Chapter 00 모바일 프로그래밍을 시작하면서

0.1 스마트폰

· 통화 기능 + 컴퓨터 + 다양한 기능 내장(미디어플레이어, 카메라, GPS 등)

0.2 스마트폰의 역사

· 1992년 IBM사의 사이먼(최초)
· 1996년 노키아 9000
· 2002년 마이크로소프트 포켓 PC
· 2007년 iPhone
· 2008년 Android Phone
· 2010년 Window Phone 7

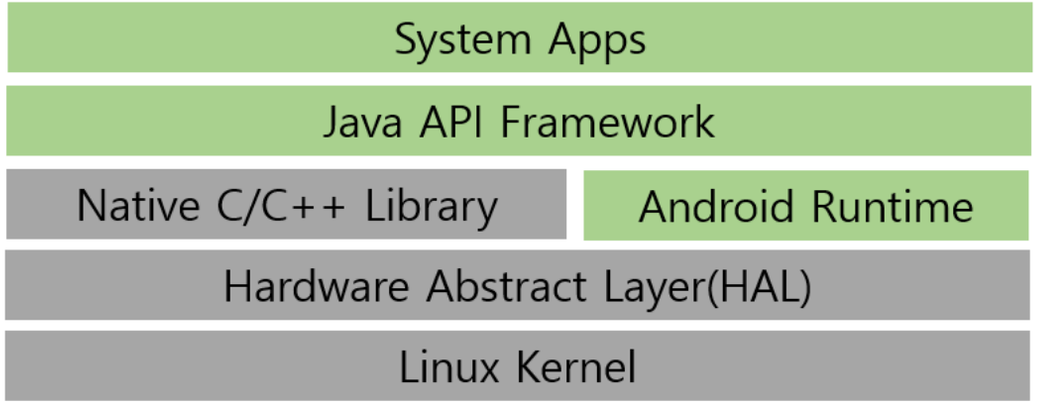
0.3 스마트폰의 운영체제

· 2021년 2분기
기준 세계 시장 점유율:
- 안드로이드 72%;
- 아이폰 23%;
- 윈도폰: 0.02%

구분	안드로이드	아이폰	윈도폰(단종)
개발언어	Java, Kotlin, C++	ObjectC, Swift	C#, VB.Net
개발 운영체제	Windows, Linux, MacOS	MacOS	Windows 8/8.1/10
개발 도구	Eclipse, Android Studio IntelliJ	XCode	VisualStudio 2013이상
지원 장치	안드로이드-폰, 태블릿, 워치, TV	아이폰, 아이팟, 아이패드, 애플워치	윈도폰
대표 제품	삼성 갤럭시 S/폴드/플립 시리즈 구글 픽셀/넥서스 시리즈	아이폰 시리즈, 아이패드 시리즈	노키아 루미아 시리즈
최신 개발 버전	Android 13	iOS 16	윈도폰 10
앱스토어	구글 플레이, 삼성 Apps, One 스토어 등	애플 앱스토어	Windows 스토어

Chapter 01 안드로이드와 코틀린

1.1 운영체제와 플랫폼



출처: https://developer.android.com/guide/platform?hl=ko

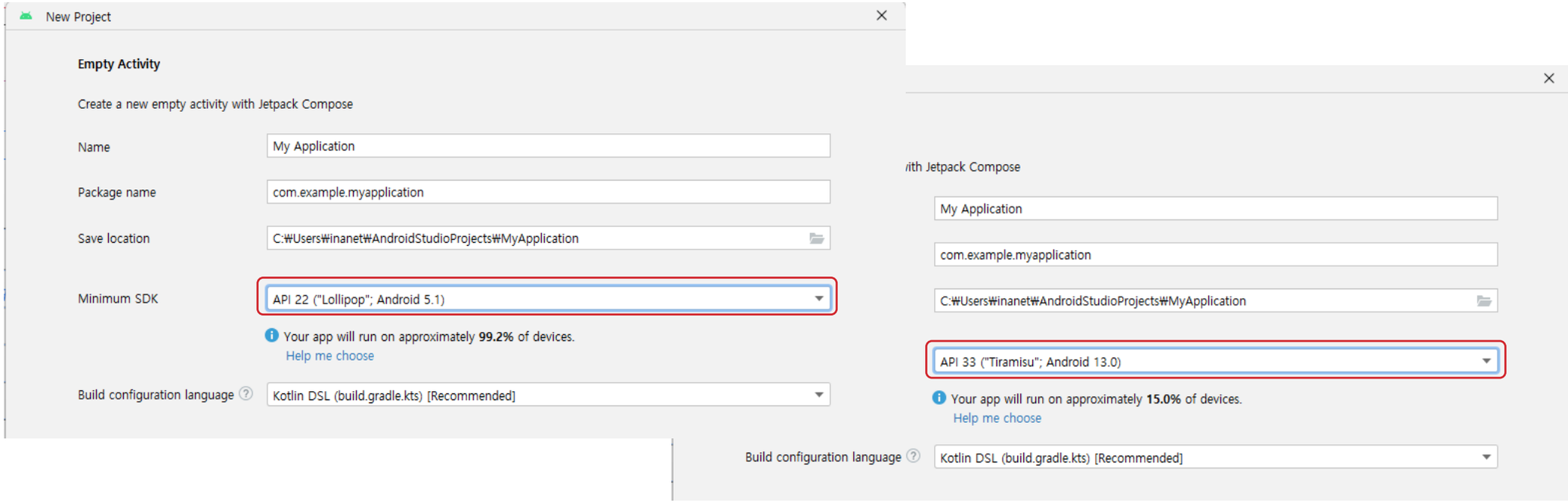
1.2 플랫폼 버전과 API 레벨

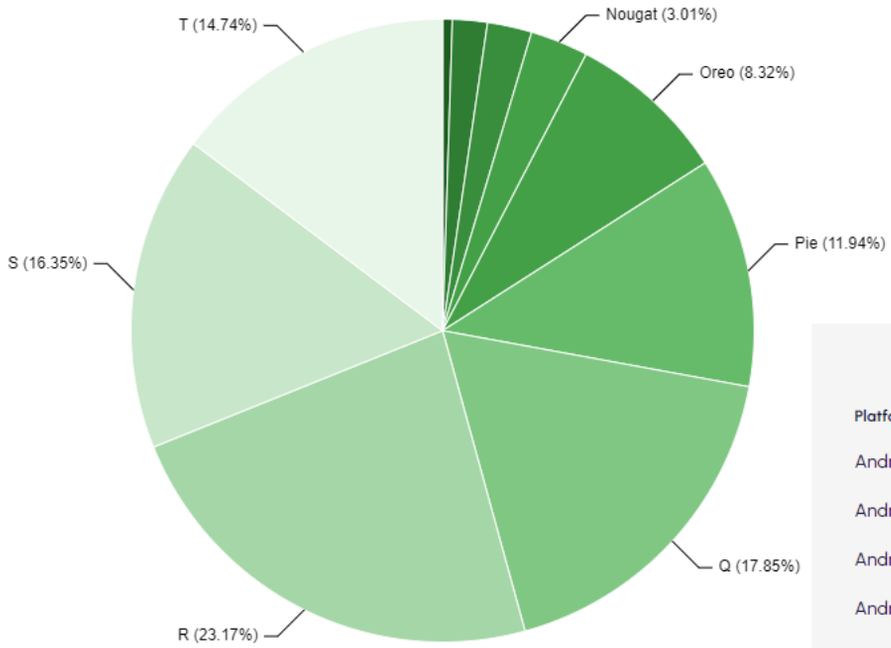
플랫폼 버전	코드네임	API 레벨	Released	버전 코드
13	Tiramisu	33	2022.08	BuildVersionCodes.Tiramisu
12	Snow Cone	31	2021.10	BuildVersionCodes.SnowCone
11	Red Velvet Cake	30	2020.09	BuildVersionCodes.RedVelvetCake
10.0	Queen Cake	29	2019.09	BuildVersionCodes.QueenCake
9.0	Pie	28	2018.08	BuildVersionCodes.Pie
8.1	Oreo	27	2017.12	BuildVersionCodes.OMr1
8.0	Oreo	26	2017.08	BuildVersionCodes.O
7.1	Nougat	25	2016.12	BuildVersionCodes.NMr1
7.0	Nougat	24	2016.08	BuildVersionCodes.Nougat
6.0	Marshmallow	23	2015.08	BuildVersionCodes.Marshmallow
5.1	Lollipop	22	2015.03	BuildVersionCodes.LollipopMr1
5.0	Lollipop	21	2014.11	BuildVersionCodes.Lollipop
4.4W	Kitkat Watch	20	2014.06	BuildVersionCodes.KitKatWatch
4.4	Kitkat	19	2013.10	BuildVersionCodes.KitKat
4.3	Jelly Bean	18	2013.07	BuildVersionCodes.JellyBeanMr2
4.2-4.2.2	Jelly Bean	17	2012.11	BuildVersionCodes.JellyBeanMr1
4.1-4.1.1	Jelly Bean	16	2012.06	BuildVersionCodes.JellyBean
4.0.3-4.0.4	IcecreamSandwich	15	2011.12	BuildVersionCodes.IceCreamSandwichMr1
4.0-4.0.2	IcecreamSandwich	14	2011.10	BuildVersionCodes.IceCreamSandwich
3.2	HoneyComb	13	2011.06	BuildVersionCodes.HoneyCombMr2
3.1.x	HoneyComb	12	2011.02	BuildVersionCodes.HoneyCombMr1
3.0.x	HoneyComb	11	2011.02	BuildVersionCodes.HoneyComb
2.3.3-2.3.4	GingerBread	10	2011.02	BuildVersionCodes.GingerBreadMr1

1.2 플랫폼 버전과 API 레벨

플랫폼 버전	코드네임	API 레벨	Released	버전 코드
2.3-2.3.2	GingerBread	9	2010.11	BuildVersionCodes.GingerBread
2.2.x	Froyo	8	2010.06	BuildVersionCodes.Froyo
2.1.x	Eclair	7	2010.01	BuildVersionCodes.EclairMr1
2.0.1	Eclair	6	2009.12	BuildVersionCodes.Eclair01
2.0	Eclair	5	2009.11	BuildVersionCodes.Eclair
1.6	Donut	4	2009.09	BuildVersionCodes.Donut
1.5	Cupcake	3	2009.05	BuildVersionCodes.Cupcake
1.1	Base	2	2009.02	BuildVersionCodes.Base11
1.0	Base	1	2008.10	BuildVersionCodes.Base

출처: https://learn.microsoft.com/ko-kr/xamarin/android/app-fundamentals/android-api-levels?tabs=windows





API Distribution

Platform Version	API	Distribution
Android 4.4 (KitKat)	19	0.5%
Android 5.0 (Lollipop)	21	0.3%
Android 5.1 (Lollipop)	22	1.5%
Android 6.0 (Marshmallow)	23	2.3%
Android 7.0 (Nougat)	24	1.5%
Android 7.1 (Nougat)	25	1.5%
Android 8.0 (Oreo)	26	2.2%
Android 8.1 (Oreo)	27	6.1%
Android 9.0 (Pie)	28	11.9%
Android 10.0 (Q)	29	17.8%
Android 11.0 (R)	30	23.1%
Android 12.0 (S)	31	16.3%
Android 13.0 (T)	33	14.7%

출처: https://www.composables.com/tools/distribution-chart

참고: Android Studio Versions

코드네임(버전)		Released
1.0		2014.12
~		
4.2		2021.05
Arctic Fox	(20200301)	2021.07
Bumblebee	(20210101)	2022.01
Chipmunk	(20210201)	2022.05
Dolphine	(20210301)	2022.09
Electric Eel	(20220101)	2023.01
Flamingo	(20220101)	2023.04
Giraffe	(20220301)	2023.07
Hedgehog	(20230101)	TBD
Iguana	(20230201)	TBD

** Kotlin 적용

출처: https://en.wikipedia.org/wiki/Android_Studio

* TBA: To Be Announced, 추후 발표
* TBD: To Be Determined(Decided), 추후 결정

1.3 코틀린 Kotlin

- 2017년 5월 Goole I/O에서 코틀린을 공식 언어로 채택
- 2019년 Goole I/O에서 코틀린 First 선언
  모든 문서에서 코틀린 코드를 먼저 노출

- JetBrain 사에서 개발된 프로그래밍 언어
- JetBrain 사의 개발 도구인 IntelliJ를 Android Studio에 적용

버전	Released	주요 내용
1.0	2016.02	정식 버전(안정화 버전) 출시
1.1	2016.02	Kotlin/JS 안정화 버전
1.2	2017.11	Kotlin/Multiplatform 실험 버전 출시 JVM과 자바스크립트 간의 코드 공유 기능 추가
1.3	2018.10	코루틴 지원 추가
1.4.0	2020.08	Object-C / Swift 지원 추가 Kotlin/Multiplatform 알파 버전 출시
1.5.0	2021.05	자바 레코드 클래스 지원 추가 sealed 클래스지원 추가 신규 IR 컴파일러 공개

버전	Released	주요 내용
1.6.0	2021.11	Kotlin/Native 신규 메모리 관리자 추가 Duration API 안정화
1.7.0	2022.06	코틀린 K2 컴파일러 알파 버전 공개 Gradle 증분 빌드 시 새로운 접근 방법 적용
1.8.0	2022.12	Kotlin-reflect 성능 개선 Object-C 및 Swift 상호 운용성 개선
1.9.0	2023.06	1.x의 마지막 릴리즈
2.0.0	-	코틀린 K2 컴파일러 안정화

출처: https://ko.wikipedia.org/wiki/코틀린_(프로그래밍_언어)

1.4 함수형 프로그래밍 언어: 코틀린

· Java vs. Kotlin

Java -

```
class Hello {
    public static void main(String args[]) {
        System.out.print("Hello World!");
    }
}
```

Kotlin - 객체 지향으로 함수를 객체(class) 외부에서 호출

```
System.out.print("Hello World!");
```

1.5 안드로이드 개발에서 자바와 코틀린의 차이

· 자바:

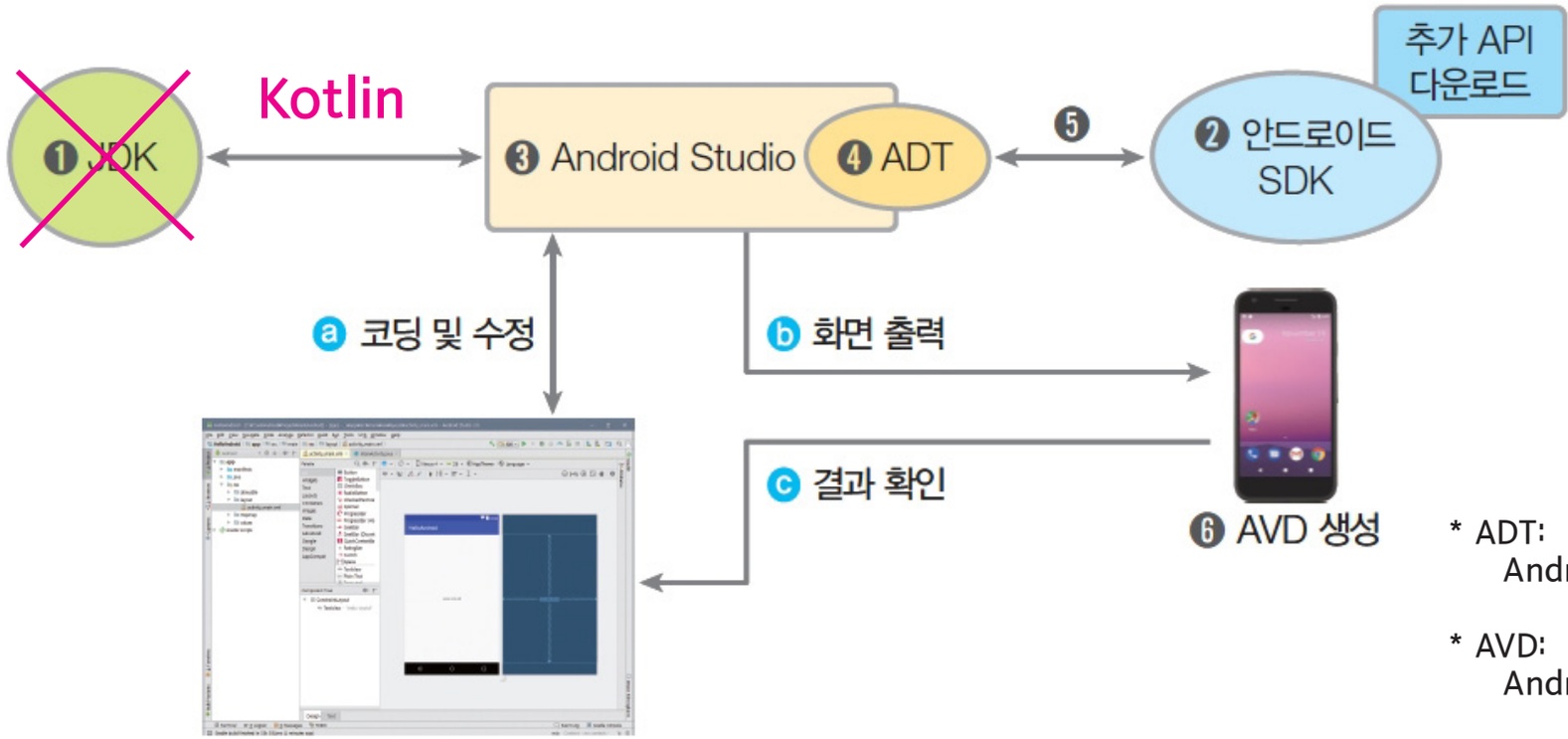
- 객체 지향으로 모든 함수(메서드)는 객체 내부에서 정의되고 객체(class) 내부에서 호출

· 코틀린:

- 객체 지향으로 모든 함수(메서드)는 객체 외부에서 정의될 수도 있고, 객체(class) 내부에서 호출
- 안드로이드에서는 자바와의 호환 문제로 자바 언어의 형식에 맞추어 개발
- 자바 보다는 코드의 양이 줄어든다.

2.1 소스 코드 작성에서 실행까지

- 소스 코드 작성: 코틀린으로 소스 코드 작성
- 설치 파일 생성: 명령을 통해 안드로이드에서 실행 가능한 설치 파일의 형태로 변환
- ~~업로드: 구글 플레이 스토어에 앱을 업로드~~
- ~~앱 등록: 구글 플레이 스토어에 앱을 등록~~
- 앱 선택/설치: 구글 플레이 스토어에 접속하여 앱을 선택/설치
- 앱 실행: 앱 아이콘을 터치하여 앱을 실행



* ADT:
Android Development Tool

* AVD:
Android Virtual Device

** 참고

1. Java Compiler
소스코드(.java)를 바이트코드(.class)로 번역한다.

2. android apk build 과정
.java → .class → .dex → .apk → 서명 및 최적화 → release
- java compiler에 의해 .class 파일로 컴파일
- 생성된 .class 파일을 android DX tool을 통해 .dex로 변환
- .dex과 resource를 합쳐 .apk로 압축

3. JIT Compiler (Just In Time) → Dalvik
실행 시점에 바이트코드(.dex)를 기계어로 번역한다.
인터프리터 방식의 언어들이 성능 향상을 목적으로 도입하는 경우가 많으며,
자주 사용되는 코드는 매 번 번역하지 않고, 캐싱하여 성능을 개선한다.

메소드 단위로 JIT 컴파일하는 방식과
실행 흐름을 실시간으로 추적하며 컴파일하는 Tracing JIT 방식으로 분류된다.
용량↓, 설치속도↑, 실행속도↓, 배터리 사용↑, CPU 사용↑

4. AOT Compiler (Ahead Of Time) → Android RunTime
설치 시점에 기계어로 번역한다.
용량↑, 설치 속도↓, 실행 속도↑, 배터리 사용↓, CPU 사용↓

출처: https://mrnamu.blogspot.com/2019/10/jit-aot_21.html

3.1 새롭게 다루는 것 - 젯팩

- 모든 버전과 기기에서 일관되게 동작하는 젯팩 Jetpack 라이브러리
- 젯팩을 사용하는 이유
  - 하위 버전 호환성: 이전 버전과의 호환성 기능 통합으로 비정상 종료와 메모리 누수를 줄일 수 있게 한다.
  - 반복되는 코드 제거: 백그라운드 작업, 수명 주기 관리 등 반복적이고 지루한 코드를 제거해 준다.
  - 낮은 복잡도: 일관된 동작으로 코드의 복잡도를 낮추어 준다.

- 젯팩 라이브러리
  - 컬렉션(3장)
  - 컨스트레인트 레이아웃(4장)
  - 리사이클러뷰(5장)
  - 뷰페이저2(5장)
  - 프리퍼런스(7장)
  - 룸(8장)

3.2 새롭게 다루는 것 - 뷰 바인딩

- 전통적인 방식의 findViewById()를 대체하여 코틀린 익스텐션 사용
- 2021년 9월 기준 코틀린 익스텐션 제거
- 안드로이드 스튜디오 3.6부터 코틀린 익스텐션을 개선한 뷰바인딩 View Binding 지원
- 뷰 바인딩은 레이아웃 파일이 사용되는 모든 곳에서 코틀린 익스텐션 대체

3.3 코루틴

- 백그라운드 처리를 위해 사용되던 스레드를 경량화한 코루틴 제공