

Hanbit RealTime

유동환 지음





# 안드로이드를 위한 Gradle 안드로이드 스튜디오를 이용한 빌드 자동화

**유동환** 지음





표지 사진 **김은숙** 이 책의 표지는 김은숙 님이 보내 주신 풍경사진을 담았습니다. 리얼타임은 독자의 시선을 담은 풍경사진을 책 표지로 보여주고자 합니다. 사진 보내기 ebookwriter@hanbit.co.kr

#### 안드로이드를 위한 Gradle 안드로이드 스튜디오를 이용한 빌드 자동화

**초판발행** 2016년 6월 16일

지은이 유동환 / **펴낸이** 김태헌 **펴낸곳** 한빛미디어(주) / **주소** 서울시 마포구 양화로7길 83 한빛미디어(주) IT출판부 전화 02-325-5544 / 팩스 02-336-7124 등록 1999년 9월 30일 제10-1779호 ISBN 978-89-6848-824-5 95000 / **정가** 10,000원

총괄 전태호 / 책임편집 김창수 / 기획·편집 정지연 디자인 표지 강은영, 내지 여동일, 조판 최송실 마케팅 박상용, 송경석, 변지영 / 영업 김형진, 김진불, 조유미

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오. 한빛미디어 홈페이지 www.hanbit.co.kr / 이메일 ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea Copyright ⓒ 2016 유동환 & HANBIT Media, Inc. 이 책의 저작권은 유동환과 한빛미디어(주)에 있습니다. 저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다. 책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요. 한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

#### 지은이\_ 유동환

생각을 즐기는 프로그래머. 유동의 브런치(https://brunch.co.kr/@yudong)를 운 영 중이며 LG전자에서 안드로이드 앱을 개발하고 있습니다. 최근에는 IoT 관련 프 로젝트를 하며 바쁜 날들을 보내고 있습니다. 스무 살 때부터 자바카페와 JCO(한국자 바개발협의회) 커뮤니티에서 수년간 활동하며 소심했던 성격도 고치고 누군가에 도움 이 되는 사람이 되려고 노력 중입니다. 사내에서는 독서습관클럽을 만들어 운영한 지 일 년이 되어 가고 있습니다. 연세대학교 정보대학원에서 경영정보학을 전공하였고 『Professional Java Web Services』(정보문화사, 2002)를 공역하였습니다. 2014년 어느 날 새로운 개발팀에 들어가 보니 낯선 개발 환경이 눈에 들어왔습니다. 첫 번째는 이클립스가 아니라 안드로이드 스튜디오<sup>Android Studio</sup>라는 인텔리제이<sup>Intellij</sup> 기 반의 IDE였고, 두 번째는 빌드 도구인 Gradle이었습니다.

다른 팀에서 사용하지 않는 생소한 도구 때문에 처음 3개월간 매우 불편하였지만 이내 익숙해졌고, 이후 새로운 앱을 개발하면서 Gradle 덕분에 십여 개의 다른 개발팀과 협업을 원활하게 진행하였습니다. 이런 추억을 되돌아보니 프로그래밍 언어뿐만 아 니라 지속해서 새로운 개발 도구를 도입해야 개발 생산성이 향상될 수 있음을 배웠습 니다.

이 책은 안드로이드 개발자가 활용하는 Gradle에 대해 다룹니다. 그동안 Gradle에 관한 책은 몇 권 있었지만, 안드로이드 개발자가 바로 따라 해보기는 어려웠습니다. 그 래서 필자가 실무에서 겪은 내용을 알기 쉽게 풀어내려고 최대한 노력하였습니다. 예 를 들어, 부서에 신입사원이 들어온다면 어떤 순서로 Gradle을 설명할 수 있을까 하고 여러 번 자신에게 되물으면서 만든 산출물입니다.

처음 집필할 때는 안드로이드 스튜디오가 1.4 버전이었는데, 이제는 2.0 버전을 넘어 2.1.2 버전까지 나왔습니다. 그동안 많은 변화가 있었지만 2.0 버전부터는 안정 버전 이라고 생각해도 무방합니다.

이 작은 책을 쓰는 데도 많은 사람의 도움을 잊을 수 없습니다. 매주 일요일 함께 글 쓰 고 피드백을 해준 친구 유현석과 자바카페 강사준비팀 3기 멤버들, 그리고 투박했던 초고를 매끄럽게 다듬어주신 한빛미디어 여러분(특히 정지연님), 마지막으로 사랑하는 아내 지영에게 감사의 말을 전합니다.



이 책은 안드로이드 개발 경험이 있는 개발자 중에서 안드로이드 스튜디오와 Gradle 의 기본적인 내용을 빠르게 학습하고자 하는 독자를 대상으로 합니다. 간단한 안드로 이드 앱을 만들어본 경험이 있다면 누구든 이 책을 이해하는 데 문제가 없습니다. 책에 있는 예제는 Hello World와 간단한 버튼이 있는 UI 정도입니다.

Gradle의 백미는 멀티 프로젝트의 활용입니다. 현재 이클립스로 진행 중인 개발 프로 젝트가 있다면 단지 Gradle로 동일하게 변환하는 데 만족하지 말고 독립적인 기능을 별도 모듈로 구성하여 멀티 프로젝트로 구성해 보기를 추천합니다. Gradle 내부를 이 해하는 데 큰 도움이 됩니다.

안드로이드 스튜디오 버전은 2.0을 기반으로 하고 있습니다. 그 이전 버전에서는 책에 서 언급한 내용이 동일하게 동작하지 않을 수 있습니다.



#### chapter 1 안드로이드 스튜디오와 Gradle 009

- 1.1 안드로이드 스튜디오 ---- 009
  - 1.1.1 안드로이드 스튜디오의 기능 ----- 010
  - 1.1.2 안드로이드 스튜디오의 장점 \_\_\_\_\_ 012
- 1.2 Gradle 014
- 1.3 안드로이드를 위한 Gradle의 특징 ----- 016

#### chapter 2 첫 Gradle 프로젝트 생성하기 ----- 019

- 2.1 Hello World 프로젝트 생성 ----- 019
- 2.2 프로젝트 build.gradle ---- 025
- 2.3 모듈 build.gradle ---- 026

#### chapter 3 Gradle 태스크와 생명주기 ----- 029

- 3.1 프로젝트 빌드하고 실행하기 ---- 029
- 3.2
   Gradle 태스크 개념잡기
   034

   3.2.1
   태스크에서 로그 출력하기
   034

   3.2.2
   태스크에 그룹 지정하기
   036

   3.2.3
   태스크에 설명 지정하기
   038

   3.2.4
   태스크에 의존성 지정하기
   039
- 3.3 Gralde의 생명주기 ---- 040

#### chapter 4 외부 라이브러리 추가 043

- 4.1 가장 기본적인 방법 ----- 044
- 4.2 그 외 다양한 방법 ----- 045
- 4.2.1 다른 모듈의 소스 코드 참조하기
   045

   4.2.2 디버그 모드와 릴리스 모드에 맞게 참조하기
   046

   4.2.3 안드로이드 유닛 테스트 참조하기
   047

   4.2.4 공개 라이브러리 참조하기 JAR 파일
   047

   4.2.5 공개 라이브러리 참조하기 AAR 파일
   048

   4.2.6 로컬에 있는 AAR 파일 참조하기
   049

   4.2.7 .so(JNI) 파일 참조하기
   050
- 4.3 Omni Notes 사례 ---- 050

#### chapter 5 멀티 프로젝트 활용 055

- 5.1 Android 뷰와 Project 뷰 ----- 055
- 5.2 안드로이드 라이브러리 모듈 만들기 ----- 058
- 5.3
   로컬 저장소 만들기
   061

   5.3.1
   로컬 저장소는 언제 필요한가
   061

   5.3.2
   로컬 저장소 만드는 방법
   062

#### chapter 6 테스트 ---- 067

- 6.1 로컬 유닛 테스트 ---- 068
- 6.2 안드로이드 테스트 ---- 071
- 6.3 Espresso 연동하기 ---- 074
- 6.4 로컬 유닛 테스트의 제약사항 ---- 077

#### chapter **7** 빌드 변형 ----- 079

- 7.1 빌드 타입 ----- 079
- 7.2 앱 서명 첨부하기 ---- 081

chapter 8 고급 활용 ----- 095

- 8.1 확장 플러그인
   095

   8.1.1 CheckStyle 플러그인
   096

   8.1.2 Retrolambda 플러그인
   099

   8.1.3 람다 표현식 사용해보기
   100

   8.2 리소스 줄이기
   101
- 8.3
   기타 유용한 팁
   102

   8.3.1 APK 생성 시 버전명 자동 기입하기
   102

   8.3.2 생성된 APK를 특정 폴더로 복사하기
   103

#### 부록 A Git 콘솔에서 Gradle 활용하기 ----- 105

- A.1 최신 소스 코드 받아 바로 빌드하기 ----- 105
- A.2 클린하고 다시 빌드하기 ---- 107
- A.3 빌드하여 타깃에 바로 APK 설치하기 ----- 108
- A.4 라이브러리 의존성 확인하기 ----- 109

# 안드로이드 스튜디오와 Gradle

chapter 1

1장에서는 안드로이드 스튜디오라는 IDE(통합개발환경)와 빌드 도구인 Gradle을 소개합니다. 안드로이드 환경에서의 Gradle은 IDE를 제외하면 설명하기 힘들 정도로 밀접하게 연관되어 있습니다. 또한, 기존 이클립스 개발 환경과 비교해도 많은 것이 달라졌습니다. 안드로이드 스튜디오가 Gradle과 관련하여 어떤 편의 기능을 제공하는지 알아보고, Gradle의 개념과 주요 특징, 이클립스와 비교하여 폴더 구조가 어떻게 달라졌는지를 살펴보겠습니다.

# 1.1 안드로이드 스튜디오

2015년 구글은 기존의 이클립스 ADT<sup>Android Development Toolkit</sup> 지원을 중단하고 안 드로이드 스튜디오를 공식 IDE로 선언하였습니다(2016년 현재 ADT는 구글 사이트에 서 다운로드할 수 없습니다). 안드로이드 스튜디오는 JetBrain사의 인텔리제이 기반 IDE로, 강력한 검색 기능과 다양한 플러그인을 지원합니다.

필자는 2014년 새로운 개발팀에 합류하면서 안드로이드 스튜디오 1.0을 처음 사 용해볼 수 있었습니다. 그전까지는 이클립스를 사용하고 있었는데, 말 그대로 너 무 달랐습니다. IDE의 전체 외관과 단축키, Gradle이라는 새로운 빌드 도구까지 모든 것이 달라서 익숙해지는 데 3개월은 걸렸습니다. 이제는 너무 잘 사용하고 있고 이클립스로 다시 돌아갈 일은 없을 것 같습니다. 안드로이드 스튜디오의 최신 버전은 2.1.2입니다. 그동안 구글에서 지속적으로 업데이트해 왔지만, Windows 용 1.2 버전까지는 속도도 느리고 메모리도 많이 사용한 것이 사실입니다. 1.3 버전부터 드라큘라<sup>Dracula Theme</sup>도 지원하고 안정성도 대폭 향상되어 그야말로 안드로이드 스튜디오 시대가 열렸다고 할 수 있습니다. 이 책에서는 Android Studio 2.0을 기준으로 특징을 간단히 설명하겠습니다.

# 1.1.1 안드로이드 스튜디오의 기능

가장 먼저 눈에 띄는 것은 IDE의 좌우와 하단의 다양한 기능 버튼으로, 이클립스 보다 더 많은 것을 지원하고 있습니다. 여기서는 현업 개발자가 자주 사용하는 기 능들을 정리해 보겠습니다.



그림 1-1 안드로이드 스튜디오 기본 화면

# 좌측

Project 프로젝트에 포함된 파일들을 다양한 관점에서 요약하여 보여줍니다.
 안드로이드 개발자에게는 Android가 가장 편리하며, 실제 파일 구조를 보기

위해서는 Project로 변경합니다. 이클립스가 익숙한 개발자는 Project로 시 작하고 점차 Android를 활용하면 됩니다.

- Captures 필요한 스크린 샷을 캡처하는 기능으로, 실제 타깃을 연결하여 개 발하는 경우 유용합니다. 이클립스와 달리 캡처한 이미지를 IDE에서 바로 볼 수 있어 편리합니다.
- 3. Build Variants 모듈별 빌드 변형<sup>Build Variants</sup>이 표시됩니다. 이는 중급 이상의 내용으로 7장에서 상세하게 다루겠습니다.

# 하단

- Terminal 내장된 명령창입니다. 보통 IDE만으로 충분히 빌드할 수 있지만, 실무에서는 Terminal 혹은 git console을 함께 띄워놓고 빌드하는 것이 일 반적입니다. 콘솔에서 IDE 없이 빌드하는 방법을 반드시 익혀야 합니다.
- Android Monitor 기존에는 'Android Logcat'이라는 이름이었습니다. 이 클립스와는 달리 여러 대의 타깃이 연결되어 있어도 각각 편리하게 로그와 메 모리 현황 등을 볼 수 있으며, 기능이 점점 늘어날 예정입니다(더 자세한 내용은 안드로이드 스튜디오 관련 서적에서 참고하시기 바랍니다).
- 3. TODO 이클립스와 마찬가지로 //TODO, //FIXME 등의 목록을 볼 수 있습 니다.
- 4. Event Log 실행 상태를 표시하는 창입니다. 앱 실행 또는 test 실행 시 함께 띄워놓고 보면 좋습니다.
- 5. Gradle Console Gradle 태스크<sup>Task</sup> 실행 결과를 표시하는 곳입니다. Gradle이 익숙해질 때까지는 자주 살펴봐야 하는 곳입니다.

# 우측

 Gradle 프로젝트와 각 모듈에 포함된 Gradle 태스크를 표시합니다. 태스크 는 android, build, install과 같이 항목별로 구별되어 있습니다.  Android Model Android Studio 1.4에서 새로 포함된 내용입니다.
 Gradle의 전체 내용을 카테고리별로 요약하여 표시합니다. build tools 버 전이나 타깃 SDK 버전 등을 알아볼 수 있습니다.

# 1.1.2 안드로이드 스튜디오의 장점

사용자 입장에서 체험한 안드로이드 스튜디오의 눈에 띄는 특징은 다음과 같습 니다.

# 1. 어디서나 검색 기능

안드로이드 스튜디오의 검색을 한마디로 표현하면 [Shift + Shift]입니다. 어떤 파일을 찾을 때 Shift 키를 두 번 누르면(더블 시프트) 다음과 같이 검색창이 뜨고 그 아래에는 최근에 검색한 파일과 검색결과가 표시됩니다. 필요한 내용을 빠르게 찾아주는 마성의 기능입니다. 이클립스를 오랫동안 활용하여 안드로이드 스튜디 오에 익숙하지 않으신 분들은 자주 사용해보기 바랍니다. 이클립스를 실행했을 때 도 무의식적으로 더블 시프트를 누르는 자신을 발견할 수 있습니다.

그림 1-2 안드로이드 스튜디오 내 검색



#### 2. 강력한 Code Assistant

Code Assistant는 코드를 작성할 때 유사한 변수나 메서드를 추천해주는 기능 입니다. 이클립스의 경우 정확하게 철자가 맞을 때만 코드를 추천했는데, 안드로 이드 스튜디오는 일부만 입력해도 추천의 정확도가 높습니다. 이클립스는 고정된 검색 기법을 가지고 있지만, 안드로이드 스튜디오의 Code Assistant는 매우 유 연합니다.

#### 3. Gradle과 완벽한 통합

이클립스의 빌드 도구는 Ant인데, 일반적인 개발자에게는 그 활용 여부가 감춰져 있습니다. IDE에서 Ant 태스크를 실행하여 빌드와 앱 서명<sup>singing</sup> 등의 업무를 할 수 있다는 장점도 있지만, 감춰져 있는 만큼 빌드 스크립트를 변경하는 경우 IDE 와 밀접하게 통합되어 있지 않아 불편합니다.

안드로이드 스튜디오는 빌드 도구인 Gradle과 전면적으로 통합되어 있습니다. [그림 1-1]에서도 보듯이 소스 코드와 함께 Gradle의 필수 요소인 build.gradle 과 settings.gradle 파일 등이 별도 카테고리에 구별되어 있고, 기본 화면의 좌 우와 하단에도 Gradle 관련 사항을 편리하게 열람할 수 있는 다양한 기능이 내 장되어 있습니다. 마치 Gradle을 알아야 안드로이드 개발을 할 수 있다는 느낌을 줄 정도입니다. 이에 관한 자세한 내용은 뒤에서 살펴보겠습니다.

#### 4. 편리한 단축키

안드로이드 스튜디오의 단축키는 이클립스의 단축키보다 누르기 편합니다. 안드 로이드 스튜디오는 F3과 같은 키보드 상단의 기능 키의 활용을 최소화하여 소스 코드를 입력하는 도중에 흐름을 끊기지 않고 단축키를 누를 수 있습니다. 안드로 이드 스튜디오에서 자주 사용하는 단축키 목록은 다음과 같습니다.

- Shift + Shift 어디서나 검색 기능
- Ctrl + B 선언부로 이동

- Ctrl + Alt+ B 구현부로 이동
- Alt + F7 함수가 어디서 사용되었는지 검색
- Shift + F6 이름 바꾸기
- Ctrl + Alt + 〈LEFT〉/〈RIGHT〉 이전(이후)에 사용했던 커서로 이동
- Alt + 〈LEFT〉/〈RIGHT〉 탭 좌우로 이동

지금까지 안드로이드 스튜디오에 대해 간단히 알아보았습니다. Gradle은 안드로 이드 스튜디오라는 IDE와 밀접하게 통합되어 있으므로 IDE를 잘 아는 것이 무엇 보다 중요합니다. 가능한 이클립스의 기억은 최대한 잊고 안드로이드 스튜디오 스 타일에 익숙해지기를 바랍니다.

# 1.2 Gradle

Gradle은 Gradle사에서 만든 범용 빌드 도구입니다. 안드로이드에서 빌드란 간단하게 말하면 JAR<sup>Java Archive</sup> 파일, AAR<sup>Android Archive</sup> 파일, AP<sup>KAndroid Application <sup>Package</sup> 파일을 만드는 방법입니다. Gradle은 안드로이드 앱 뿐만 아니라 Java, C/C++, Groovy 등의 범용 언어를 모두 지원합니다. 한마디로 Gradle을 한번 배워두면 다른 언어로 개발하더라도 빌드 스크립트를 처음부터 다시 작성할 필요 없이 재사용할 수 있습니다.</sup>

Gradle 홈페이지에서 소개한 Gradle의 주요 특징은 다음과 같습니다.<sup>01</sup>

# 1. 폴리글랏 빌드<sup>Polyglot Build</sup>

링크드인<sup>Linkedin</sup>은 60개의 서로 다른 언어(Java, Scala, Phython, C/C++ 등)를 Gradle을 사용하여 빌드하고 관리한다고 합니다. Gradle에서는 각 언어를 플 러그인으로 구별하는데, 예를 들어, 일반 Java는 java, Java 웹 프로젝트는 war, 안드로이드 앱은 com.android.application 플러그인을 사용하면 됩니다.

<sup>01</sup> http://gradle.org/whygradle-build-automation/

Gradle은 이클립스, 안드로이드 스튜디오, IntelliJ와 같은 IDE에서 편리하게 사용할 수 있도록 창<sup>Window</sup>를 제공하고 있고 CI<sup>Continuous Integration</sup> 도구인 젠킨스 <sup>Jenkins</sup>와도 함께 활용할 수 있습니다. 예를 들어, 소스 코드가 Git에 업로드되면 서 버에서 CheckStyle, FindBugs 등의 플러그인을 활용하여 소스 코드가 잠재적 으로 가진 문제를 검출하여 개발자에게 통보하거나 위험한 코드를 merge할 수 없도록 강제할 수 있습니다.

# 3. 외부 라이브러리 관리 자동화

Gradle의 또다른 장점 중 하나는 개발자가 더는 외부 라이브러리를 관리하지 않 아도 된다는 점입니다. 이클립스에서 개발하는 경우 libs 폴더에 원하는 외부 라 이브러리 파일을 직접 복사하였지만, Gradle에서는 단순히 외부 저장소 위치와 라이브러리의 그룹, 이름, 버전 등을 지정해주면 알아서 다운로드하고 빌드에 포 함시킵니다.

# 4. 고성능 빌드

Gradle은 점진적인 빌드Incremental builds, 빌드 캐싱build caching, 병렬 빌드parallelization 기능을 지원하여 고성능 빌드를 추구합니다. 하지만 실제로 윈도우 환경에서 안 드로이드 앱을 빌드해보면 Gradle의 메모리 사용량이 많고 빌드 시 CPU 점유율 이 높습니다. 실무에서 권장하는 사양은 메모리 8GB와 SSD 장착입니다. 메모리 4GB에서는 모듈의 개수가 늘어나면 Out of Memory 오류가 종종 발생하기도 합니다. 이 부분은 향후 개선될 것으로 생각합니다.

# 1.3 안드로이드를 위한 Gradle의 특징

구글은 2014년 구글 I/O에서 새로운 빌드 시스템을 발표하였습니다. 그 목표는 다음과 같습니다.<sup>02</sup>

- 코드와 리소스를 재활용하기 쉽게 한다.
- 다양한 변형<sup>variants</sup>을 만들 수 있고 앱의 특성<sup>flavors</sup>에 따라 단일 소스에서 여러 APK를 만 들 수 있게 한다.
- 빌드 절차를 쉽게 설정하고 확장하고 커스터마이즈할 수 있게 한다.
- IDE와의 밀접한 통합을 추구한다.

Gradle은 안드로이드의 새로운 빌드 시스템의 핵심입니다. 앞의 목표를 달성하 기 위해 안드로이드 애플리케이션의 폴더 구조가 그 전보다 복잡하게 변경되었습 니다. 안드로이드 스튜디오와 이클립스에서 각각 Hello World 프로젝트를 생성 하여 프로젝트 구조를 비교하면 다음과 같습니다.

그림 1-3 안드로이드 스튜디오 프로젝트 구조 vs 이클립스 프로젝트 구조



02 http://tools.android.com/tech-docs/new-build-system/user-guide

첫째, Gradle은 멀티 프로젝트 구조입니다. 안드로이드 스튜디오에서 프로젝트 를 새로 생성하면 단일 프로젝트가 아니라 멀티 프로젝트로 생성됩니다(개발자가 원하면 싱글 프로젝트 구조로 변경할 수 있지만 권장하지는 않습니다). [그림 1-3]의 왼쪽을 보면 'app'이라는 폴더가 있습니다. 이를 Gradle에서는 모듈<sup>Module</sup>이라고 부릅 니다. 이클립스의 프로젝트는 src 폴더를 한 개만 가질 수 있다면, Gradle에서는 app 모듈뿐만 아니라 새로운 모듈을 추가하여 모듈별로 src 폴더를 포함하게 됩 니다.

둘째, src 폴더의 구조가 다릅니다. 이클립스에서는 src 하위에 소스 코드가 패 키지 명과 함께 바로 등장하지만 Gradle에서는 소스 코드가 바로 오지 않고 src 폴더 하위에 androidTest, main, test가 추가됩니다. 원래 androidTest 와 main만 있었지만, Anrdroid Studio 1.4로 오면서 test 폴더가 추가되었 습니다. test 폴더는 구글 I/O 2015에서 소개된 Local Unit Test를 지원합니 다(이 폴더에 관한 자세한 내용은 구글 문서를 참고하기 바랍니다).<sup>03</sup> main 폴더에는 안드 로이드 앱 소스 코드가 있고, androidTest에는 안드로이드 테스팅(정확하게는 Instrumentation Testing) 소스 코드가 있습니다.

셋째, libs 폴더입니다. 이클립스의 libs 폴더에는 빌드하는 데 필요한 외부 라 이브러리 파일이 직접 포함되어 있습니다. 하지만 Gradle에서는 의존성 관리를 Gradle이 담당하므로 libs 폴더를 사용하지 않아도 됩니다. 필요한 스크립트 파 일에서 외부 라이브러리의 저장소와 버전 등을 지정하면 빌드할 때 알아서 해당 버전을 다운로드하여 포함합니다. 또한, + 옵션 등을 적용하면 최신 버전을 자동 으로 다운로드할 수도 있습니다.

넷째, bin 폴더입니다. 기존 이클립스 프로젝트에서는 빌드를 완료한 APK가 bin 폴더에 있지만, 안드로이드 Gradle에서는 build/output/apk 폴더에 위

<sup>03</sup> http://tools.android.com/tech-docs/unit-testing-support

치합니다. 빌드의 결과물은 적용한 플러그인에 따라 APK 파일 혹은 AAR 파일입 니다.

이제 빌드는 단지 APK(또는 AAR)를 생성하는 데서 벗어나 점점 다양한 언어, 프 레임워크와 어우러지고 자동화되어 가고 있습니다.