

• 더 멋진 내일 Tomorrow 을 위한 내일 My Career •

내일은



김현석 지음

코틀린

Kotlin

기초 입문편

김앤북

KIM & BOOK



CHAPTER

01

내
일
인
코
틀
린

안드로이드 개발 환경 설정

KIM & BOOK

- 01 안드로이드와 코틀린 소개
- 02 안드로이드 스튜디오와 코틀린 설치
- 03 개발 도구 및 에뮬레이터 설정
- 04 첫 코틀린 프로젝트 생성 및 실행

01. 안드로이드와 코틀린 소개

안드로이드 운영체제의 역사와 발전, 그리고 구글이 개발한 현대적 프로그래밍 언어인 코틀린의 특징과 안드로이드 개발에의 통합에 대해 배웁니다. 이러한 배경 지식은 안드로이드 앱 개발과 코틀린 언어의 중요성을 이해하는 데 필요한 기본적인 이해를 제공합니다.

02. 안드로이드 스튜디오와 코틀린 설치

안드로이드 앱 개발의 핵심 도구인 안드로이드 스튜디오의 설치 및 기본 설정 방법과 코틀린 플러그인의 추가 과정을 학습합니다. 이 과정을 통해 사용자는 안드로이드 앱 개발을 위한 필수 개발 환경을 구축하고, 코틀린 언어를 사용할 준비를 할 수 있습니다.

03. 개발 도구 및 에뮬레이터 설정

Android SDK의 설치 및 관리, Gradle 설정 방법을 배우고, 안드로이드 애플리케이션을 다양한 환경에서 테스트할 수 있는 안드로이드 에뮬레이터 설정 방법을 익힙니다. 이러한 지식은 앱 개발 과정을 효율화하고, 앱을 다양한 디바이스와 환경에서 테스트하는 데 필수적입니다.

04. 첫 코틀린 프로젝트 생성 및 실행

코틀린 언어를 사용하여 안드로이드 스튜디오에서 첫 안드로이드 프로젝트를 어떻게 생성하고 실행하는지 배웁니다. 프로젝트의 기본 구조를 이해하고, 실제로 앱을 빌드하고 에뮬레이터 또는 실제 기기에서 실행하는 방법을 학습함으로써 코틀린을 사용한 안드로이드 앱 개발의 기본적인 절차를 익힙니다.

각 내용은 안드로이드 앱 개발의 기본적인 이해에서부터 실제 코틀린을 사용한 앱 개발까지의 과정을 단계별로 안내하며, 이를 통해 독자들은 안드로이드 개발에 필요한 필수 지식과 실습 경험을 얻을 수 있습니다.

CHAPTER

02

내
일
인
코
틀
린

코틀린 언어 기초

KIM & BOOK

- 01 코틀린 언어 소개
- 02 변수와 데이터 타입
- 03 연산자
- 04 제어문
- 05 함수
- 06 컬렉션

01. 코틀린 언어 소개

코틀린은 JetBrains에서 개발한 언어로, 주로 Android 앱 개발에 사용되며 Java와 100% 호환성을 제공합니다. 정적 타입 언어이면서 표현력이 뛰어나며, 간결하고 실용적인 문법을 지원합니다. 또한 함수형 프로그래밍과 객체 지향 프로그래밍을 모두 지원하여 다양한 프로그래밍 스타일을 채택할 수 있습니다.

02. 변수와 데이터 타입

코틀린에서는 변수를 선언할 때 명시적으로 타입을 지정하거나, 초기값에 따라 자동으로 타입을 추론하는 방식이 가능합니다. 다양한 기본 데이터 타입과 함께 널(null) 안전성을 강조하는 특징이 있습니다.

03. 연산자

코틀린은 기본적인 산술 연산자부터 비교 연산자, 논리 연산자 등을 지원합니다. 또한 연산자 오버로딩을 통해 사용자가 직접 연산자를 정의할 수 있는 기능도 제공합니다.

04. 제어문

조건문인 if-else문부터, when문을 사용한 조건 처리, 반복문인 for문과 while문 등 다양한 제어문이 있습니다. 특히 when문은 다양한 조건을 간결하게 표현할 수 있는 강력한 기능을 제공합니다.

05. 함수

코틀린에서 함수는 일급 객체로 취급되며, 함수형 프로그래밍의 특징을 지원합니다. 함수를 변수에 할당하거나 매개 변수로 전달하는 등의 고차 함수 개념이 도입되어 있습니다. 또한 람다식을 통해 간결하게 함수를 표현할 수 있습니다.

06. 컬렉션

코틀린은 리스트, 세트, 맵 등 다양한 컬렉션을 지원합니다. 불변(immutable) 컬렉션과 가변(mutable) 컬렉션을 제공하며, 함수형 프로그래밍 스타일을 지원하는 컬렉션 연산이 풍부합니다.

챕터 2에서는 코틀린 언어의 기본 개념을 탐험하였습니다. 변수와 데이터 타입, 연산자, 제어문, 함수, 그리고 여러 가지 컬렉션에 대해 학습하여, 다양한 프로그래밍 상황에서 코틀린을 유연하게 활용할 수 있게 되었습니다.

CHAPTER

03

내
일
인
코
틀
린

객체 지향 프로그래밍

KIM & BOOK

- 01 클래스와 객체
- 02 생성자
- 03 상속
- 04 오버라이딩
- 05 캡슐화
- 06 다형성
- 07 추상 클래스와 인터페이스

01. 클래스와 객체

클래스는 객체를 생성하기 위한 설계도이며, 객체는 클래스의 인스턴스입니다. 클래스는 속성(멤버 변수)과 동작(메서드)을 가지며, 객체는 이러한 속성과 동작을 실제로 가지게 됩니다.

02. 생성자

생성자는 클래스의 인스턴스를 초기화하기 위해 사용됩니다. 객체를 생성할 때 호출되며, 초기값을 설정하거나 필요한 작업을 수행하는 역할을 합니다.

03. 상속

상속은 기존 클래스를 확장하여 새로운 클래스를 생성하는 개념입니다. 상속을 통해 기존 클래스의 속성과 동작을 재사용하고 확장할 수 있습니다.

04. 오버라이딩

오버라이딩은 상위 클래스의 메서드를 하위 클래스에서 재정의하는 것을 말합니다. 하위 클래스는 상위 클래스의 메서드를 동일한 이름으로 재정의하고 자신에게 맞게 구현할 수 있습니다.

05. 캡슐화

캡슐화는 데이터와 그 데이터를 다루는 메서드를 하나로 묶어 정보를 은닉하고 외부에서의 접근을 제어하는 것입니다. 객체의 내부 구현을 외부에 감추고 필요한 기능만을 제공합니다.

06. 다형성

다형성은 동일한 이름의 메서드가 다양한 형태로 동작하는 것을 말합니다. 상위 클래스 타입으로 하위 클래스의 객체를 참조할 수 있고, 실행 시에 실제 객체의 타입에 따라 적절한 메서드가 호출됩니다.

07. 추상 클래스와 인터페이스

추상 클래스는 일부 메서드를 구현하지 않고 선언만 하는 클래스입니다. 하위 클래스에서 이를 구체화하여 사용합니다. 인터페이스는 메서드의 선언만을 갖는 틀로, 다른 클래스에서 해당 인터페이스를 구현하여 필요한 메서드를 정의합니다.

챕터3에서는 객체 지향 프로그래밍의 기본 개념과 원리를 이해했습니다. 객체 지향 프로그래밍의 개념과 기능을 이해하고 활용하면, 유지·보수가 쉽고 확장 가능한 코드를 작성할 수 있으며, 코드의 재사용성과 가독성을 높일 수 있습니다. 객체 지향 프로그래밍은 복잡한 문제를 해결하기 위한 효율적인 방법론으로 널리 사용되고 있습니다.

CHAPTER

04

내
일
인
코
틀
린

안드로이드의 기본 개념

KIM & BOOK

01 안드로이드 애플리케이션의 구조

02 액티비티와 프래그먼트

03 인텐트와 네비게이션

04 뷰와 뷰 그룹

01. 안드로이드 애플리케이션의 구조

안드로이드 애플리케이션은 기본적으로 앱의 구조를 나타냅니다. 이는 앱이 여러 요소로 구성되는 방식을 설명합니다. 주요 구성 요소로는 액티비티, 서비스, 브로드캐스트 리시버, 콘텐츠 프로바이더 등이 있습니다.

02. 액티비티와 프래그먼트

- 액티비티: 사용자 인터페이스 화면을 표시하는 컴포넌트입니다. 하나의 앱에는 여러 개의 액티비티가 있을 수 있으며, 각각의 액티비티는 사용자와의 상호 작용을 담당합니다.
- 프래그먼트: 사용자 인터페이스의 일부를 나타내는 독립적인 모듈입니다. 액티비티 내에서 여러 프래그먼트를 조합하여 복잡한 UI를 구성할 수 있으며, 화면의 일부를 동적으로 변경하고 재사용할 수 있습니다.

03. 인텐트와 네비게이션

- 인텐트: 컴포넌트 간에 작업을 수행하는 데 사용되는 객체입니다. 명시적 인텐트를 사용하여 액티비티 간에 데이터를 전달하거나, 암시적 인텐트를 사용하여 다른 애플리케이션의 컴포넌트를 시작할 수 있습니다.
- 네비게이션: 사용자가 앱 내에서 이동하는 방법을 관리하는 것을 의미합니다. 액티비티 간의 이동은 명시적 인텐트를 사용하거나 액티비티 간의 관계를 정의하는 네비게이션 그래프를 활용하여 수행됩니다.

04. 뷰와 뷰 그룹

- 뷰(View): 사용자 인터페이스의 기본 요소로 버튼, 텍스트뷰, 이미지뷰 등이 포함됩니다. 이들은 화면에 표시되고 사용자와 상호 작용합니다.
- 뷰 그룹(ViewGroup): 뷰의 컨테이너 역할을 합니다. 뷰 그룹은 다른 뷰나 뷰 그룹을 포함할 수 있으며, 화면의 구조를 계층적으로 구성합니다. 대표적인 뷰 그룹으로는 LinearLayout, RelativeLayout, FrameLayout 등이 있습니다.

이번 챕터에서는 안드로이드 애플리케이션의 구조와 주요 구성 요소들을 학습했습니다. 애플리케이션의 전반적인 구조를 이해하고, 액티비티와 프래그먼트를 통해 화면을 구성하는 방법을 배웠습니다. 그 다음으로는 인텐트와 네비게이션을 활용하여 애플리케이션의 다양한 화면 간 전환과 네비게이션을 관리하는 방법을 익혔습니다. 마지막으로 뷰와 뷰 그룹을 통해 사용자 인터페이스를 설계하고 구현하는 방법을 학습했습니다. 이를 통해 안드로이드 애플리케이션 개발의 기본 원리와 구성 요소들을 이해하고 활용할 수 있게 되었습니다.

CHAPTER

05

내
일
인
코
틀
린

안드로이드 UI 디자인

KIM & BOOK

- 01 레이아웃과 위젯
- 02 텍스트와 이미지 처리
- 03 사용자 입력과 폼 구성
- 04 리스트뷰와 리사이클러뷰

01. 레이아웃과 위젯

레이아웃은 화면에 UI 요소들을 배치하는 데 사용됩니다. ConstraintLayout, LinearLayout, RelativeLayout 등 다양한 레이아웃을 사용하여 UI를 구성할 수 있습니다. 위젯은 사용자 인터페이스를 구성하는 요소로, 버튼, 텍스트뷰, 이미지뷰 등이 있습니다. 위젯은 레이아웃 내에 배치되어 사용자와 상호 작용합니다.

02. 텍스트와 이미지 처리

안드로이드에서 텍스트는 TextView를 통해 표시되며, 다양한 속성을 사용하여 스타일링할 수 있습니다. 이미지 처리는 ImageView를 사용하여 이미지를 표시하고, 이미지 리소스를 로드하거나 비트맵으로 변환하여 표시할 수 있습니다. 또한 텍스트와 이미지를 함께 표시하는 것도 가능하며, 텍스트와 이미지를 함께 사용하는 다양한 디자인 패턴을 익힐 수 있습니다.

03. 사용자 입력과 폼 구성

사용자 입력은 EditText, Button, CheckBox 등을 사용하여 구현됩니다. 사용자 입력을 받고 처리하는 방법을 배우고, 입력값의 유효성을 검사하는 방법을 익힐 수 있습니다. 폼은 여러 입력 요소를 그룹화하여 사용자에게 정보를 입력하도록 하는 데 사용됩니다. 폼을 효과적으로 구성하고 처리하는 방법을 배울 수 있습니다.

04. 리스트와 리사이클러뷰

리스트는 여러 항목을 세로로 나열하여 표시하는 데 사용됩니다. ListView, RecyclerView 등을 사용하여 리스트를 구현할 수 있으며, 각 항목의 클릭 이벤트를 처리하는 방법을 익힐 수 있습니다. 리사이클러뷰는 리스트를 표시하기 위한 고급 위젯으로, 성능과 유연성이 뛰어납니다. 리사이클러뷰를 사용하여 동적인 리스트를 구현하고, 각 항목의 뷰를 재활용하여 메모리 사용량을 최적화하는 방법을 배울 수 있습니다.

안드로이드 UI 디자인에서 배운 내용은 UI를 다양한 방식으로 구성하는 방법을 포함합니다. 이를 통해 사용자와의 상호 작용을 향상시키고, 앱의 사용자 경험을 개선할 수 있습니다. 텍스트와 이미지를 다루는 방법을 익히며, 사용자 입력을 처리하고 폼을 만드는 방법을 배웠습니다. 리스트와 리사이클러뷰를 활용하여 동적인 내용을 표시하고, 스타일과 테마를 사용하여 일관된 디자인을 유지하는 방법도 배웠습니다. 이러한 기술을 통해 안드로이드 앱의 UI를 보다 효과적으로 디자인하고 관리할 수 있습니다.

● 더 멋진 내일 Tomorrow 을 위한 내일 My Career ●

내일은



김현석 지음

코틀린 Kotlin

응용 실전편

김앤북

KIM & BOOK



CHAPTER

07

내
일
이
코
틀
린

안드로이드 앱 개발 심화

- 01 예외 처리
- 02 라이브러리 및 패키지
- 03 데이터베이스 연동
- 04 API 연동
- 05 알림
- 06 로컬 저장소와 파일 시스템

01. 예외 처리

애플리케이션뿐만 아니라 개발에서 예외 처리는 중요한 부분으로, 예외 상황을 처리하는 메커니즘을 학습하였습니다. 예외 처리는 앱이 예상치 못한 상황에 대비하고 오류를 처리하고 복구하는 방법을 제공합니다. 예외 처리를 통해 앱의 안정성을 향상시키고 사용자에게 더 나은 사용 경험을 제공할 수 있습니다.

02. 라이브러리 및 패키지

안드로이드 애플리케이션을 개발할 때 외부 라이브러리와 패키지를 활용할 수 있습니다. 여기에서는 다양한 라이브러리와 패키지를 가져오고 사용하는 방법을 배웠습니다. 외부 패키지를 활용하면 앱의 기능을 향상시킬 수 있으며, 개발 시간을 단축시킬 수도 있습니다.

03. 데이터베이스 연동

대부분의 애플리케이션은 데이터를 저장하고 관리해야 합니다. 코틀린에서는 다양한 데이터베이스와 연동할 수 있습니다. 데이터베이스에 데이터를 저장, 검색, 수정하는 방법을 학습하였습니다. 데이터베이스 연동을 통해 앱의 데이터 관리 기능을 개발할 수 있습니다.

04. API 연동

코틀린 애플리케이션은 외부 API와 통신하여 데이터를 가져올 수 있습니다. RESTful API와 통신하여 데이터를 처리하는 방법을 배웠습니다. API 요청과 응답을 처리하는 방법을 학습하고, 외부 서비스의 데이터를 앱에서 활용할 수 있습니다.

05. 알림

알림은 사용자에게 중요한 정보를 전달하는 데 도움을 주는 기능입니다. 여기에서는 안드로이드 앱에서 알림을 표시하고 푸시 알림을 구현하는 방법을 배웠습니다. 알림 기능을 추가하면 앱 사용자에게 더 좋은 상호 작용과 개인화된 경험을 제공할 수 있습니다.

06. 로컬 저장소와 파일 시스템

안드로이드 앱은 로컬 저장소와 파일 시스템을 활용하여 데이터를 저장하고 관리할 수 있습니다. 여기에서는 로컬 저장소와 파일 시스템을 다루며, 파일을 읽고 쓰는 방법을 학습하였습니다. 로컬 저장소와 파일 시스템을 사용하면 오프라인 환경에서도 앱의 데이터를 유지하고 관리할 수 있습니다.

챕터7 안드로이드 앱 개발 심화를 통해 코틀린 개발에 필요한 예외 처리, 외부 라이브러리 및 패키지 사용, 데이터베이스 및 API 연동, 알림 기능, 로컬 저장소와 파일 시스템 활용에 대한 기본 개념과 구현 방법을 학습하였습니다. 이를 통해 다양한 기능을 갖춘 효과적인 안드로이드 앱을 개발할 수 있습니다.

CHAPTER

08

내
일
인
코
틀
린

안드로이드의 고급 주제



- 01 코루틴과 비동기 처리
- 02 테스트와 디버깅
- 03 커스텀뷰와 애니메이션
- 04 클린 아키텍처와 MVVM 패턴

01. 코루틴과 비동기 처리

코루틴은 비동기 코드를 작성하고 관리하기 위한 강력한 도구로, 기존의 콜백이나 RxJava와 같은 라이브러리에 비해 간단하고 가독성이 높은 코드를 작성할 수 있습니다. 코루틴을 사용하면 UI 스레드를 차단하지 않고 비동기 작업을 처리할 수 있으며, 복잡한 비동기 작업을 간편하게 구성할 수 있습니다.

02. 테스트와 디버깅

안드로이드 스튜디오에서 제공하는 다양한 도구를 사용하여 유닛 테스트, 통합 테스트, UI 테스트를 수행하는 방법을 배우고, 앱의 버그를 식별하고 수정하는 방법을 익힐 수 있습니다. 테스트와 디버깅을 통해 안정적이고 품질 높은 앱을 개발할 수 있습니다.

03. 커스텀뷰와 애니메이션

안드로이드에서 기본 제공하는 뷰 이외에도 사용자 정의 뷰를 만들고 활용하는 방법을 학습하며, 다양한 애니메이션 효과를 적용하는 방법을 배웠습니다. 커스텀뷰와 애니메이션을 활용하여 앱의 사용자 경험을 향상시키고 독특한 디자인을 구현할 수 있습니다.

04. 클린 아키텍처와 MVVM 패턴

클린 아키텍처와 MVVM(Mode-View-ViewModel) 패턴을 소개하고, 이를 적용하여 안드로이드 앱의 코드를 구조화하는 방법을 학습하였습니다. 클린 아키텍처와 MVVM 패턴을 사용하면 코드의 유지 보수성과 확장성을 향상시키고, 개발 과정을 단순화할 수 있습니다.

여기에서는 안드로이드 앱 개발의 고급 주제를 다루며, 코루틴과 비동기 처리, 테스트와 디버깅, 커스텀뷰와 애니메이션, 그리고 클린 아키텍처와 MVVM 패턴을 학습하였습니다. 비동기 작업 처리와 UI 개선을 위한 코루틴 활용부터 시작하여 테스트와 디버깅으로 안정성을 높이고, 사용자 정의 뷰와 애니메이션을 통해 UI를 개선하며, 클린 아키텍처와 MVVM 패턴을 통해 코드 구조를 최적화할 수 있습니다.

CHAPTER

09

내
일
이
코
틀
린

안드로이드 앱 최적화 및 배포

KIM & BOOK

01 성능 최적화

02 보안 강화

03 Play 스토어에 앱 배포하기

01. 성능 최적화

안드로이드 앱의 성능 최적화는 사용자 경험을 향상시키고 앱의 효율성을 높이는 핵심적인 과제입니다. 성능 최적화를 위해 UI/UX를 개선하는 방법과 함께 메모리 관리 기법을 적용하는 방법을 배웠습니다. 또한 네트워크 통신 최적화를 통해 데이터 전송 시간을 단축하는 방법과 앱의 반응 속도를 향상시키는 방법을 살펴보았습니다.

02. 보안 강화

안드로이드 앱의 보안 강화는 사용자의 개인정보 보호와 데이터의 안전을 보장하기 위해 중요합니다. 보안을 강화하기 위해 취약점 분석 및 보안을 수행하는 방법과 함께 보안 업데이트를 주기적으로 수행하는 것에 대한 중요성을 배웠습니다. 또한 SSL/TLS 프로토콜을 통한 안전한 통신 설정과 중간자 공격 및 데이터 변조 방지 방법에 대해 살펴보았습니다.

03. Play 스토어에 앱 배포하기

Play 스토어는 안드로이드 앱을 널리 보급하고 사용자에게 접근성을 제공하는 중요한 플랫폼입니다. 앱의 등록부터 게시까지의 과정을 다루며, 광고 플랫폼을 선택하고 통합하여 앱의 가시성을 높이는 방법에 대해 배웠습니다.

챕터9 안드로이드 앱 최적화 및 배포 챕터에서는 앱을 개발하고 사용자에게 제공하기 전에 성능을 최적화하고 보안을 강화하는 방법부터 Play 스토어에 앱을 배포하고 광고 플랫폼을 이용하여 수익 창출하는 방법까지 알아보았습니다.