

# 졸업작품

## 교과목 포트폴리오

이름	김시현
과목	졸업작품
학과	컴퓨터 정보공학과
학번	20185018

## <제 목 차 례>

1. Android 스튜디오 .....	3
1.1 Android 스튜디오 소개 .....	3
1.2 프로젝트 구조 .....	3
1.3 사용자 인터페이스 .....	4
1.4 Gradle 빌드 시스템 .....	8
1.5 디버그 및 프로파일링 도구 .....	9
1.6 개발자 계정에 로그인 .....	11
2. 졸업작품 중 사용한 안드로이드 소스 코드 .....	12
2.1 네이버 지도 시작하기 .....	12
2.2 위치 .....	14
2.3 사용자 인터페이스 .....	17
2.4 XML 데이터 파싱 .....	19
3. 안드로이드 스튜디오 깃허브 협업 .....	26

# 1. Android 스튜디오

## 1.1 Android 스튜디오 소개

Android 스튜디오는 Android 앱 개발을 위한 공식 통합 개발 환경이며 IntelliJ IDEA를 기반으로 한다. IntelliJ의 강력한 코드 편집기와 개발자 도구 외에도 Android 스튜디오는 Android 앱을 빌드할 때 생산성을 높여주는 다음과 같은 기능을 제공한다.



- 유연한 Gradle 기반 빌드 시스템
- 빠르고 기능이 풍부한 에뮬레이터
- 앱을 다시 시작할 필요 없이 변경사항을 적용하여 실행 중인 앱에 코드 및 리소스 변경사항을 푸시
- 일반적인 앱 기능을 빌드하고 샘플 코드를 가져오는 데 도움이 되는 코드 템플릿과 Github 통합
- 광범위한 테스트 도구 및 프레임워크
- 성능, 사용성, 버전 호환성 및 기타 문제를 파악하는 린트 도구
- C++ 및 NDK 지원
- Google 클라우드 메시징과 App Engine을 간편하게 통합하는 Google Cloud Platform을 기본적으로 지원

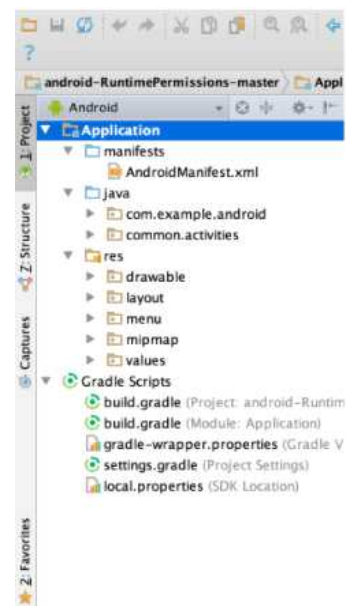
## 1.2 프로젝트 구조

Android 스튜디오의 각 프로젝트에는 소스 코드 파일 및 리소스 파일과 함께 모듈이 하나 이상 포함된다. 모듈의 유형은 다음과 같다.

- Android 앱 모듈
- 라이브러리 모듈
- Google App Engine 모듈

Android 스튜디오는 기본적으로 Android 프로젝트 뷰에 프로젝트 파일을 표시한다. 프로젝트의 주요 소스 파일에 신속하게 액세스할 수 있도록 이 뷰는 모듈별로 구성된다. 모든 빌드 파일은 Gradle Scripts의 최상단에 표시되며 각 앱 모듈에는 다음 폴더가 포함된다.

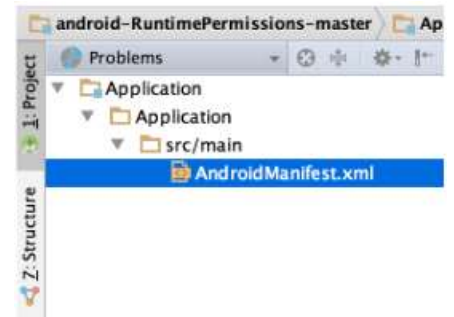
- manifests: AndroidManifest.xml 파일을 포함한다.
- java: JUnit 테스트 코드를 비롯한 자바 소스 코드 파일을 포함한다.
- res: 코드가 아닌 모든 리소스를 포함한다.(XML 레이아웃, UI 문자열,



비트맵 이미지 등)

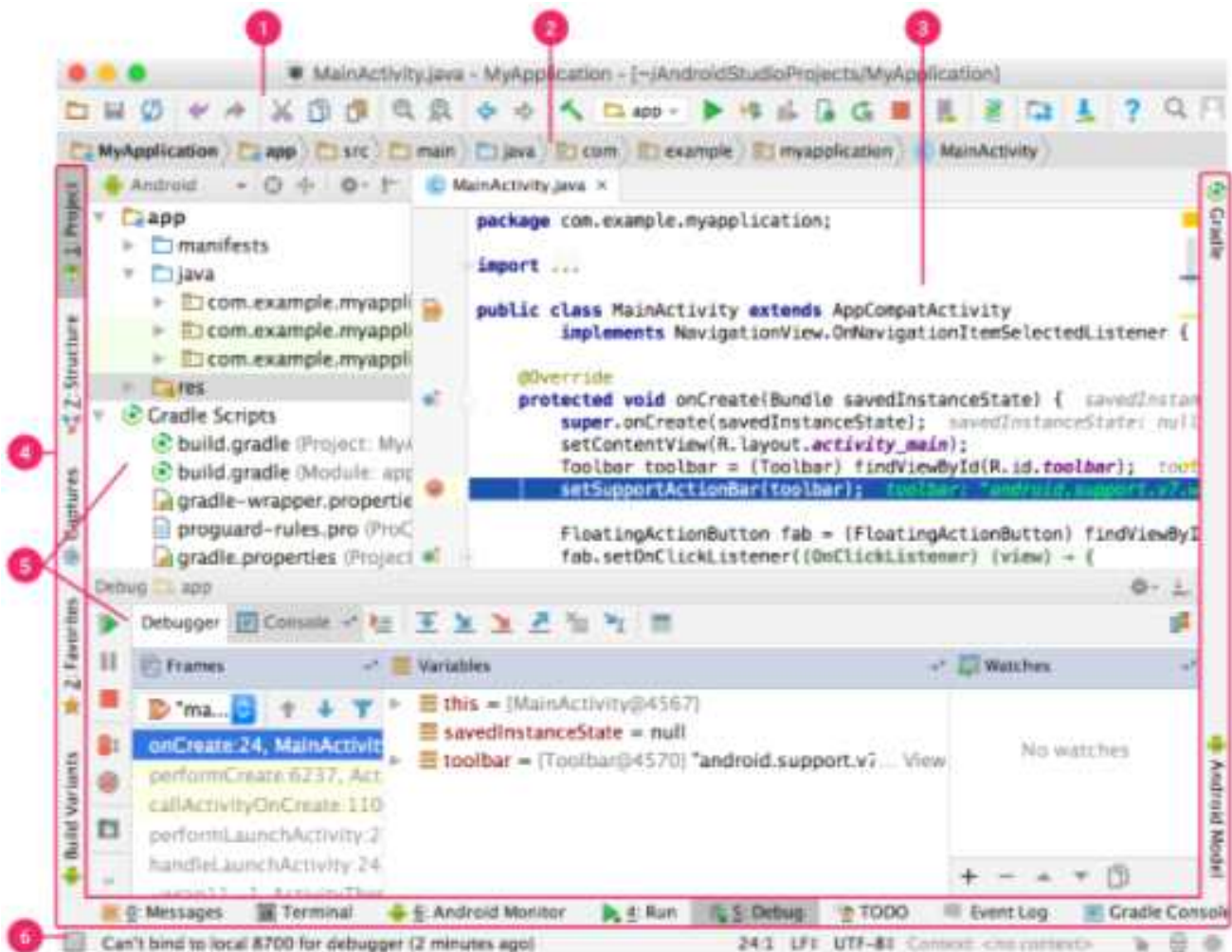
디스크의 Android 프로젝트 구조는 이 축약 표현과 다르다. 프로젝트의 실제 파일구조를 보려면, **Project** 드롭다운에서 **Project**를 선택한다.

프로젝트의 파일과 뷰가 앱 개발의 특정 측면에 중점을 두도록 개발자가 맞춤설정할 수도 있다. 예를 들어, 프로젝트의 **Problems** 뷰를 선택하면 레이아웃 파일에서 인식된 코딩 및 구문 오류(예: XML 요소 닫기 태그의 누락)가 있는 소스 파일에 관한 링크가 표시된다.



## 1.3 사용자 인터페이스

Android 스튜디오 기본 창은 다음 그림과 같이 여러 개의 논리적 영역으로 구성된다.



- ① 툴바를 사용하면 앱을 실행하거나 Android 도구를 시작하는 등 다양한 작업을 처리할 수 있다.
- ② 탐색 메뉴를 사용하면 프로젝트를 탐색하고 수정할 파일을 열 수 있다. 탐색 메뉴는 Project 창에 나타나는 구조를 좀 더 간략하게 표시한다.
- ③ 편집기 창에서는 코드를 작성하고 수정할 수 있다. 현재의 파일 형식에 따라 편집기가 바뀔 수 있다. 예를 들어, 레이아웃 파일을 볼 때 편집기는 Layout Editor를 표시한다.


- ④ 도구 창 모음은 IDE 창 외부 주변에 있으며 개별 도구 창을 펼치거나 접을 수 있는 버튼을 포함한다.
- ⑤ 도구 창에서는 프로젝트 관리, 검색, 버전 제어 등의 특정 작업에 액세스할 수 있다. 창을 펼치고 접을 수 있다.
- ⑥ 상태 표시줄은 프로젝트와 IDE의 상태를 표시하며 경고나 메시지도 표시한다.

툴바와 도구 창을 숨기거나 이동하여 더 많은 화면 공간을 확보하도록 기본 창을 구성할 수 있다. 또한 단축키를 사용하여 대부분의 IDE 기능에 액세스할 수도 있다.

언제든지 Shift 키를 두 번 누르거나 Android 스튜디오 창의 오른쪽 상단 모서리에 있는 돋보기를 클릭하여 소스 코드, 데이터 베이스, 작업, 사용자 인터페이스 요소 등에서 검색할 수 있다. 예를 들어, 이 검색 기능은 트리거 방법을 잊은 특정 IDE 작업을 찾으려는 경우 매우 유용할 수 있다.

## 도구 창

Android 스튜디오는 미리 설정된 구성을 사용하는 대신 상황에 따라 작업 시에 관련 도구 창을 자동으로 불러온다. 기본적으로 가장 자주 사용하는 도구 창이 애플리케이션 창 가장자리의 도구 창 모음에 고정된다.

- 도구 창을 펼치거나 접으려면 도구 창 모음에서 도구 이름을 클릭한다. 또한 도구 창을 드래그하거나, 고정하거나, 고정 해제하거나, 연결하거나 분리할 수도 있다.
- 현재의 기본 도구 창 레이아웃으로 돌아가려면 Window > Restore Default Layout을 클릭하거나 Window > Store Current Layout as Default를 클릭하여 기본 레이아웃을 맞춤설정한다.
- 전체 도구 창 모음을 표시하거나 숨기려면 Android 스튜디오 창의 왼쪽 하단 모서리에 있는 창 아이콘  을 클릭한다.
- 특정 도구 창을 찾으려면 창 아이콘 위로 마우스를 가져간 후 메뉴에서 도구 창을 선택한다. 또한 단축키를 사용하여 도구 창을 열 수도 있다. 다음 표는 가장 일반적인 창의 단축키를 보여준다.

도구 창	Windows 및 Linux	Mac
프로젝트	Alt + 1	Command + 1
버전 제어	Alt + 9	Command + 9
실행	Shift + F10	Control + R
디버그	Shift + F9	Control + D
Logcat	Alt + 6	Command + 6
편집기로 돌아가기	Esc	Esc
모든 도구 창 숨기기	Control + Shift + F12	Command + Shift + F12

모든 툴바, 도구 창 및 편집기 탭을 숨기려면 View > Enter Distraction Free Mode를 클릭

한다. 그러면 Distraction Free Mode가 활성화된다. Distraction Free Mode를 종료하려면 View > Exit Distraction Free Mode를 클릭한다.

Speed Search를 사용하면 Android 스튜디오의 대부분의 도구 창 내에서 검색하고 필터링할 수 있다. Speed Search를 사용하려면 도구 창을 선택한 다음 원하는 검색어를 입력한다.

## 코드 작성

Android 스튜디오에는 단축키를 사용하여 액세스할 수 있는 세 가지 코드 작성 유형이 있다.

유형	설명	Windows 및 Linux	Mac
기본완성	변수, 유형, 메서드, 표현식 등에 관한 기본 제안을 표시한다. 한 행에서 기본 완성을 두 번 호출하는 경우, 비공개 멤버 및 가져오지 않은 정적 멤버를 비롯하여 더 많은 결과가 나타난다.	Control + 스페이스	Control + 스페이스
스마트완성	상황에 따라 관련 옵션을 표시한다. 스마트 완성은 예상되는 유형과 데이터 흐름을 인식한다. 한 행에서 스마트 완성을 두 번 호출하는 경우, 체인을 비롯하여 더 많은 결과가 나타난다.	Control + Shift + 스페이스	Control + Shift + 스페이스
명령문완성	현재 명령문을 자동으로 완성하고, 누락된 소괄호, 대괄호, 서식 등을 추가한다.	Control + Shift + Enter	Shift + Command + Enter

Alt + Enter를 눌러 신속하게 수정하고 원하는 작업을 표시할 수도 있다.

## 샘플 코드 찾기

Android 스튜디오의 코드 샘플 브라우저를 사용하면 프로젝트에서 현재 강조표시된 기호를 바탕으로 Google에서 제공하는 고품질 Android 코드 샘플을 손쉽게 찾을 수 있다.

## 탐색

다음은 Android 스튜디오를 탐색하는 데 필요한 몇 가지 유용한 정보이다.

● 최근에 액세스한 파일 간에 전환하려면 Recent Files 작업을 사용한다.

Control + E(Mac은 Command + E)를 눌러 Recent Files 작업을 불러온다. 기본적으로 마지막 액세스한 파일이 선택한다. 이 작업에서 왼쪽 열을 통해 모든 도구 창에 액세스할 수도 있다.

● 현재 파일의 구조를 보려면 File Structure 작업을 사용한다.

Control + F12(Mac은 Command + F12)를 눌러 File Structure 작업을 불러온다. 이 작업을 사용하면 현재 파일의 모든 부분을 신속하게 탐색할 수 있다.

- 프로젝트에서 특정 클래스를 검색하고 탐색하려면 **Navigate to Class** 작업을 사용한다. **Control + N**(Mac은 **Command + O**)을 눌러 이 작업을 불러온다. **Navigate to Class**는 캐털 험프, 경로, 라인 탐색, 중간 이름 일치 등의 고급 표현식을 지원한다. 이 작업을 한 행에서 두 번 호출하는 경우, 프로젝트 클래스의 결과를 표시한다.
- 파일이나 폴더를 탐색하려면 **Navigate to File** 작업을 사용한다. **Control + Shift + N**(Mac은 **Command + Shift + O**)을 눌러 **Navigate to File** 작업을 불러온다. 파일이 아니라 폴더를 검색하려면 표현식 끝에 **/**를 추가한다.
- 이름별로 메서드나 필드를 탐색하려면 **Navigate To Symbol** 작업을 사용한다. **Control + Shift + Alt + N**(Mac은 **Command + Option + O**)을 눌러 **Navigate to Symbol** 작업을 불러온다.
- 현재 커서 위치에서 클래스, 메서드, 필드, 매개변수 또는 명령문을 참조하는 모든 코드 조각을 찾으려면 **Alt + F7**(Mac은 **Option + F7**)을 누른다.

## 스타일 및 서식

수정 중에 Android 스튜디오는 코드 스타일 설정에 지정된 대로 서식과 스타일을 자동으로 적용한다. 프로그래밍 언어로 코드 스타일 설정을 맞춤설정할 수 있으며, 여기에는 탭과 들여쓰기 간격, 공백, 줄바꿈 및 괄호, 빈 줄의 규칙을 지정하는 것이 포함된다. 코드 스타일을 맞춤설정하려면 **File > Settings > Editor > Code Style**(Mac은 **Android Studio > Preferences > Editor > Code Style**)을 클릭한다.

IDE는 작업 중에 서식을 자동으로 적용하지만 **Control + Alt + L**(Mac은 **Opt + Command + L**)을 눌러 **Reformat Code** 작업을 명시적으로 호출하거나 **Control + Alt + I**(Mac은 **Control + Option + I**)를 눌러 모든 줄에 자동 들여쓰기 간격을 적용할 수 있다.

(서식 지정 전의 코드)

(서식 지정 후의 코드)

## 버전 제어 기본사항

Android 스튜디오에서는 다양한 버전 제어 시스템(VCS)을 지원하며 여기에는 **Git**, **GitHub**, **CVS**, **Mercurial**, **Subversion** 및 **Google Cloud Source Repositories**가 포함된다.

앱을 Android 스튜디오로 가져온 후에 Android 스튜디오의 VCS 메뉴 옵션을 사용하여 원하는 버전 제어 시스템에 VCS를 지연하고, 저장소를 만들고, 새 파일을 버전 제어로 가져오고, 기타 버전 제어 작업을 처리한다.

1. Android 스튜디오의 VCS 메뉴에서 Enable Version Control Integration을 클릭한다.

2. 드롭다운 메뉴에서 프로젝트 루트에 연결할 버전 제어 시스템을 선택한 다음 OK를 클릭한다.

이제 선택한 시스템에 따라 VCS 메뉴에 여러 가지 버전 제어 옵션이 표시된다.

## 1.4 Gradle 빌드 시스템

Android 스튜디오는 빌드 시스템의 기반으로 Gradle을 사용하며, 더 많은 Android 전용 기능이 Gradle용 Android 플러그인을 통해 제공된다. 이 빌드 시스템은 Android 스튜디오 메뉴에서는 통합 도구로 실행되고, 명령줄에서는 독립적으로 실행된다. 이 빌드 시스템의 기능을 사용하여 다음 작업을 실행할 수 있다.

- 빌드 프로세스 맞춤설정, 구성 및 확장
- 동일한 프로젝트 및 모듈을 사용하여 다양한 기능을 갖춘 앱을 위한 여러 APK 생성
- 전체 소스 세트에 걸쳐 코드 및 리소스 재사용

Gradle의 유연성을 활용하면 앱의 주요 소스 파일을 수정하지 않고도 이 모든 작업을 처리할 수 있다. Android 스튜디오 빌드 파일은 `build.gradle`로 명명된다. 이 빌드 파일은 Groovy 구문을 사용하는 일반 텍스트 파일이며 Gradle용 Android 플러그인이 제공하는 요소들로 빌드를 구성한다. 전체 프로젝트에서 각 프로젝트에는 하나의 최상위 빌드 파일이 있으며 모듈별로 별도의 모듈 수준 빌드 파일이 있다. 개발자가 기존 프로젝트를 가져오면 Android 스튜디오는 필요한 빌드 파일을 자동으로 생성한다.

### 빌드 변형

빌드 시스템을 사용하여 단일 프로젝트에서 동일한 애플리케이션의 다양한 버전을 만들 수 있다. 이 기능은 무료 버전 앱과 유료 버전 앱이 둘 다 있거나 Google Play에서 다른 기기 설정에 관해 여러 APK를 배포하려는 경우에 유용하다.

### 다중 APK 지원

다중 APK 지원을 사용하면 화면 밀도 또는 ABI를 기반으로 여러 APK를 효율적으로 만들 수 있다. 예를 들어, `hdpi` 및 `mdpi` 화면 밀도에 맞게 앱의 개별 APK를 만드는 동시에 이들을 단일 변형으로 간주하여 테스트 APK, `javac`, `dx` 및 ProGuard 설정을 공유하도록 허용할 수 있다.

### 리소스 축소

Android 스튜디오의 리소스 축소는 패키지 앱과 라이브러리 종속 항목으로부터 미사용 리소스를 자동으로 삭제한다. 예를 들어, 애플리케이션에서 Google Play 서비스를 사용하여 Google 드라이브 기능에 액세스하고 현재 Google 로그인을 사용하고 있지 않은 경우 리소스 축소를 사용하면



SignInButton 버튼의 다양한 드로어블 애셋을 삭제할 수 있다.

## 종속 항목 관리

프로젝트의 종속 항목은 build.gradle 파일에 이름으로 지정된다. Gradle은 종속 항목을 찾고 빌드에서 사용할 수 있게 만든다. 모듈 종속 항목, 원격 바이너리 종속 항목, 로컬 바이너리 종속 항목을 build.gradle 파일에서 선언할 수 있다. 기본적으로 Android 스튜디오는 Maven Central Repository를 사용하도록 프로젝트를 구성한다. (이 구성은 프로젝트의 최상위 빌드 파일에 포함된다.)

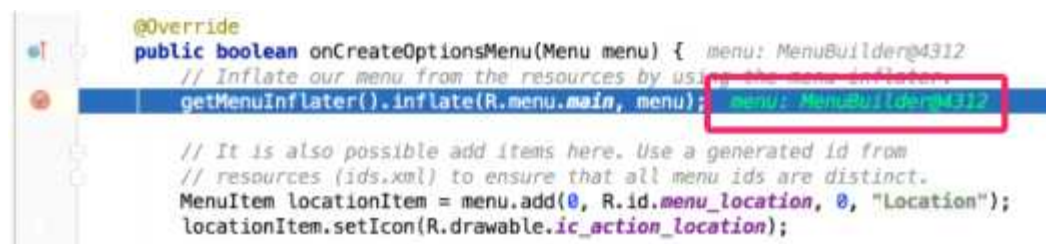
## 1.5 디버그 및 프로파일링 도구

Android 스튜디오는 코드를 디버깅하고 코드 성능을 개선하도록 도와주며 여기에는 인라인 디버깅 및 성능 분석 도구가 포함된다.

### 인라인 디버깅

참조, 표현식 및 변수 값의 인라인 확인 기능이 있는 디버거 뷰에서 인라인 디버깅을 사용하여 코드 검토 능력을 높인다. 인라인 디버그 정보에는 다음과 같은 항목이 포함된다.

- 인라인 변수 값
- 선택한 객체를 참조하는 참조 객체
- 메서드 반환 값
- 람다 및 연산자 표현식
- 도움말 값



인라인 디버깅을 사용 설정하려면 Debug 창에서 Settings를 클릭하고 Show Values Inline의 체크박스를 선택한다.

## 성능 프로파일러

Android 스튜디오는 성능 프로파일러를 제공하며 이를 통해 앱의 메모리 및 CPU 사용량을 보다 쉽게 추적하고, 할당 해제된 객체를 확인하고, 메모리 누수를 찾아내고, 그래픽 성능을 최적화하고, 네트워크 요청을 분석할 수 있다. 앱이 기기 또는 에뮬레이터에서 실행 중일 때 Android Profiler 탭을 연다.

## 힙 덤프

Android 스튜디오에서 메모리 사용량을 프로파일링 중인 경우, 동시에 가비지 컬렉션을 시작하고 자바 힙을 Android 전용 HPROF 바이너리 형식 파일로 힙 스냅샷에 덤프할 수 있다. HPROF 뷰어는 클래스, 각 클래스의 인스턴스, 참조 트리를 표시하므로 메모리 사용량을 추적하고 메모리 누수를 찾는 데 도움이 된다.

## 메모리 프로파일러

메모리 프로파일러를 사용하면 메모리 할당을 추적하고 특정 작업을 실행할 때 객체가 어디에서 할당되는지를 살펴볼 수 있다. 이러한 할당을 알고 있으면 작업과 관련된 메서드 호출을 조정하여 앱의 성능과 메모리 사용을 최적화할 수 있다.

## 데이터 파일 액세스

Systrace 및 logcat과 같은 Android SDK 도구는 자세한 앱 분석을 위해 성능 및 디버깅 데이터를 생성한다.

사용 가능한 생성된 데이터 파일을 보려면 Captures 도구 창을 연다. 생성된 파일 목록에서 파일을 더블클릭하여 데이터를 본다. 임의의 .hprof 파일을 마우스 오른쪽 버튼으로 클릭하여 표준 RAM 사용량 조사 파일 형식으로 변환한다.

## 코드 검사

개발자가 프로그램을 컴파일할 때마다 Android 스튜디오는 구성된 린트 및 기타 IDE검사를 자동으로 실행하여 코드의 구조적 품질 문제를 쉽게 식별하고 수정하도록 도와준다.

린트 도구는 Android 프로젝트 소스 파일을 검사하여 잠재적 버그를 찾아내고 정확성, 보안, 성능, 사용성, 접근성 국제화를 개선하여 최적화한다.



린트 검사 외에도, Android 스튜디오는 IntelliJ 코드 검사를 실행하고 주석의 유효성을 검사하여 코딩 워크플로를 능률화한다.

## Android 스튜디오의 주석

Android 스튜디오는 변수, 매개변수 및 반환 값에 관한 주석을 지원하므로, 개발자는 null 포인터 예외 및 리소스 유형 충돌 등의 버그를 포착할 수 있다. Android SDK Manager는 Android 스튜디오에서 사용할 수 있도록 Support-Annotations 라이브러리를 Android 지원 저장소에서 패키징한다. Android 스튜디오는 코드를 검사하는 동안 구성된 주석의 유효성을 검사한다.

## 로그 메시지

Android 스튜디오로 앱을 빌드 및 실행할 때 Logcat 창에서 adb 출력 및 기기 로그 메시지를 볼 수 있다.


## 성능 프로파일링

앱의 CPU, 메모리 및 네트워크 성능을 프로파일링하려면 View > Tool Windows > Android Profiler를 클릭하여 Android Profiler를 연다

## 1.6 개발자 계정에 로그인

Android 스튜디오에서 개발자 계정에 로그인하여 Android Studio용 Cloud Tools 및 App Actions Test Tool과 같이 인증이 필요한 추가 도구에 액세스할 수 있다. 로그인하면 이러한 도구에 다양한 Google 서비스에서 내 데이터를 보고 관리할 권한을 부여하게 된다.

Android 스튜디오에서 프로젝트를 연 후에 다음과 같은 방법으로 개발자 계정에 로그인하거나 개발자 계정을 전환할 수 있다.

1.  그림과 같이 툴바 끝에 있는 프로필 아이콘을 클릭한다.
2. 창이 열리면 다음 단계 중 하나를 따른다.
  - 아직 로그인하지 않았다면 Sign In을 클릭하여 표시된 서비스에 Android 스튜디오가 액세스할 수 있도록 허용한다.
  - 이미 로그인 했다면 Add Account를 클릭하여 다른 Google 계정으로 로그인한다. 또는 Sign Out을 클릭한 후 이전 단계를 반복하여 다른 계정에 로그인할 수 있다.

## 2. 졸업작품 중 사용한 안드로이드 소스 코드

### 2.1 네이버 지도 시작하기

네이버 지도 SDK를 사용하기 위해서는 네이버 클라우드 플랫폼에서 클라이언트 ID를 발급받고, 발급받은 ID를 SDK에 지정해야 한다.

#### 지원 OS 버전

네이버 지도 SDK는 안드로이드 OS 버전 4.0.3(API 레벨 15) 이상에서 사용할 수 있다.

#### 클라이언트 ID 발급

네이버 지도 SDK를 사용하려면 다음과 같이 애플리케이션을 등록하고 클라이언트 ID를 발급받아야 한다.

1. 네이버 클라우드 플랫폼에 로그인한 후 콘솔에 들어간다.
2. Products & Services에서 AI-Application Service 하위의 AI-NAVER API를 선택한다.
3. Application 등록을 선택하고 Maps 하위의 Mobile Dynamic Map을 체크한다.
  - 선택하지 않으면 인증 실패 오류(429)가 발생한다.
4. Android 앱 패키지 이름에 네이버 지도 SDK를 사용하고자 하는 앱의 패키지명을 추가하고 등록한다.
  - 올바르게 입력하지 않으면 인증 실패 오류(401)가 발생한다.
5. 등록한 애플리케이션의 인증 정보를 선택해 Client ID를 확인한다.

#### 의존성 추가

네이버 지도 SDK는 <https://naver.jfrog.io/artifactory/maven/Maven> 저장소에서 배포된다. 루프 프로젝트의 build.gradle에 저장소 설정을 추가한다.

다음은 저장소 설정을 추가한 예제이다.

```
repositories { RepositoryHandler it ->
    google()
    mavenCentral()
    jcenter() // Warning: this repository is going to shut down soon
    maven{
        url 'https://naver.jfrog.io/artifactory/maven/'
    }
}
```

그리고 앱 모듈의 build.gradle에 네이버 지도 SDK에 대한 의존성을 선언한다.

다음은 네이버 지도 SDK에 대한 의존성을 선언한 예제이다.

```
implementation 'com.naver.maps:map-sdk:3.12.0'
```

네이버 지도 SDK는 appcompat-v7 28 버전을 사용한다. 만약 앱에서 다른 버전을 사용한다면 SDK의 의존성에서 com.android.support 그룹을 제외시키고 명시적으로 appcompat-v7에 대한 의존성을 선언한다.

다음은 appcompat-v7 27.0.0 버전을 사용하는 예제이다.

```
dependencies {  
    implementation 'com.android.support:appcompat-v7:27.0.0'  
  
    implementation('com.naver.maps:map-sdk:3.13.0') {  
        exclude group: 'com.android.support'  
    }  
}
```

## 클라이언트 ID 지정

발급받은 클라이언트 ID를 SDK에 지정하면 지도 API를 사용할 수 있다. 클라이언트 ID는 두 가지 방식으로 지정할 수 있다.

## AndroidManifest.xml에 지정

AndroidManifest.xml의 <meta-data>로 클라이언트 ID를 지정할 수 있다. <application> 아래에 <meta-data> 요소를 추가하고, name 으로 com.naver.maps.map.CLIENT\_ID를, value로 발급받은 클라이언트 ID를 지정한다.

다음은 AndroidManifest.xml에 클라이언트 ID를 지정하는 예제이다.

```
<meta-data  
    android:name="com.naver.maps.map.CLIENT_ID"  
    android:value="1gsptfryrl" />
```

공공기관용 클라이언트는 추가로 name이 com.naver.maps.map.CLIENT\_TYPE, value가 gov인 <meta-data>를 지정해야 한다.

다음은 AndroidManifest에 공공기관용 클라이언트를 지정하는 예제이다.

```
<manifest>  
    <application>  
        <meta-data  
            android:name="com.naver.maps.map.CLIENT_ID"  
            android:value="YOUR_CLIENT_ID_HERE" />  
        <meta-data  
            android:name="com.naver.maps.map.CLIENT_TYPE"  
            android:value="gov" />  
    </application>  
</manifest>
```

## 지도 표시

클라이언트 ID를 지정하고 MapFragment를 앱의 레이아웃에 추가하면 지도가 화면에 나타난다. 다음은 <MapView> 요소로 XML 레이아웃에 FrameLayout을 추가해 지도를 화면에 나타난 예제이다.

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FCF5E5">

    <com.naver.maps.map.MapView
        android:id="@+id/navermap"
        android:layout_width="match_parent"
        android:layout_height="643dp" />

</FrameLayout>
```

## 인증 실패 처리

클라이언트 ID 또는 유형을 잘못 지정했거나 사용량이 쿼터를 초과하면 지도가 화면에 나타나지 않으며, Logcat으로 인증 오류 메시지가 출력된다. NaverMapSdk.onAuthFailedListener 속성에 OnAuthFailedListener를 등록하면 인증 실패 이벤트를 받을 수 있다. 리스너를 등록하면 인증 실패 시 onAuthFailed() 콜백 메서드가 호출되고 토스트는 호출되지 않는다.

인증 실패 이벤트는 명백하게 인증에 실패했음이 확인된 경우에만 발생한다. 따라서 인터넷 연결이 불가능할 경우 이벤트가 발생하지 않는다. 캐시된 데이터로 지도를 로딩한 후 인터넷 연결이 재개되면 다시 인증을 시도하고, 이 때 인증에 실패하면 이벤트가 발생한다.

onAuthFailed()에는 인증 실패의 원인을 나타내는 exception 파라미터가 전달된다. exception의 타입과 errorCode 속성으로 원인을 파악할 수 있다. 주로 발생하는 에러 코드는 다음과 같다.

errorCode	예외 타입	설명
401	UnauthorizedClientException	- 잘못된 클라이언트 ID를 지정함 - 잘못된 클라이언트 유형을 사용함 - 콘솔에서 앱 패키지 이름을 잘못 등록함
429	QuatoExceededException	- 콘솔에서 Maps 서비스를 선택하지 않음 - 사용 한도가 초과됨
800	ClieitUnspecifiedException	- 클라이언트 ID를 지정하지 않음

그 외의 원인으로 인증에 실패하면 AuthFailedException이 발생한다.

## 2.2 위치

지도를 사용하는 앱은 사용자의 위치를 추적하고 지도에 표현하는 경우가 많다. 네이버 지도 SDK

는 이런 기능을 손쉽게 구현할 수 있도록 위치 오버레이와 위치 추적 기능을 제공한다. 내장된 위치 추적 기능을 사용하지 않고 직접 위치 관련 기능을 구현할 수도 있다.

## 위치 오버레이

사용자의 위치를 지도에 나타내고자 할 때는 위치 오버레이를 사용하는 것이 권장된다. 위치 오버레이는 지도상에 단 하나만 존재하며, 좌표, 방향 등 위치 표현에 특화된 기능을 제공한다.

## 권한과 LocationSource

네이버 지도 SDK는 기본적으로 사용자의 위치 정보를 사용하지 않으므로 사용자에게 위치와 관련된 권한을 요구하지 않는다. 따라서 위치 추적 기능을 사용하고자 하는 앱은 AndroidManifest.xml에 ACCESS\_COARSE\_LOCATION 또는 ACCESS\_FINE\_LOCATION 권한을 명시해야 한다.

다음은 AndroidManifest.xml에 ACCESS\_FINE\_LOCATION 권한을 명시하는 예제이다.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

또한 setLocationSource()를 호출해 LocationSource 구현체를 지정해야 한다. LocationSource는 네이버 지도 SDK에 위치를 제공하는 인터페이스이다. activate(), deactivate() 등 LocationSource의 메서드는 지도 객체가 호출하므로 개발자가 직접 호출해서는 안된다.

## FusedLocationSource

네이버 지도 SDK는 Google Play 서비스의 FusedLocationProviderClient와 지자기, 가속도 센서를 활용해 최적의 위치를 반환하는 구현체인 FusedLocationSource를 제공한다. FusedLocationSource를 사용하려면 앱 모듈의 build.gradle에 play-services-location에 대한 의존성을 추가해야 한다.

```
implementation 'com.google.android.gms:play-services-location:18.0.0'
```

FusedLocationSource는 런타임 권한 처리를 위해 액티비티 또는 프래그먼트를 필요로 한다. 생성자에 액티비티나 프래그먼트 객체를 전달하고 권한 요청 코드를 지정해야 한다. 그리고 onRequestPermissionsResult()에 전달해야 한다.

다음은 액티비티에서 FusedLocationSource를 생성하고 NaverMap에 지정하는 예제이다.



```

public class LocationTrackingActivity extends AppCompatActivity
    implements OnMapReadyCallback {
    private static final int LOCATION_PERMISSION_REQUEST_CODE = 1000;
    private FusedLocationSource locationSource;
    private NaverMap naverMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // ...
        locationSource =
            new FusedLocationSource(this, LOCATION_PERMISSION_REQUEST_CODE);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,
        @NonNull String[] permissions, @NonNull int[] grantResults) {
        if (locationSource.onRequestPermissionsResult(
            requestCode, permissions, grantResults)) {
            if (!locationSource.isActivated()) { // 권한 거부됨
                naverMap.setLocationTrackingMode(LocationTrackingMode.None);
            }
            return;
        }
        super.onRequestPermissionsResult(
            requestCode, permissions, grantResults);
    }

    @Override
    public void onMapReady(@NonNull NaverMap naverMap) {
        this.naverMap = naverMap;
        naverMap.setLocationSource(locationSource);
    }
}

```

## 위치 추적 모드

NaverMap에 LocationSource를 지정하면 위치 추적 기능을 사용할 수 있다. 위치 추적 기능을 사용하는 방법은 크게 두가지이다.

- 위치 추적 모드 지정: setLocationTrackingMode()를 호출하면 프로그램적으로 위치 추적 모드를 지정할 수 있다.
- 현위치 버튼 컨트롤 사용: UiSettings.setLocationButtonEnabled(true)로 현위치 버튼



컨트롤을 활성화하면 사용자의 클릭에 따라 위치 추적 모드를 변경할 수 있다.

위치 추적 모드는 다음의 네 가지이며, LocationTrakingMode 열거형에 정의되어 있다.

- None: 위치를 추적하지 않는다
- NoFollow: 위치 추적이 활성화되고, 현위치 오버레이가 사용자의 위치를 따라 움직인다. 그러나 지도는 움직이지 않는다.
- Follow: 위치 추적이 활성화되고, 현위치 오버레이와 카메라의 좌표가 사용자의 위치를 따라 움직인다. API나 제스처를 사용해 임의로 카메라를 움직일 경우 모드가 NoFollow로 바뀐다.
- Face: 위치 추적이 활성화되고, 현위치 오버레이, 카메라의 좌표, 베어링이 사용자의 위치 및 방향을 따라 움직인다. API나 제스처를 사용해 임의로 카메라를 움직일 경우 모드가 NoFollow로 바뀐다.

다음은 setLocationTrackingMode()를 호출해 위치 추적 모드를 Follow로 지정하는 예제다.

```
naverMap.setLocationTrackingMode(LocationTrackingMode.Follow);
```

## 2.3 사용자 인터페이스

사용자는 컨트롤, 제스처 등 다양한 방법으로 지도와 상호작용할 수 있다. 개발자는 사용자가 지도와 상호작용하는 방법을 제어하고, UI 이벤트를 처리해 알맞은 피드백을 줄 수 있다.

### UI설정

UiSettings는 UI와 관련된 설정을 담당하는 클래스이다. UiSettings의 속성을 변경하면 컨트롤, 제스처 등 UI와 관련된 요소를 제어할 수 있다.

### 객체 접근

UiSettings 클래스의 인스턴스는 직접 생성할 수 없고 NaverMap.getUiSettings()를 호출해 가져와야 한다. 하나의 UiSettings 인스턴스 하나의 NaverMap 인스턴스와 1:1로 대응한다.

다음은 NaverMap 으로부터 UiSettings 인스턴스를 가져오는 예제이다.

```
UiSettings uiSettings = naverMap.getUiSettings();
```

### 컨트롤

컨트롤은 지도에 대한 정보 및 간단한 조작 기능을 제공하는 지도 위 버튼을 의미한다. 네이버 지도 SDK는 다음과 같은 여섯 가지 컨트롤을 제공한다.

- 나침반: 카메라의 회전 및 틸트 상태를 표현한다. 클릭하면 카메라의 베어링과 틸트가 0으로 초기화된다. 베어링과 틸트가 0이 되면 자동으로 사라진다.



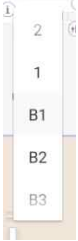
compassEnabled 속성으로 활성화 여부를 지정할 수 있다.

- **축척 바:** 지도의 축척을 표현한다. 지도를 조작하는 기능은 없다.



scaleBarEnabled 속성으로 활성화 여부를 지정할 수 있다.

- **실내지도 층 피커:** 노출 중인 실내지도 구역의 층 정보를 표현한다.



층을 선택하면 해당 층의 실내지도가 노출된다. 실내지도가 보이는 상황에만 나타난다.

indoorLevelPickerEnabled 속성으로 활성화 여부를 지정할 수 있다.

- **현위치 버튼:** 위치 추적 모드를 표현한다. 클릭하면 모드가 변경된다.



location ButtonEnabled 속성으로 활성화 여부를 지정할 수 있다.

단, 현위치 버튼을 활성화하더라도 LocationSource가 지정되지 않으면 컨트롤을 클릭하더라도 위치 추적 기능이 동작하지 않는다.

- **네이버 로고:** 클릭하면 범례, 법적 공지, 오픈소스 라이선스를 보여주는 다이얼로그가 열린다. 네이버 로고는 비활성화할 수 없으나, 위치를 조정하거나 클릭을 비활성화할 수는 있다. 네이버 지도 SDK를 사용하는 앱은 반드시 네이버 로고가 앱의 UI 요소에 가려지지 않도록 해야 한다.

- logoGravity 속성으로 그래비티를, logoMargin 속성으로 마진을 지정해 위치를 조정할 수 있다.
- logoClickEnabled 속성으로 클릭 활성화 여부를 지정할 수 있다. 로고 클릭을 비활성화한 앱은 반드시 앱 내에 네이버 지도 SDK의 법적 공지 액티비티 및 오픈소스 라이선스 액티비티로 연결되는 메뉴를 만들어야 한다.

UiSettings의 속성을 변경하면 각 컨트롤의 활성화 여부를 제어할 수 있다. 비활성화된 컨트롤은 화면에서 사라진다.

다음은 나침반을 비활성화하고 현위치 버튼을 활성화하는 예제이다.

```
uiSettings.setCompassEnabled(false);
uiSettings.setLocationButtonEnabled(true);
```

## 컨트롤 배치

UiSettings로 활성화 여부를 제어하는 기본 컨트롤의 위치는 변경할 수 없다. 그러나 각 컨트롤은 뷰의 형태로 제공되므로 UiSettings에서 기본 컨트롤을 비활성화하고 별도의 컨트롤 뷰를 원하는 위치에 배치할 수 있다. 컨트롤 뷰는 widget 패키지에 있다. 뷰를 레이아웃에 추가하고 map 속성에 지도 객체를 지정하면 컨트롤이 동작한다.

다음은 기본 줌 컨트롤을 비활성화하고 오른쪽 아래에 별도의 줌 컨트롤을 배치하는 예제이다.

```
<FrameLayout ...>
    <FrameLayout
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.naver.maps.map.widget.ZoomControlView
        android:id="@+id/zoom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="12dp"
        android:layout_gravity="end|bottom" />
</FrameLayout>
```

```
FragmentManager fm = getSupportFragmentManager();
MapFragment mapFragment = (MapFragment)fm.findFragmentById(R.id.map);
if (mapFragment == null) {
    NaverMapOptions options = new NaverMapOptions().zoomControlEnabled(false);
    mapFragment = MapFragment.newInstance(options);
    fm.beginTransaction().add(R.id.map, mapFragment).commit();
}
mapFragment.getMapAsync(naverMap -> {
    ZoomControlView zoomControlView = findViewById(R.id.zoom);
    zoomControlView.setMap(naverMap);
});
```

## 2.4 XML 데이터 파싱

확장성 마크업 언어(XML)는 컴퓨터에서 읽을 수 있는 형식으로 문서를 인코딩하기 위한 일련의 규칙을 모은 것이다. XML은 인터넷에서 데이터를 공유하는 데 흔히 사용되는 형식이다. 뉴스 사이트나 블로그와 같이 콘텐츠를 자주 업데이트하는 웹사이트에서는 종종 XML 피드를 제공하여 외부 프로그램이 콘텐츠 변경사항을 잘 파악할 수 있도록 한다. XML 데이터 업로드 및 파싱은 네트워크에 연결된 앱의 일반적인 작업이다.

### XmlPullParser 인터페이스

XmlPullParser는 next(), nextInt() 등의 메소드를 가지고 있는 인터페이스로 본 예제에서는

next() 메소드를 활용해 문서를 순차적으로 읽으며 다음 이벤트를 발생시킬수 있다. getEventType을 활용해서 다음과 같은 이벤트 처리 상태를 얻을수 있다.

상태	의미
START_DOCUMENT	문서의 시작을 의미
END_DOCUMENT	문서의 끝을 의미
START_TAG	태그의 시작 (예:<data>)
END_TAG	태그의 끝 (예:</data>)
TEXT	태그의 시작과 끝 사이 (예:<data>content</data>)

START\_TAG와 END\_TAG는 getName()을 사용해야 하고 TEXT는 getText()를 사용해야 한다. <data>이순신</data>가 있다면 START\_TAG, TEXT END\_TAG가 순차적으로 발생한다. 만약 TEXT인 '이순신'이 없어도 발생하는 이벤트는 동일하다.

또 다른 예로는 <data><data1>content</data1></data>인 경우 START\_TAG, TEXT, START TAG, TEXT, END\_TAG 순으로 이벤트가 발생한다.

다음은 XmlPullParser가 가지고 있는 중요 메소드이다. 다음 메소드를 사용하기 위해서는 먼저 XmlPullParser객체를 생성해야 한다.

메소드	기능
getEventType	START_DOCUMENT,START_TAG,END_TAG,TEXT등 이벤트 유형을 반환한다.
getName	태그명을 가져온다.
getText	태그의 시작과 끝 사이의 데이터를 가져온다.
next	다음 이벤트를 가져온다.

## XmlPullParser 객체 생성하기

XmlPullParserFactory 클래스의 메소드인 newPullParser를 사용해 XmlPullParser 객체를 생성해야 한다. 이때, XmlPullParserFactory 클래스의 객체를 먼저 생성해야 한다. XmlPullParserFactory 클래스의 객체를 생성할 때는 new 연산자가 아닌 newInstance를 사용한다는 점에 유의해야 한다.

```
XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
factory.setNamespaceAware(true);
XmlPullParser xmlpp = factory.newPullParser();
```

## AsyncTask

AsyncTask 클래스를 상속하는 DownloadWebData 클래스를 제작해서 사용하겠습니다. AsyncTask는 백그라운드 쓰레드를 가지고 있어서 별도의 Handler나 Thread없이 쉽게 코드를 작성할 수 있다. 만약 화면을 갱신하는 UI 쓰레드가 통신 작업을 하고 있을 때, 화면을 터치하는 이벤트가 발생하게 된다면 ANR(Application Not Responding) 에러가 발생하게 된다. 즉, 통신 작업을 수행해 줄 다른 쓰레드가 필요하게 된다. 이럴 때, AsyncTask 클래스를 상속받아 사

용하면 이러한 에러 걱정없이 편리하게 사용할 수 있다.

## AsyncTask 실행 단계

AsyncTask 클래스를 활용한 작업 단계는 다음과 같다. 4단계 중 2단계인 doInBackground만 다른 스레드에서 실행되고 나머지는 메인 스레드에서 실행된다. 본 예제에서는 2단계와 4단계만 사용한다.

1단계	onPreExcute	doInBackground 단계 전에 UI Thread에 의해서 실행되며 Background 작업 진행 정보에 대한 프로그래스바를 표시할 경우 사용된다.
2단계	doInBackground	onPreExecute가 끝난후에 호출되며 실제로 Background 스레드에서 작업을 수행한다. 이때 작업에 필요한 값을 execute에서 파라미터를 전달받는다. 본 예제에서는 downData 메소드를 호출하여 웹 문서를 다운로드하는 기능을 수행한다.
3단계	onProgressUpdate	doInBackground 작업 수행 중에 호출되는 publishProgress 실행 후, UI Thread에 의해서 호출된다. 주로 doInBackground 작업의 진행 정도를 표시하기 위한 프로그래스바의 업데이트 등에 사용된다.
4단계	onPostExecute	doInBackground 작업이 끝나면 메인 스레드에 의해 실행된다. doInBackground 수행 후 result 값을 받아 XmlPullParser를 사용해서 데이터를 처리한다.

## AsyncTask 사용법

AsyncTask 클래스를 상속하는 DownloadWebContent 클래스 객체를 생성하고 객체명.execute를 사용하여 AsyncTask를 작동시킨다.

```
DownloadWebContent dwcl = new DownloadWebContent();
dwcl.execute(strUrl);
```

## doInBackground 단계 - 웹 서버에서 가져온 문서를 화면에 표시하기

doInBackground 단계에서 실행되는 downloadByUrl 메소드에서는 웹과 연결하여 데이터를 Byte 단위로 받고, 이를 문자로 변환시키는 역할을 수행한다. 다음은 이를 처리하기 위해 사용된 클래스이다.

클래스	기능
HttpURLConnection	웹에서 데이터를 주고받기 위한 통신 규약 기능을 가지고 있다. URL을 사용해 참조하는 자원에 대해 읽고 쓰는 일을 할 수 있다. openConnection 메소드를 사용해 웹과 연

	결하고 BufferedInputStream 클래스를 연결된 URL의 소스를 Byte 단위로 다운받는다.
InputStreamReader	Byte 단위의 데이터를 문자로 변환
BufferedReader	문자 단위로 데이터를 버퍼에 저장하는 역할. 버퍼를 사용하면 웹 소스와 직접 작업을 하지 않고 중간에 버퍼(=메모리)를 사용하므로 속도가 매우 향상된다

```

public String downloadByUrl(String myurl) throws IOException {
    HttpURLConnection conn = null;

    try {
        URL url = new URL(myurl);
        conn = (HttpURLConnection) url.openConnection();
        BufferedInputStream buffer = new BufferedInputStream(conn.getInputStream());
        BufferedReader buffer_reader = new BufferedReader(new InputStreamReader(buffer, "utf-8"));
        String line = null;
        String getData = "";

        while((line = buffer_reader.readLine()) != null) {
            getData += line;
        }

        return getData;
    } finally {
        conn.disconnect();
    }
}

```

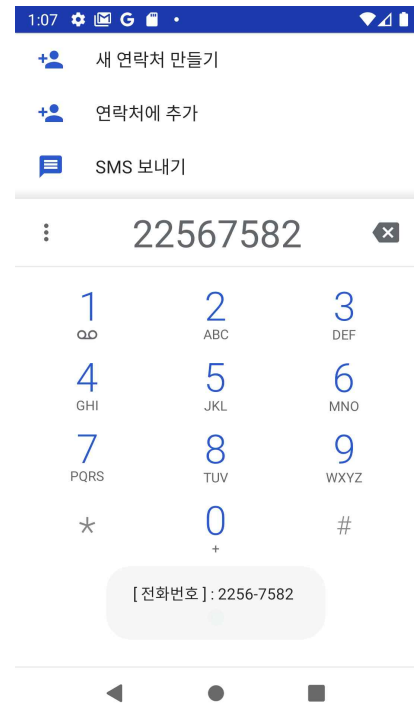
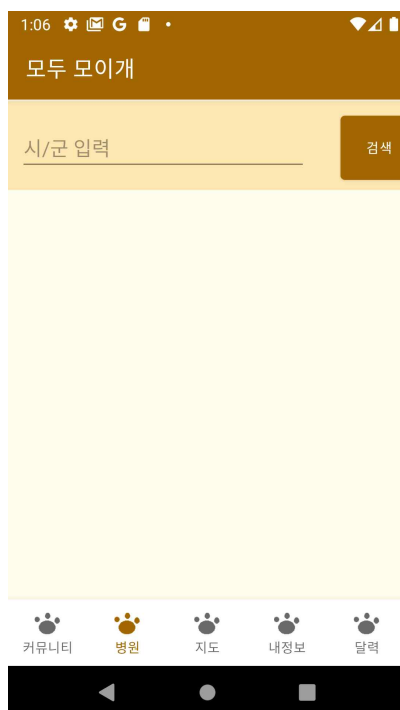
파일 입력과 관련된 클래스와 메소드는 다음과 같다. 여기서 Stream은 한쪽 방향에서 다른 방향으로 통신을 연결하고 데이터를 전송해주는 것이다. Stream은 Byte단위로 데이터를 전송한다.

클래스 및 메소드	기능
BufferedInputStream	Byte 단위로 데이터를 전송한다. InputStream의 자손 클래스이며 버퍼를 사용하기 때문에 입출력 속도가 우수하다.
BufferedReader	문자 단위로 데이터를 전송한다. InputStream 대신에 Reader를 쓰면 문자 단위로 데이터를 전송한다는 의미이다. Java는 C와 다르게 문자형 char가 2Byte이므로 Byte 기반의 Stream으로 2Byte를 처리하기 어렵다. 문자 형태의 웹 데이터를 사용하기 위해서 InputStreamReader를 사용해서 문자형 데이터로 변환시키고 BufferedReader 클래스를 사용해 버퍼에 자료를 담는다.
InputStreamReader	Byte 기반의 Stream 데이터를 지정된 인코딩 문자 데이터로 변환해주는 작업을 한다.
readLine()	BufferedReader 클래스의 메소드로 파일을 Line(줄) 단위로 읽어준다.



## 실사용한 xml 데이터와 코드

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <header>
    <resultCode>0000</resultCode>
    <resultMsg>OK</resultMsg>
  </header>
  <body>
    <items>
      <item>
        <rnum>1</rnum>
        <publisher>SEON08</publisher>
        <creator>서울시</creator>
        <collectionDb>서울시 열린데이터광장 DB</collectionDb>
        <title>(주)동물의료영상센터</title>
        <subjectCategory>동물병원</subjectCategory>
        <reference/>
        <source/>
        <insertDate>2020-12-16 09:48:21.0</insertDate>
        <venue>서울특별시 강남구 포이동 231번지 6 호 민빌딩 2층 </venue>
        <spatial/>
        <description/>
        <subDescription/>
        <state>폐업</state>
        <affiliation>강남구</affiliation>
      </item>
      <item>
        <rnum>2</rnum>
        <publisher>SEON08</publisher>
        <creator>서울시</creator>
        <collectionDb>서울시 열린데이터광장 DB</collectionDb>
        <title>(주)러브펫동물병원</title>
        <subjectCategory>동물병원</subjectCategory>
        <reference>02-2634-8275</reference>
        <source/>
        <insertDate>2020-12-16 09:48:21.0</insertDate>
        <venue>서울특별시 영등포구 영등포동4가 442번지 타임스퀘어 지하1층 (B102) </venue>
        <spatial/>
        <description/>
        <subDescription/>
        <state>폐업</state>
        <affiliation>영등포구</affiliation>
      </item>
      <item>
        <rnum>3</rnum>
        <publisher>SEON08</publisher>
        <creator>서울시</creator>
        <collectionDb>서울시 열린데이터광장 DB</collectionDb>
        <title>(주)아크리스동물병원</title>
        <subjectCategory>동물병원</subjectCategory>
        <reference>583-7582</reference>
      </item>
    </items>
  </body>
</response>
```



```

public void onClick(View view) {
    switch (view.getId()) {
        case R.id.button:
            //네트워크를 통해서 xml문서를 읽어오기..
            (Thread) run() → {

                items.clear();
                String str = edit.getText().toString();
                String location = URLEncoder.encode(str);
                String adress = "http://api.kcisa.kr/openapi/service/rest/convergence2019/getConver03?serviceKey=c4f35" +
                    "%7b-9b6d-4fc5-9a23-3f1749cccec4&pageNo=1&numOfRows=15&keyword=동물병원&where=" + location;

                try {

                    URL url = new URL(adress);

                    InputStream is = url.openStream(); //바이트스트림
                    InputStreamReader isr = new InputStreamReader(is);

                    XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
                    XmlPullParser xpp = factory.newPullParser();
                    xpp.setInput(isr);

                    int eventType = xpp.getEventType();

                    String tagName;
                    StringBuffer buffer = null;

                    while (eventType != XmlPullParser.END_DOCUMENT) {

                        switch (eventType) {
                            case XmlPullParser.START_TAG:

```



```

        tagName = xpp.getName();
        if (tagName.equals("item")) {
            buffer = new StringBuffer();

        } else if (tagName.equals("title")) {
            buffer.append("\n [ 병원 이름 ] : ");
            xpp.next();
            buffer.append(xpp.getText() + "\n");

        } else if (tagName.equals("venue")) {
            buffer.append(" [ 주소 ] : ");
            xpp.next();
            buffer.append(xpp.getText() + "\n");

        } else if (tagName.equals("reference")) {
            buffer.append(" [ 전화번호 ] : ");
            xpp.next();
            buffer.append(xpp.getText() + "\n");

        } else if (tagName.equals("state")) {
            buffer.append(" [ 영업 상태 ] : ");
            xpp.next();
            buffer.append(xpp.getText() + "\n");
        }
        break;

    case XmlPullParser.TEXT:
        break;

    case XmlPullParser.END_TAG:
        tagName = xpp.getName();
        if (tagName.equals("item")) {

```

```

AdapterView.OnItemClickListener listener = new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        String str = (String) adapterView.getAdapter().getItem(i);
        int nIdx = str.indexOf("[ 전화번호 ]");
        int nIdx2 = str.indexOf("[ 주소 ]");
        String str1 = str.substring(nIdx+11,nIdx2-1);

        Toast.makeText(getActivity(), str.substring(nIdx,nIdx2-1), Toast.LENGTH_SHORT).show();

        Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:"+str1));
        startActivity(intent);
    }
};

```

# 3. 안드로이드 스튜디오 깃허브 협업

## 1. Git 설치

## 2. 안드로이드 스튜디오 설정

File - Settings - Version Control - Git

Path to Git executable - Test 클릭을 통해서 Git 설치 유무와 버전 확인

File - Settings - Version Control - GitHub

깃허브에 로그인

△404/401 에러 시에 깃허브 홈페이지에서 토큰 발급 후 토큰으로 로그인

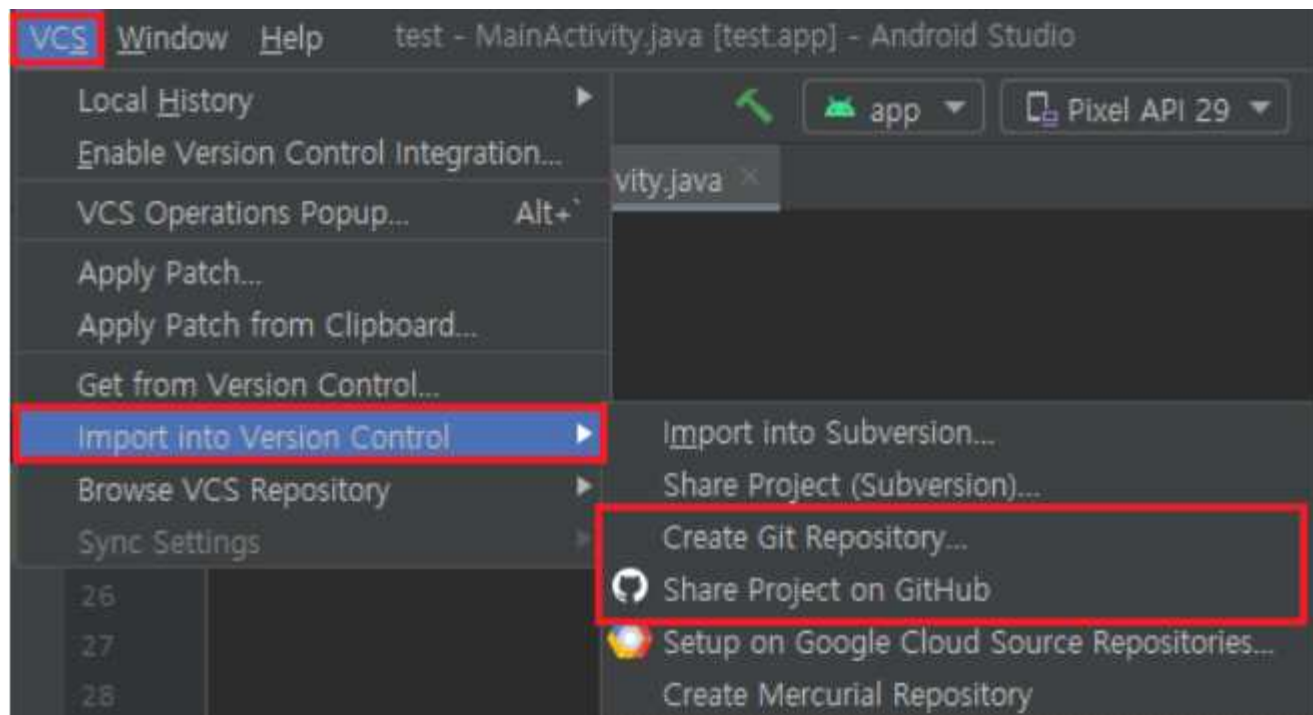
토큰 발급

깃허브 홈페이지 - 프로필 - Settings - Developer settings - Personal access token

[repo / admin:org / gist] 체크 후 발급(하위 속성까지 전부 체크)

## 3. 깃허브에 프로젝트 업로드

상단 VCS 탭



### 1. Create Git Repository

좌측 프로젝트 영역 파일들이 빨강게 바뀐

### 2. Share Project on GitHub

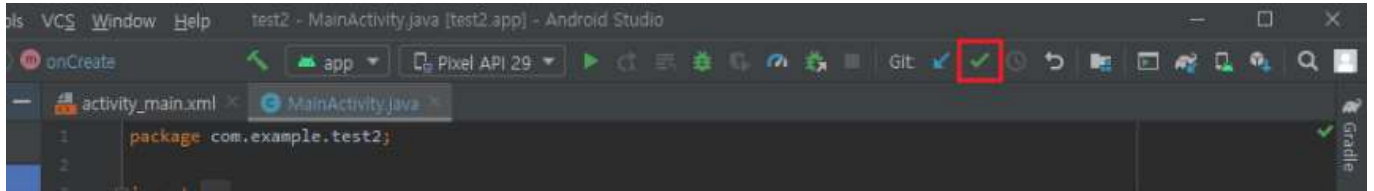
Repository에 올릴 내용을 담는 작업

Repository 이름 설정, private 설정 후

commit message 설정 - Add

### 3. 깃허브에 업로드 완료

#### 4. 수정한 프로젝트 업데이트



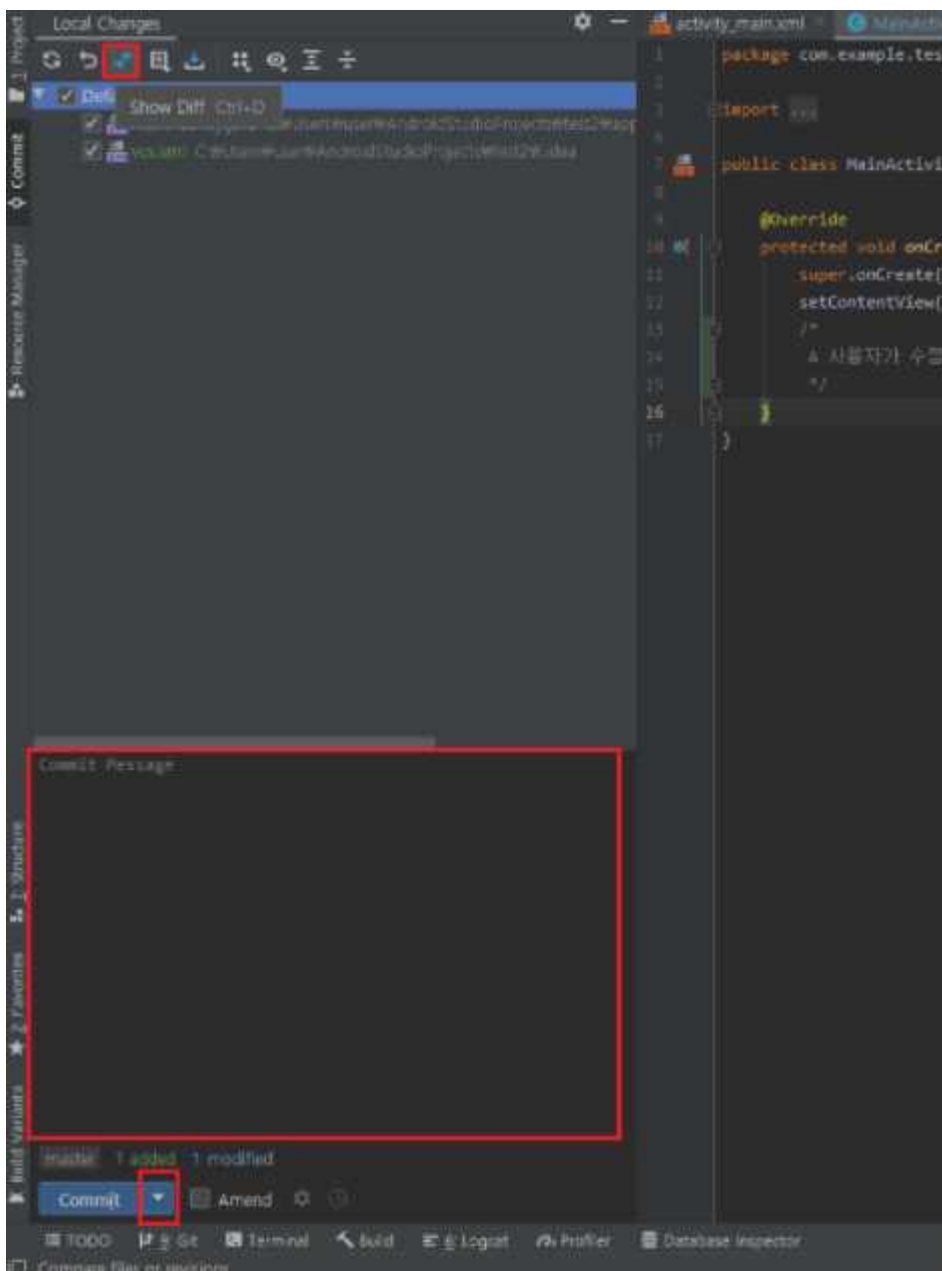
프로젝트 수정 후 업데이트 시 사용법

3가지 방법

상단 Git - 체크 표시

control + k 단축키

상단 메뉴 VCS - Git - Commit Files... 선택



좌측 Commit 메뉴에서 Show Diff(Ctrl + D)를 통해 변경내용을 아주 보기 좋게 확인 가능

이후 Commit Message 입력 후 Commit 말고 Commit 우측 삼각 메뉴를 눌러 Commit and Push를 선택

이후 깃허브 상에서 변경내용 확인 가능

## 5. 협업과정

깃허브 Repository - Settings - Manage access - Invite a collaborator 협업자 초대  
를 통한 Collaborator가 프로젝트 Access 가능

협업자(Collaborator) 기준 프로젝트 받아오기

### Manage access

Add people


☐ Select all Type ▾

☐

**KKS**  
kuchic0918 • Collaborator



☐

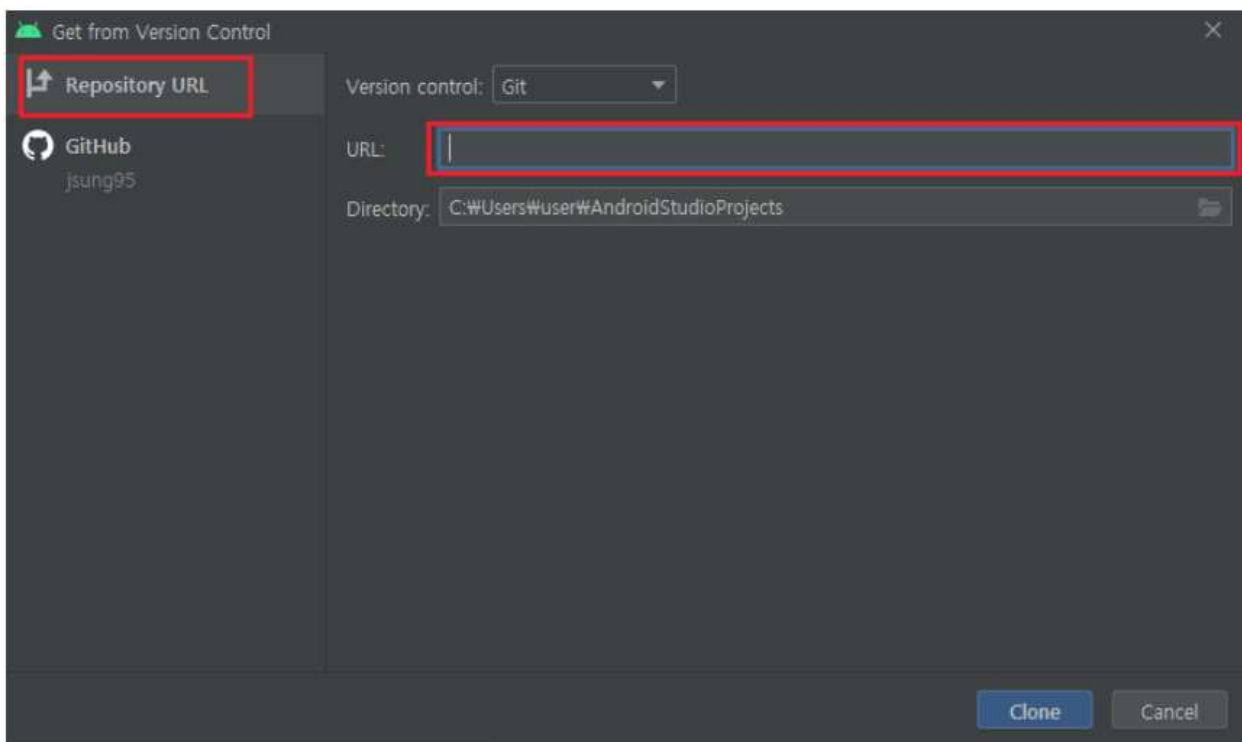
**yelim0308**  
Collaborator



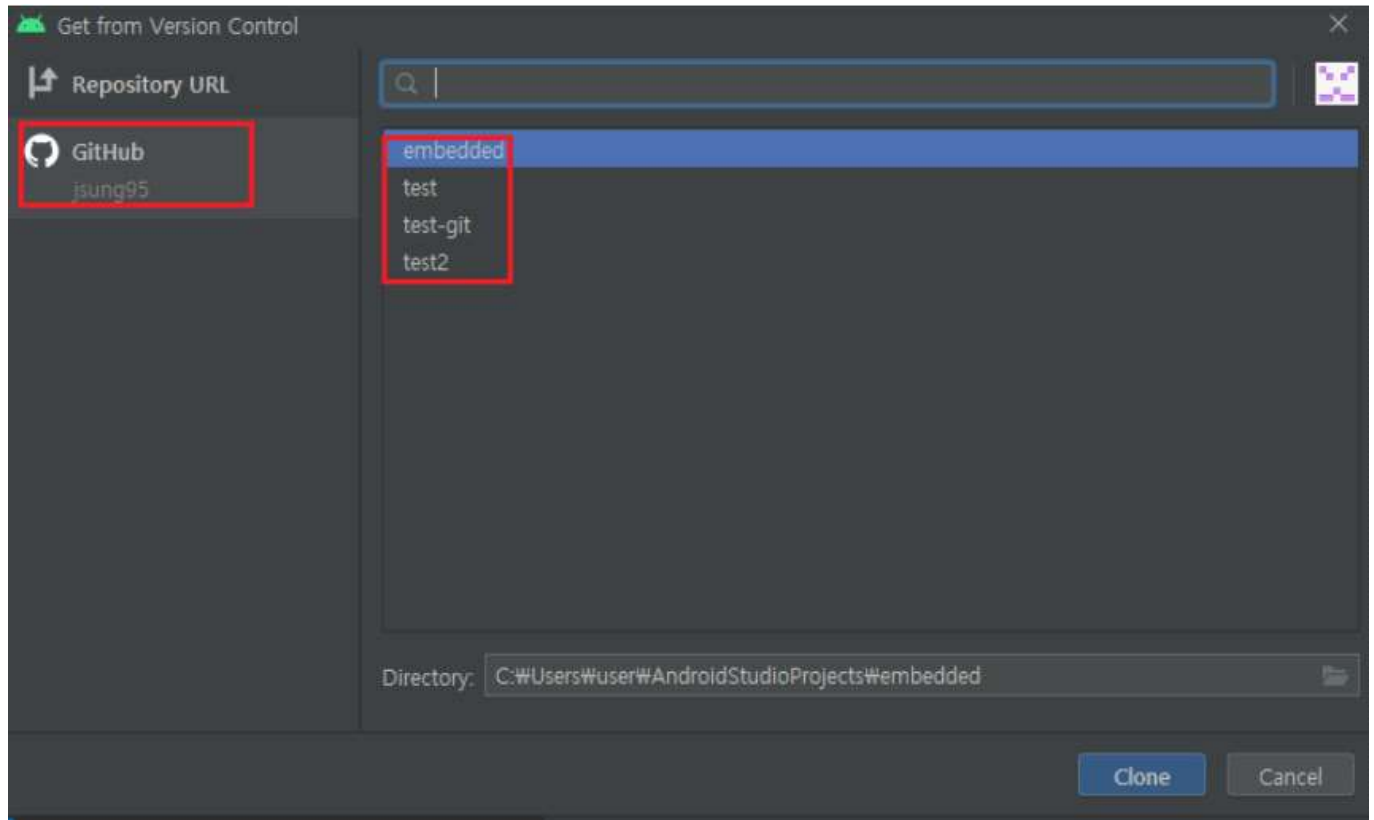
< Previous   Next >

깃허브 초대 수락 후 Repository 내에서 Clone URL 복사

이후 마찬가지로 안드로이드 스튜디오에서 Git/Github 설정 후 첫 시작화면(Close Project를 닫  
은 화면)에서 우측 Get from Version Control 메뉴 선택



복사했던 Repository Clon URL 입력 또는 GitHub 탭에서 repository 선택 가능



## 6. 협업자(Collaborator)가 코드 수정 후 -> 업데이트(Commit and Push)

협업자가 수정하던 프로젝트 Master가 수정하던 모두 마찬가지로 Commit and Push로 깃허브에 업데이트

## 7. 수정내용(변경사항)을 협업 프로젝트에 최신으로 업데이트



상단 Git - Update Project를 통해 빠르게 업데이트 또는 상단 메뉴 VCS - Git - Pull을 통해 업데이트 가능