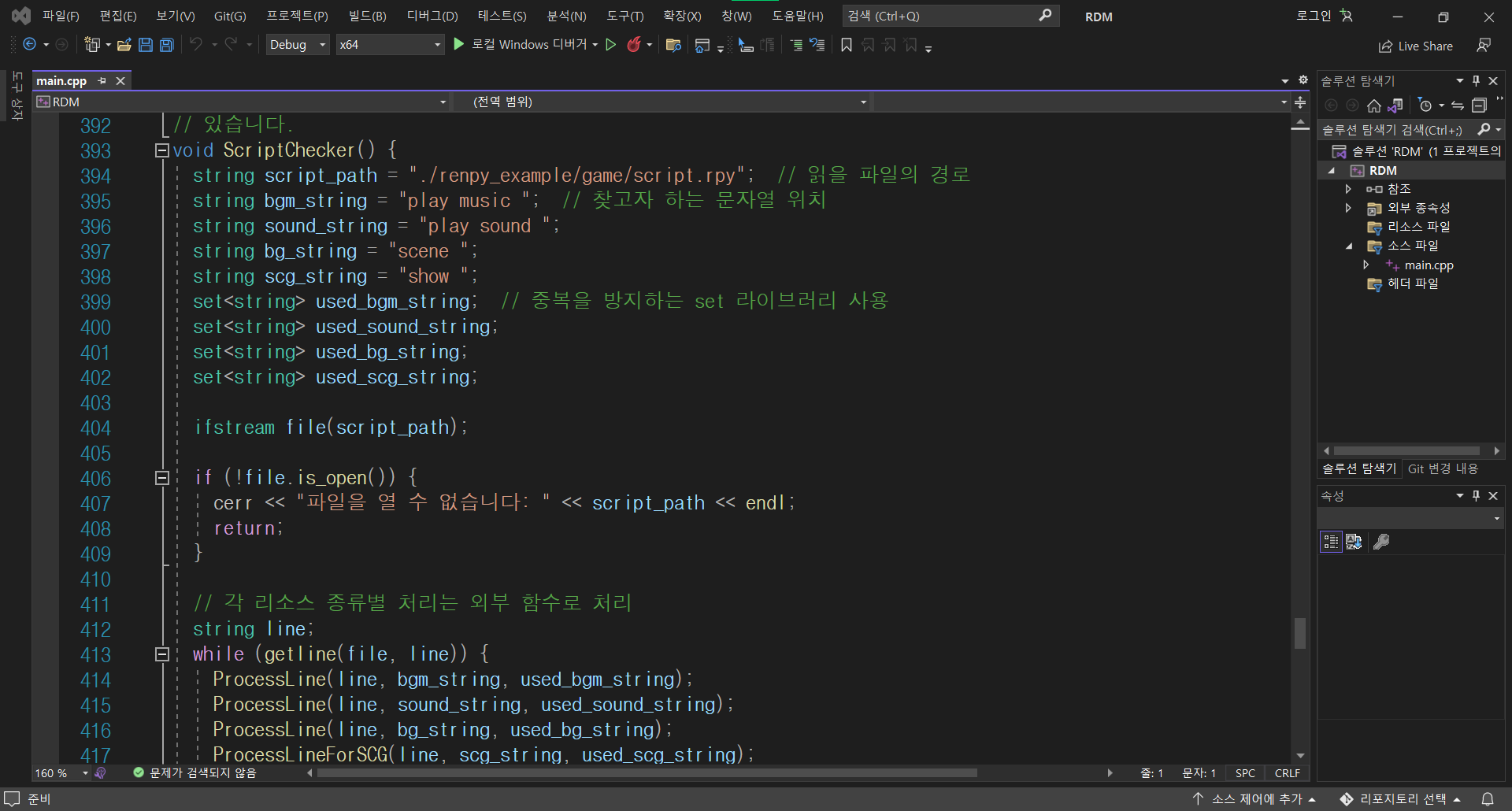
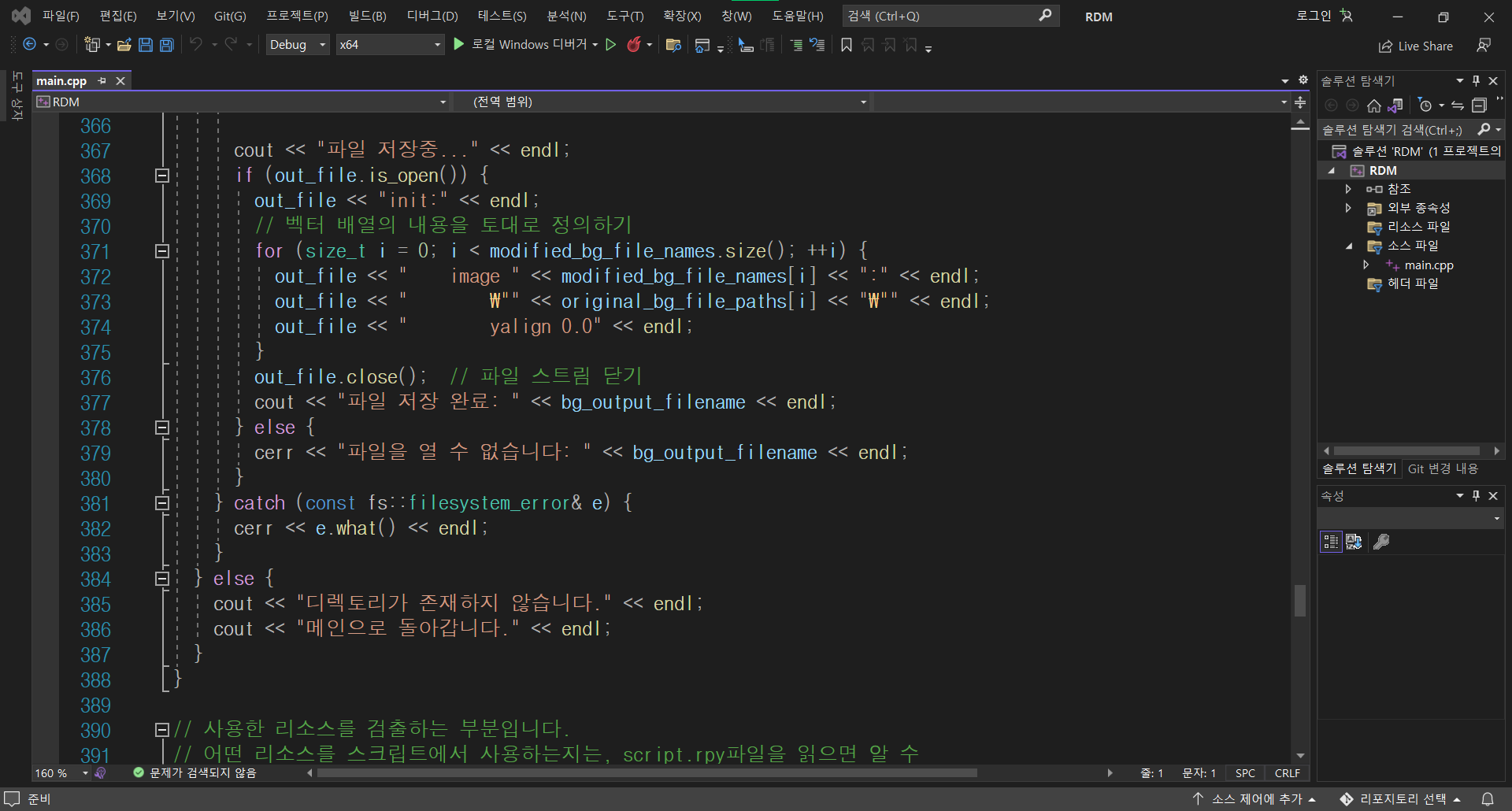
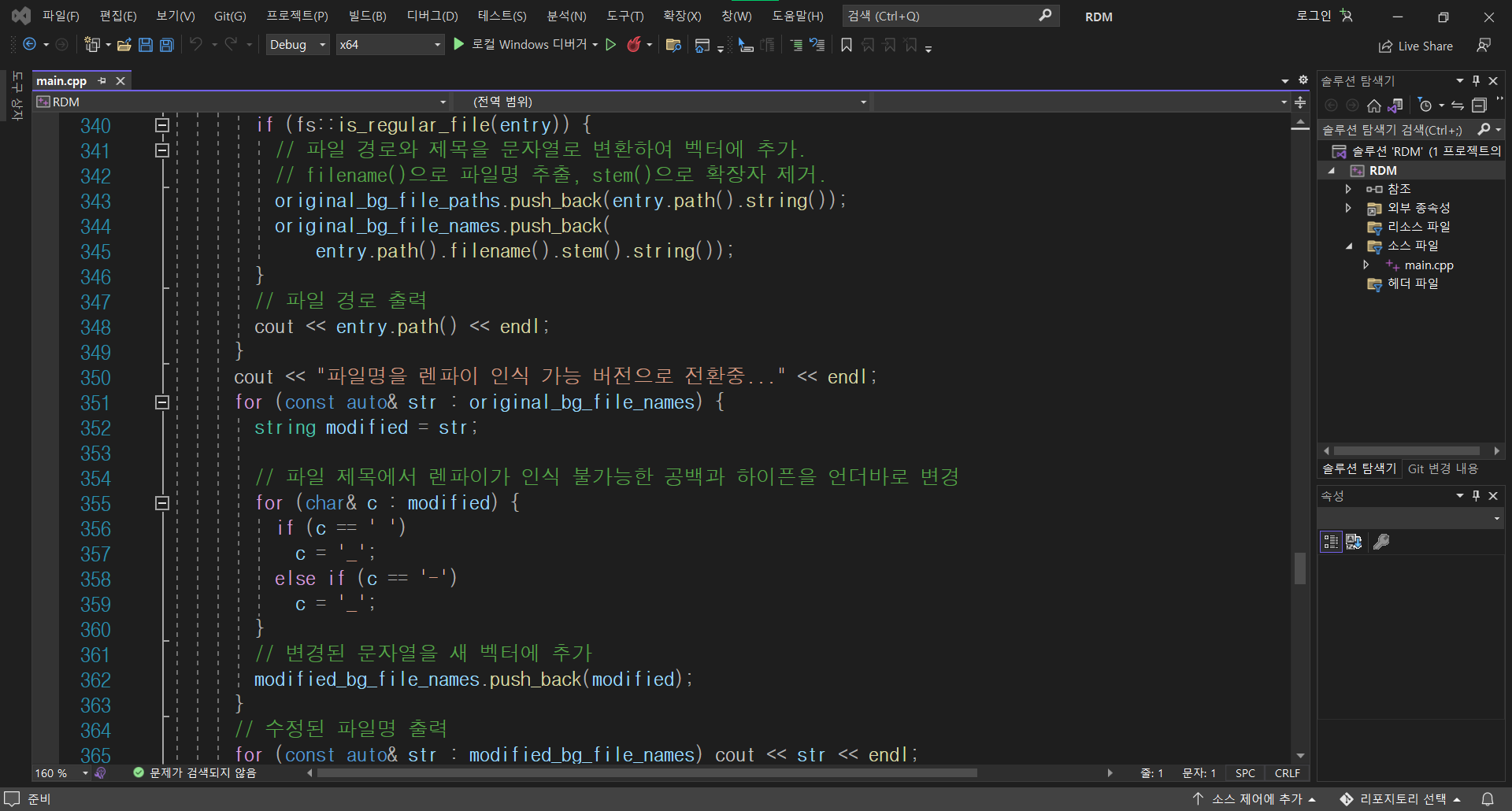
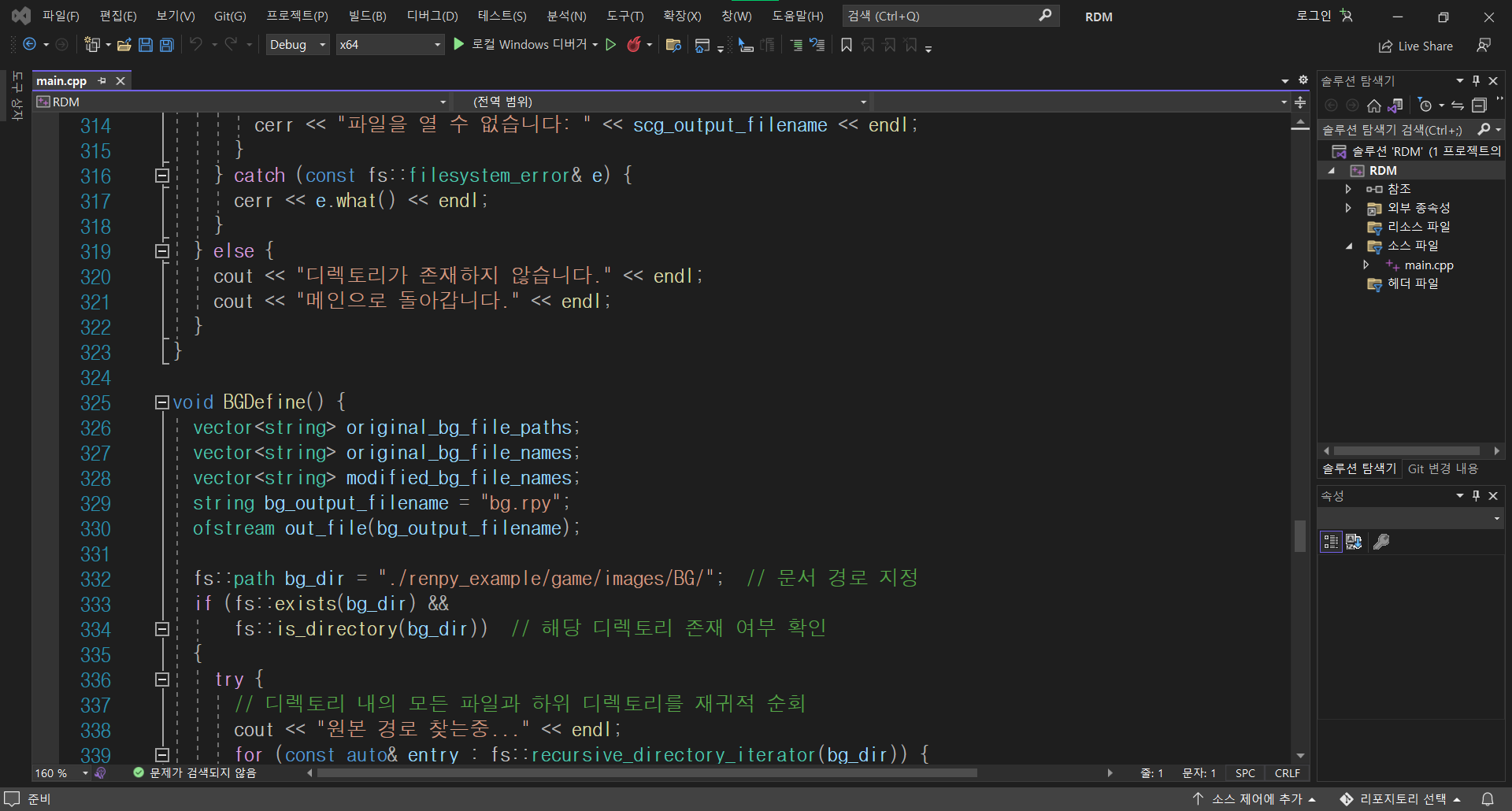
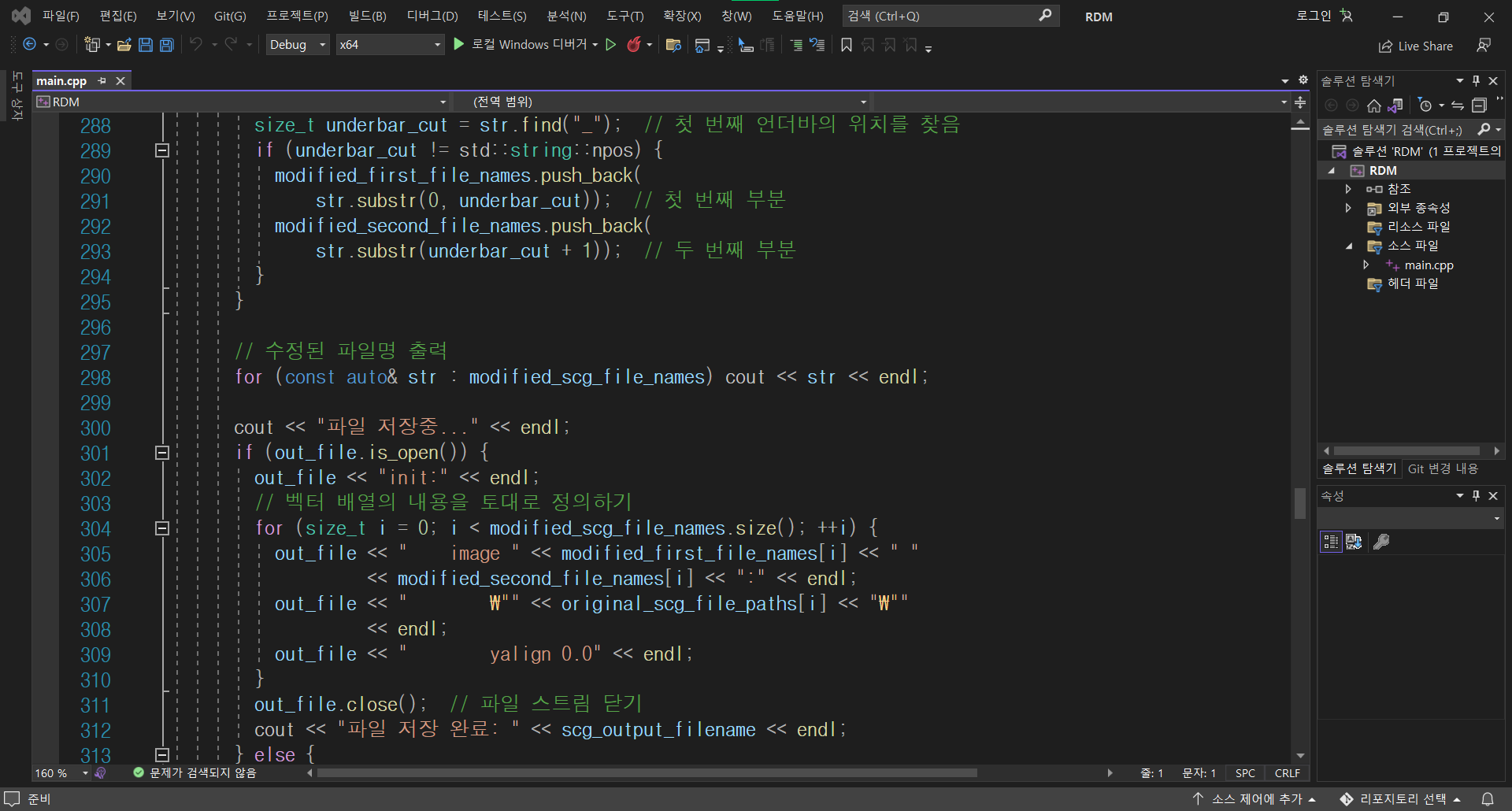
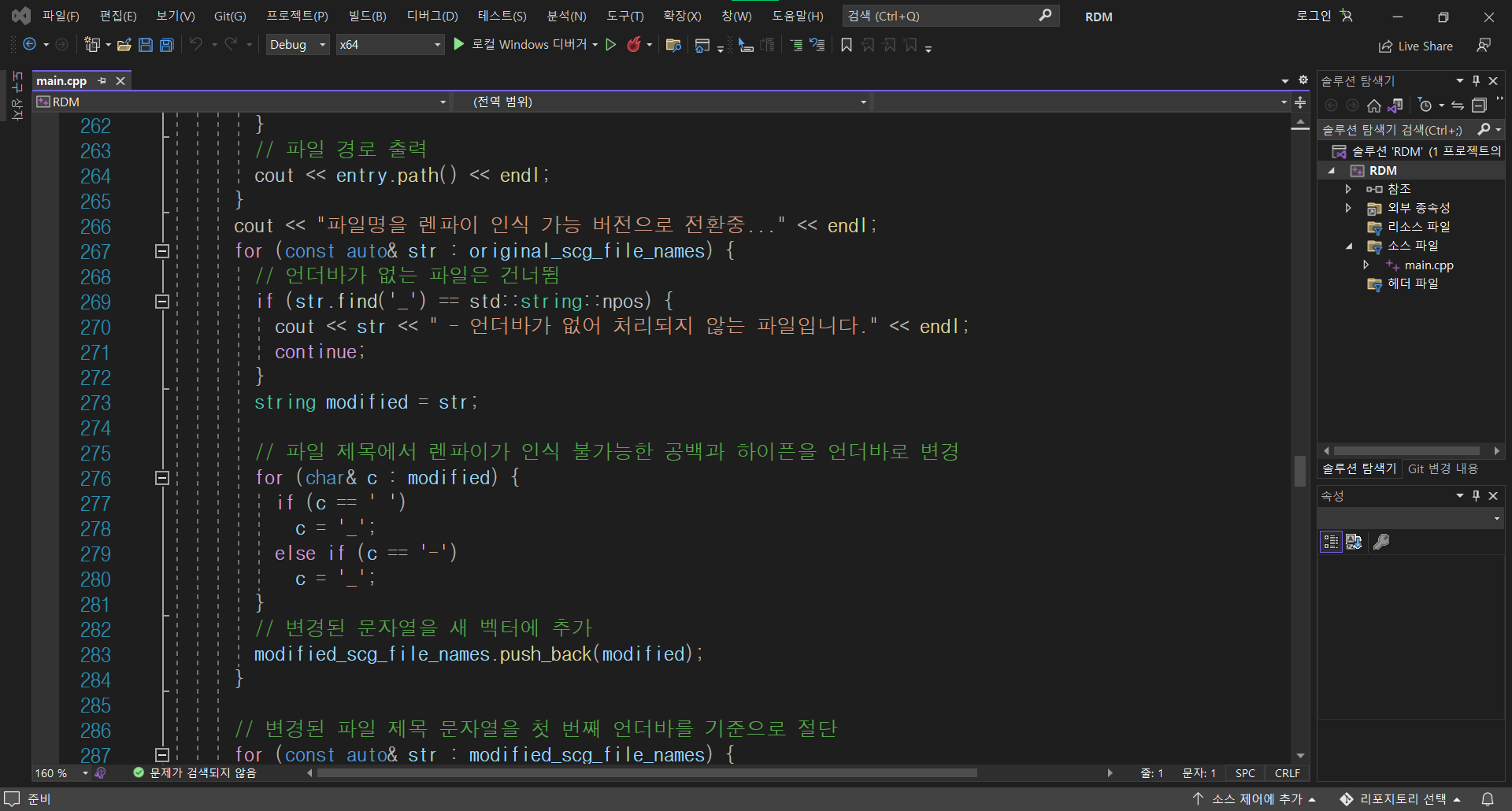
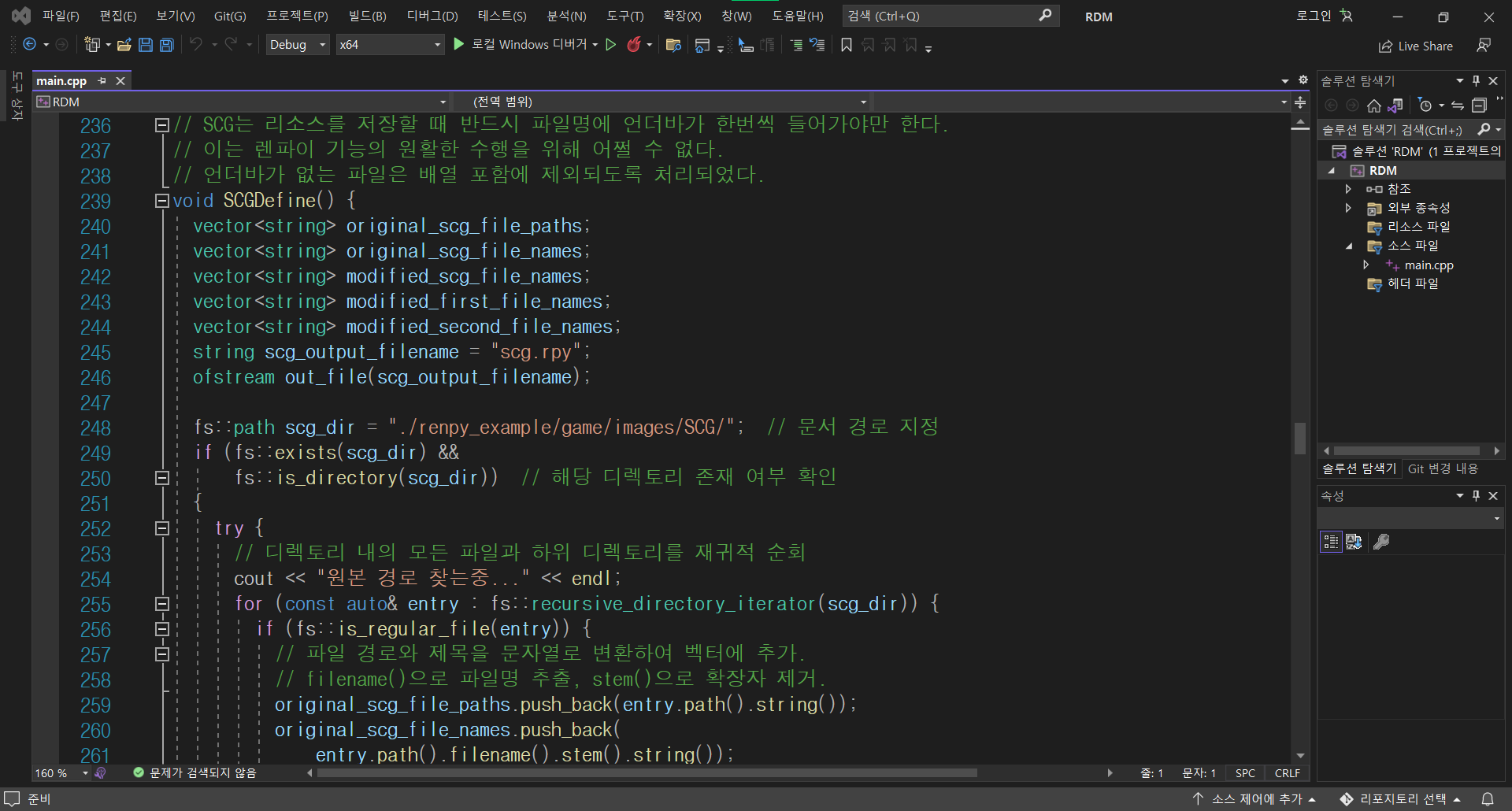
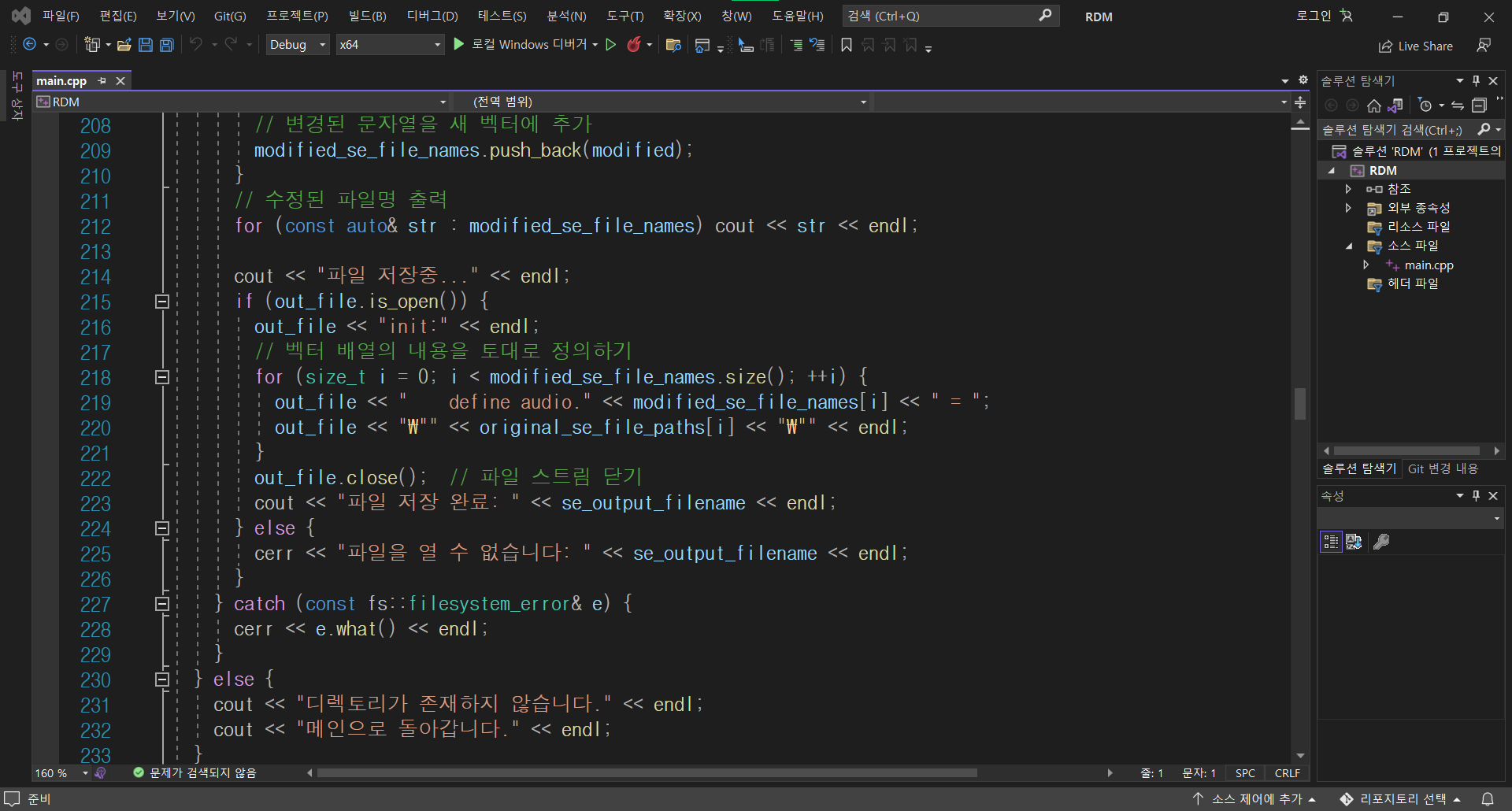
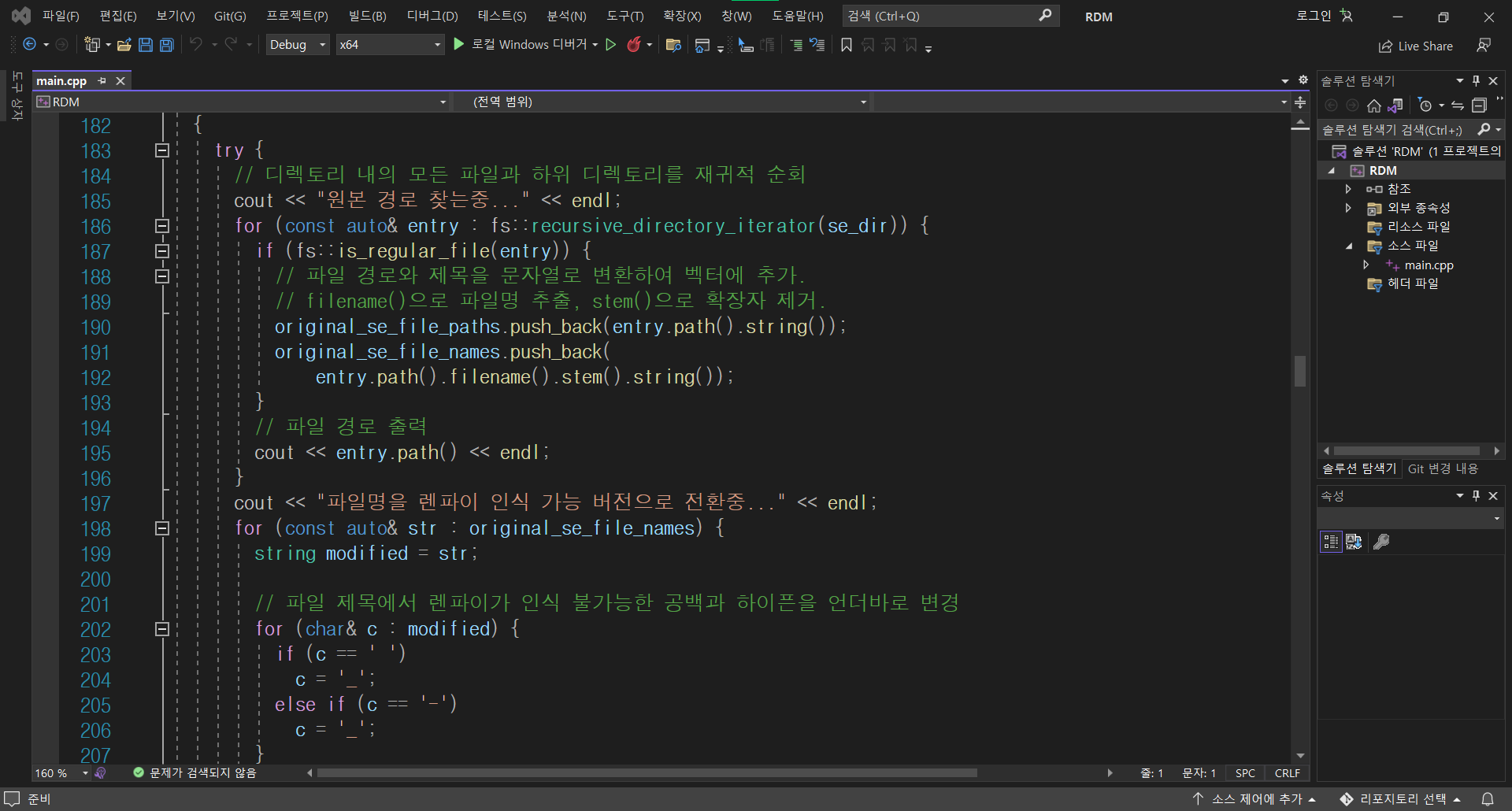
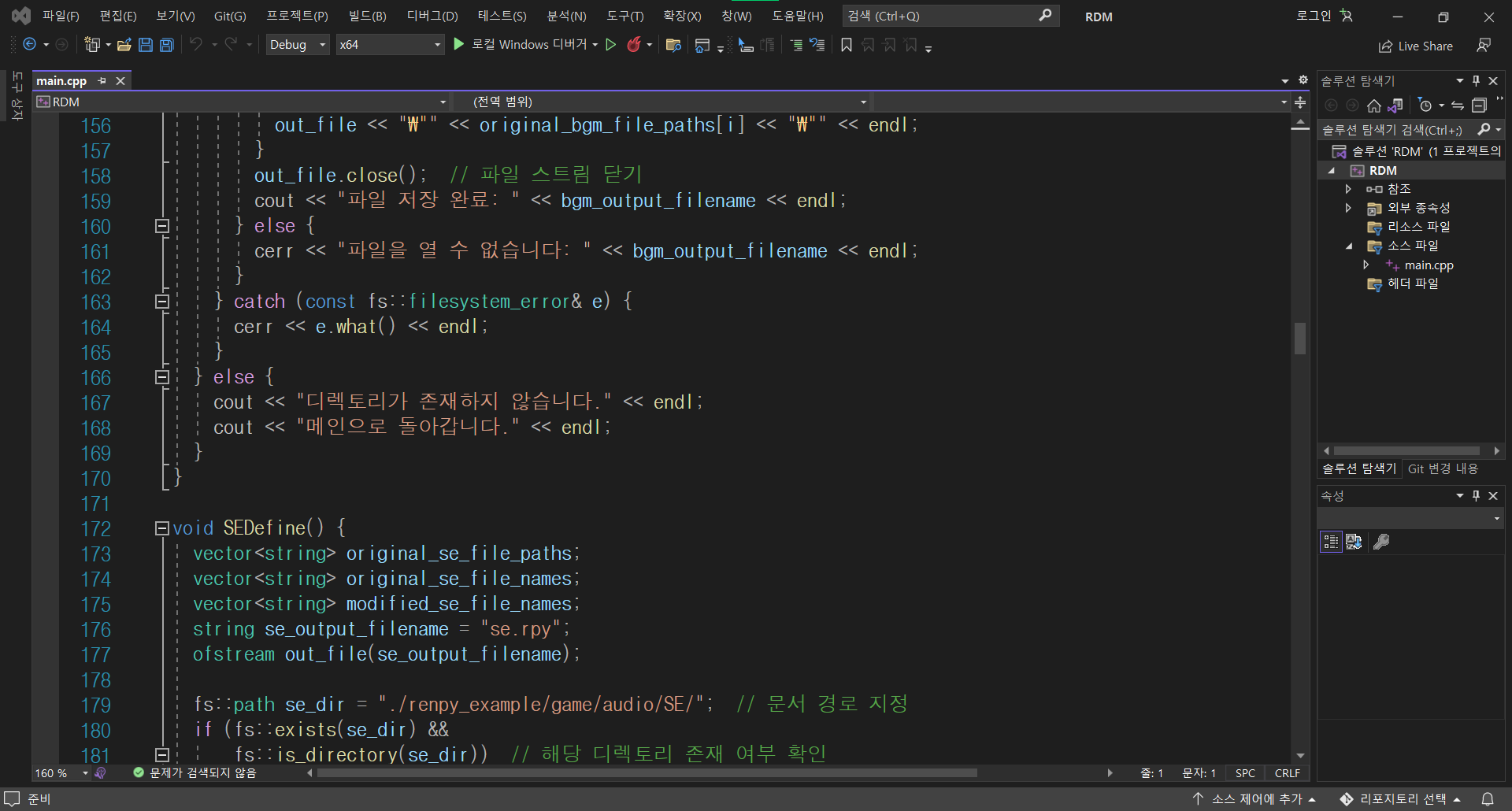
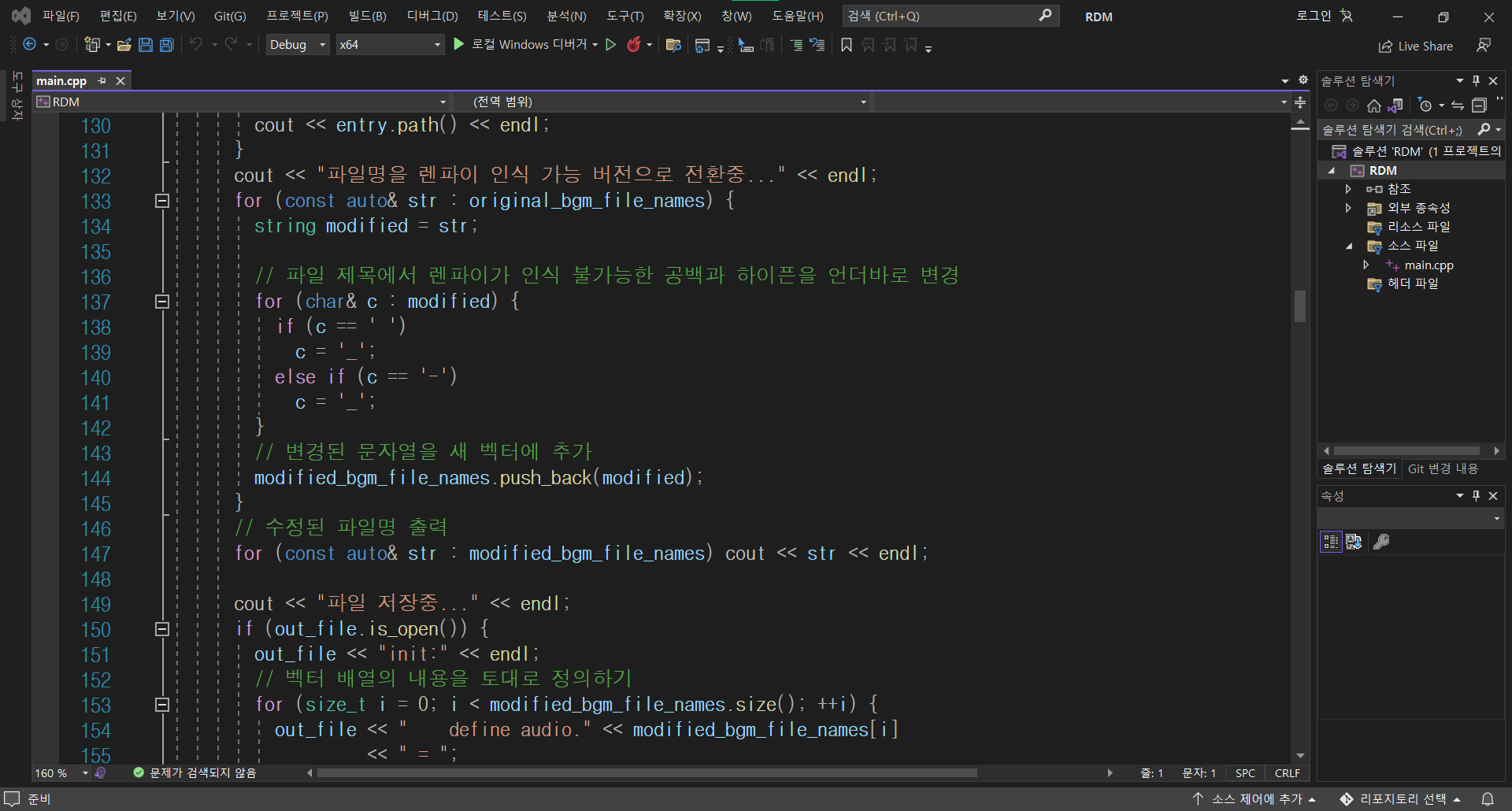
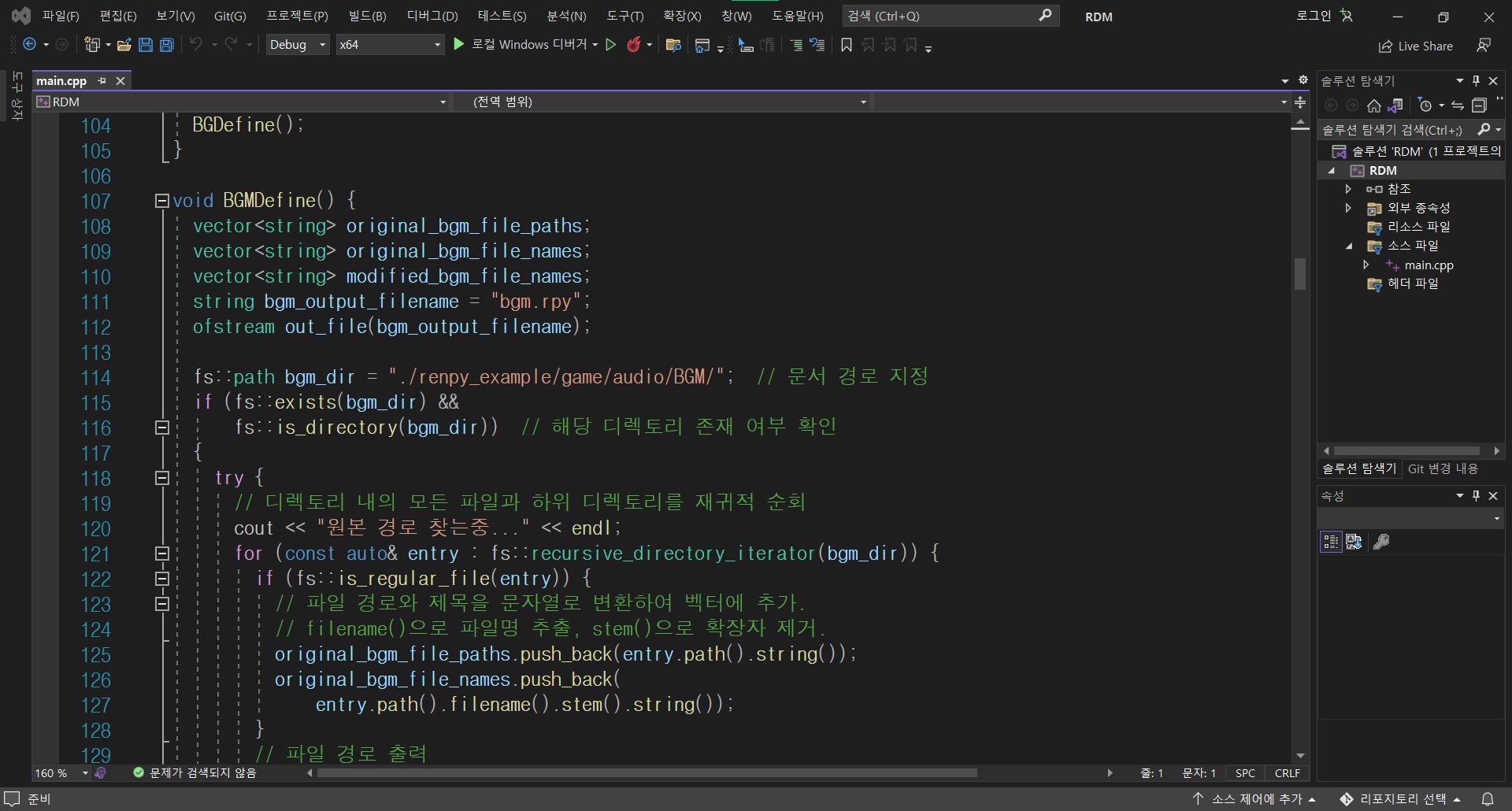
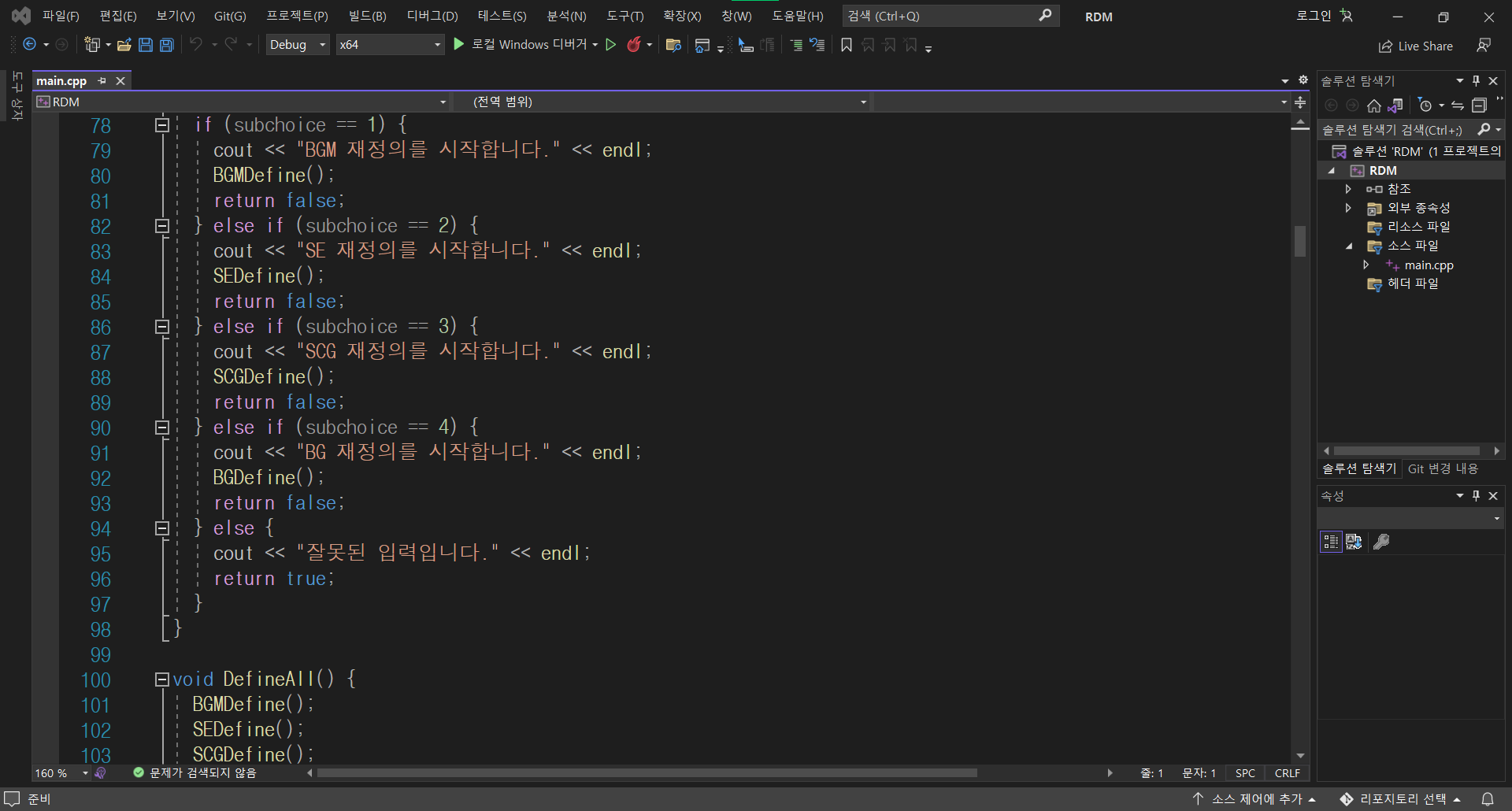
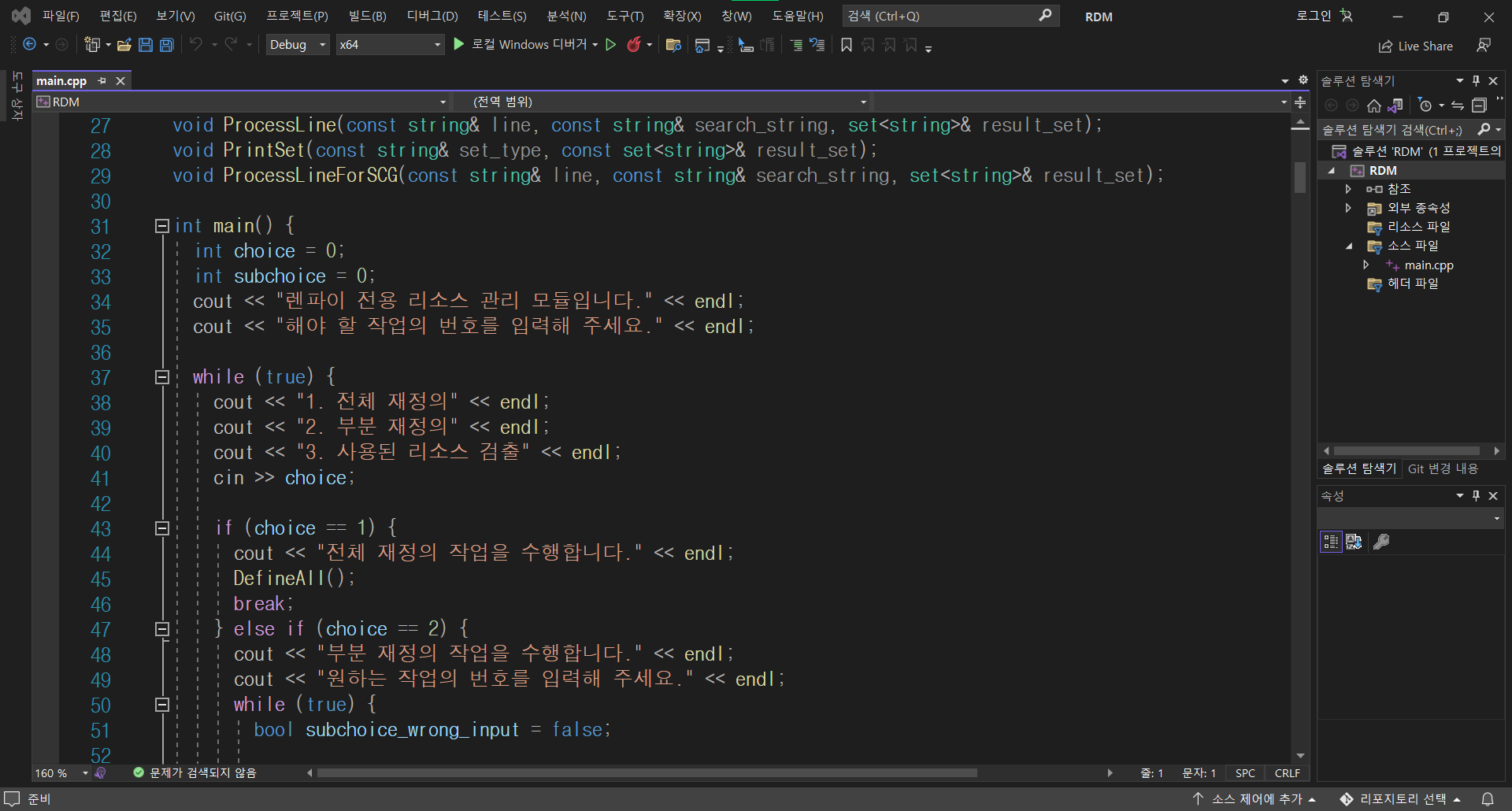
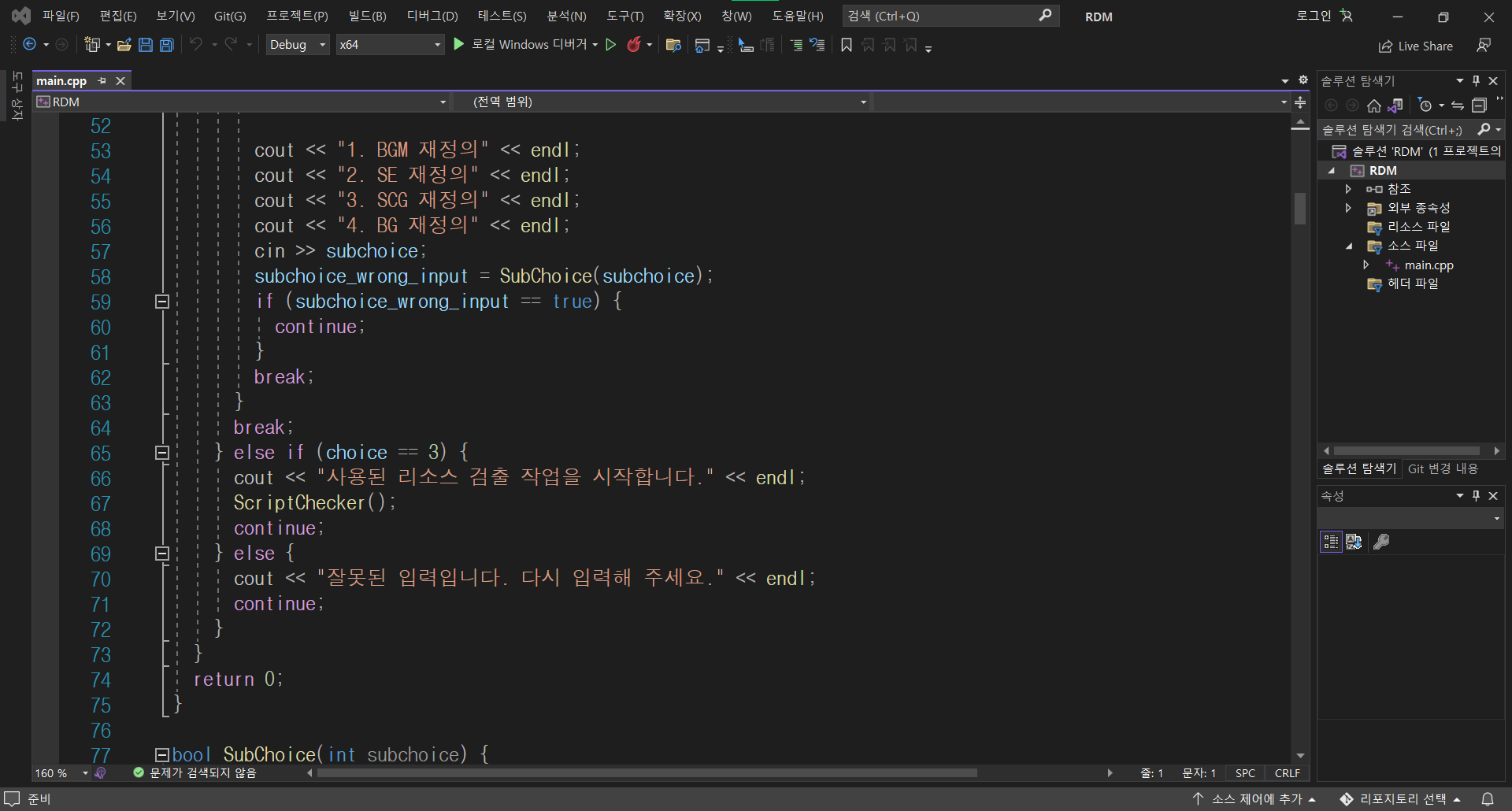
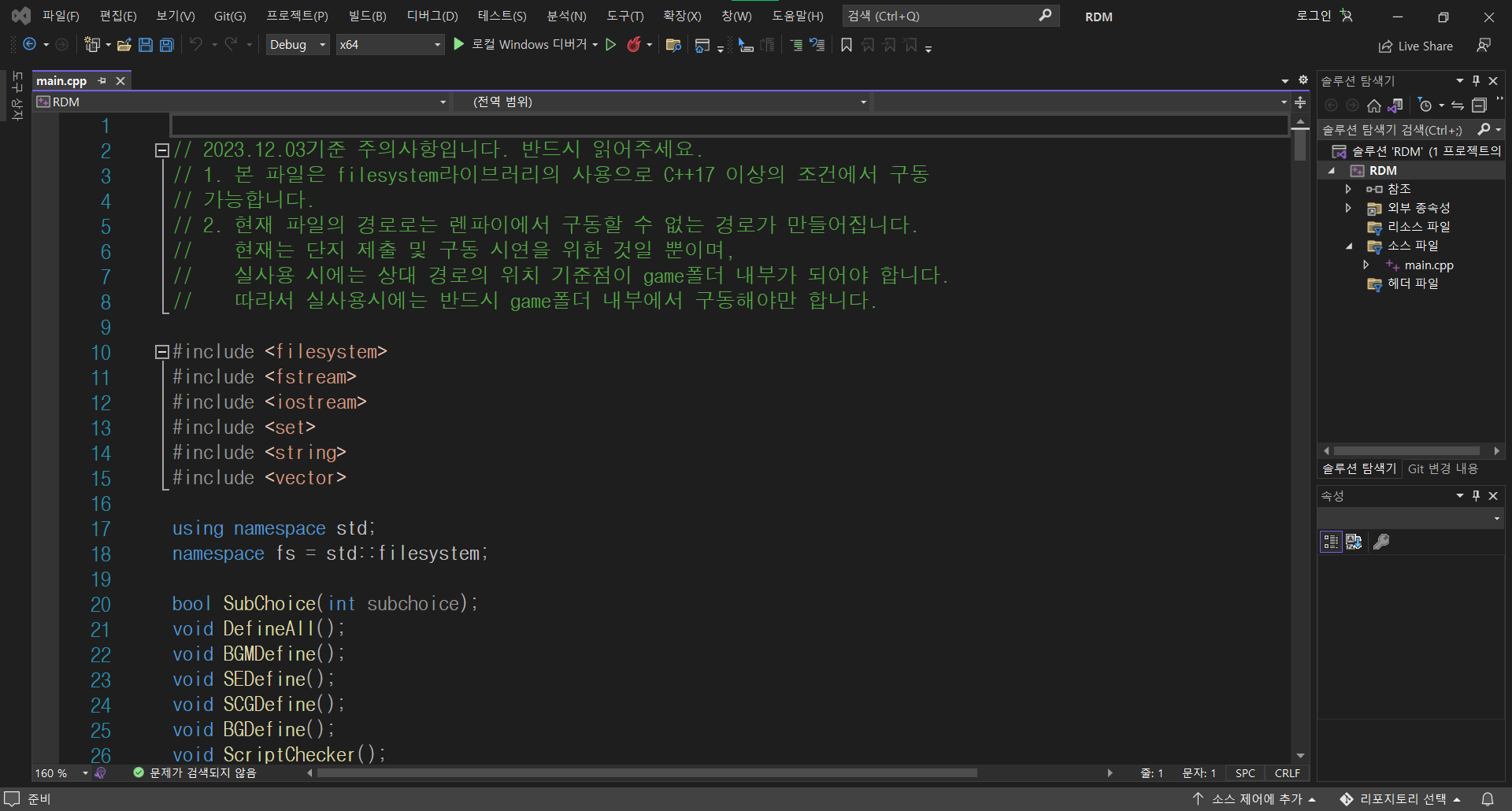
- 설명

script.rpy를 읽어 와서, 그 파일의 내용에서 특정한 문구 이후의 내용을 한 줄 추출한다. 예를 들면 play music Clair\_de\_Lune fadein 0.5라는 음악 재생을 의미하는 스크립트 명령어가 script.rpy에 저장되어 있다고 치자. 이때 “play music”을 기준으로 어떤 줄을 추출해야 하는지를 식별한다. 그 뒤 play music 뒤에 있는 Clair\_de\_Lune fadein 0.5라는 문자열에서 가장 처음 오는 공백을 기준으로 앞부분만 따로 잘라내서 배열에 저장한다. 따라서 남는 문자열은 Clair\_de\_Lune이 된다. SCG의 경우는 조금 특수한 경우이기에, 두 번째 공백을 기준으로 자르며, 첫 번째 공백은 언더바로 대체된다. 이후 이를 콘솔창에 한하여 출력하는 것이 현재까지의 기능이다.

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

조건문, 반복문, 함수, 벡터 배열 등이 사용된다. 배우지 않은 내용이지만 set라이브러리가 중복 방지용으로 사용된다.

- 코드 스크린샷



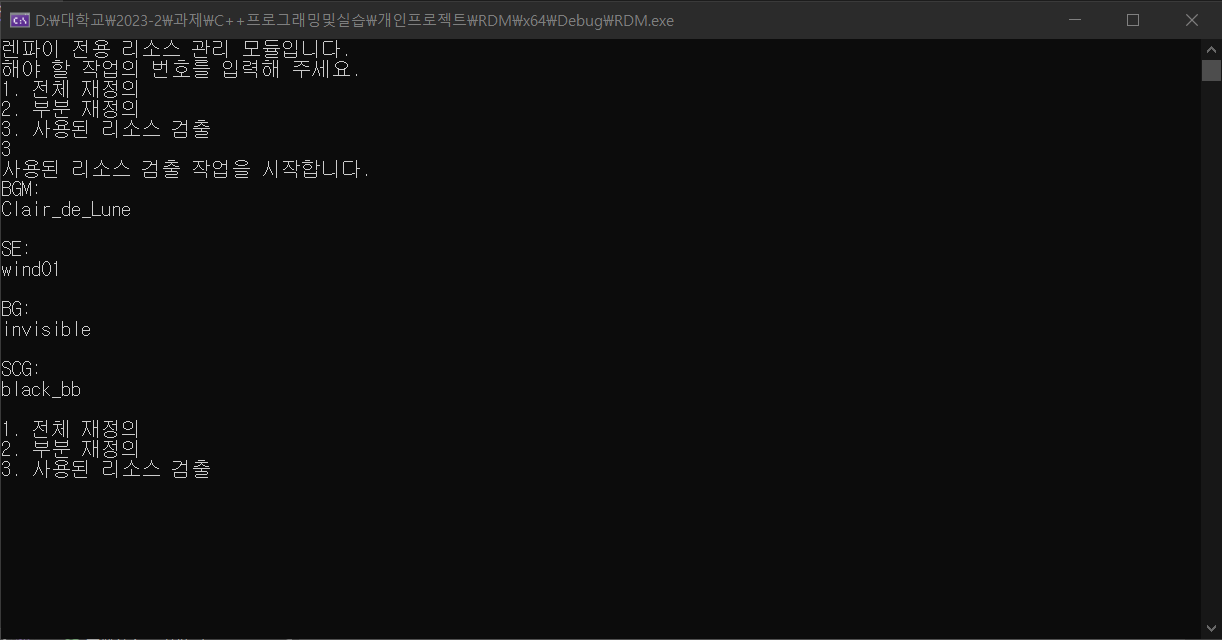
**2) 테스트 결과**

**(1) 사용되지 않은 리소스 식별 기능**

- 설명

현재는 사용된 리소스들이 식별된다. 나머지 부분은 전술한 그대로이다.

- 테스트 결과 스크린샷



**4. 계획 대비 변경 사항**

**없음.**

**5. 프로젝트 일정**

리소스 정의 부분은 SCG부분에서 언더바가 없는 파일이 존재할 시 에러를 뱉던 부분을처리되지 않도록 개선 완료함으로써 완성되었다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **업무** | **11/3** | **11/26** | **12/4** | **12/23** |
| **제안서 작성** | **완료** |  |  |  |
| **기능1** |  | **완료** | |  |
| **기능2** |  |  | **진행 중** | |
| **디버그** |  |  |  | **------------>** |