11. 안드로이드 애플리케이션과 액티비디 생명주기

- 안드로이드 5.0 롤리팝부터는 안드로이드 런타임 시스템의 통제하에 리눅스 프로세스로 실행된다. 그리고 각 애플리케이션은 액티비티나 서비스의 형태로된 컴포넌트로 구성된다.

모바일 장치에서는 리소스가 제한된다. 안드로이드 런타임 시스템은 메모리가 부족할 때 프로세스를 중단시킬 수 있다. 런타임 시스템은 프로세스 상태를 감안하여 프로세스를 중단시킨다. 프로세스 상태는 그 프로세스가 포함하여 실행하는 액티비티의 상태에 따라 좌우된다.

12. 액티비티 상태 변화 처리하기

- 액티비티들은 안드로이드 ACTIVITY클래스를 상속한다. 액티비티의 상태가 변경될 때 런타임 시스템에 의해 호출되도록 이벤트 메소드를 가지고 있다.

이 메소드들을 오버라이딩하여 액티비티는 상태 변화에 응답할 수 있다. 액티비티와 애플리케이션 모두의 현재 상태 데이터를 저장 및 복원할 수 있다.

13. 액티비 상태 변화 예제

- 애플리케이션을 생성하여 액티비티 상태 변화를 실제로 볼 수 있는 코드를 만들고 안드로이드 런타임 안에서 액티비티가 서로 다른 상태를 거치는 동안 생명주기 메소드들이 어떻게 호출되는지 확인한다.

14. 액티비티 상태를 저장하고 복원하기

- 동적 상태 데이터를 저장하고 복원할 때는 적합한 생명주기 메소드에 필요한 코드를 구현한다. 상태 데이터의 저장은 onSaveinstanceState() 메소드에 저장한다. 저장한 값을 추출하고 설정한 후 런터임 시스템에서 액티비티로 전달한 Bundlel객체에 키, 값 상으로 데이터를 저장한다.

동적 사태 데이터는 해당 액티비티의 onCreate() 또는 onRestoreInstanceState() 메소드 중 하나에서 복원할 수 있다. 이때는 Bundle객체로부터 데이터를 추출한 후 해당 액티비티의 사용자 인터페이스 뷰 객체에 설정한다.

15. 안드로이드 뷰, 뷰 그룹, 레이아웃

- 사용자 인터페이스 화면에 있는 각 요소는 뷰이며, 모든 뷰는 android.view.View 클래스의 서브 클래스이다. 뷰는 장치 화면의 직사각형 영역을 나타내며, 그 안에 나타나는 것에 대한 책임이 있고 뷰의 경계 안에서 발생하는 이벤트를 처리하는 책임도 있다. 여러 뷰를 결합하여 하나의 복합 뷰를 만들어 있다. 복합 뷰 안에 있는 뷰들은 컨테이너 뷰의 자식들이다. 컨테이너 뷰는 android.view.ViewGroup 클래스의 서브 클래스이다. 사용자 인터페이스는 뷰 계층 구조의 형태로 생성된 뷰들로 구성된다.

16. 안드로이드 스튜디오 디자이너 도구 살펴보기

- 사용자 인터페이스 생성은 안드로이드 개발에 핵심이다. 안드로이드 스튜디오는 2가지 모드로 동작하는 디자이너 도구를 사용해서 사용자 인터페이스를 생성한다. 디자인 모드에서는 팔레트에서 뷰 컴포넌트를 선택하여 안드로이드 장치 화면을 나타내는 레이아웃에 위치시킨다. 그리고 속성 창에서는 그 뷰의 속성 목을 사용해서 구성한다. 텍스트 모드에서는 사용자 인터페이스 레이아웃을 나타내는 XML을 직접 편집할 수 있고 수정된 내용은 미리보기 화면에 즉시 반영된다.

17. 안드로이드 스튜디오 디자이너 도구로 UI디자인 하기

- 디자이너 도구는 사용자 인터페이스를 디자인할 때 시각적으로 알기 쉬운 방법을 제공한다. 원하는 뷰 컴포넌트를 마우스로 끌어다 넣은 후 속성을 편집하면 디자인이 완성된다. 잘 구성되고 쉽게 이해할 수 있는 형태로 된 XML 레이아웃 리소스 파일을 직접 작성하여 사용자 인터페이스 디자인을 구현 할 수 있다.

계층 구조 뷰어를 사용하여 액티비티의 뷰 트리를 자세히 볼 수 있고 액티비티에 포함된 각 뷰의 유형별 성능을 알 수 있다.

**18장. 자바 코드로 안드로이드 UI 생성하기**

* XML 리소스 파일 기반에서는 UI변경 시 XML 파일만 변경하면 되므로 애플리케이션 코드 재컴파일이 불필요함.
* 실시간으로 동적인 변경이 필요할 경우 자바코드를 사용하는 것이 좋음.
* 뷰를 사용할 때에는 뷰의 ID를 정의해야 함.
* 각 뷰의 속성은 각 getter, setter 메서드를 통해 획득 가능함.

**19장. 안드로이드 스튜디오 디자이너에서 GridLayout 매니저 사용하기**

* GridLayout은 자식 뷰들이 격자 형태의 레이아웃에 위치할 수 있도록 함.
* 셀 내부에서 자식 뷰를 위치할 때 해당 뷰의 gravity 속성을 사용해 정의함.
* 부모 뷰의 공간을 전부 사용하느 match\_parent가 높이/너비 매개변수의 기본 값.
* 자식 뷰가 자신의 콘텐츠를 보여줄 수 있는 크기의 공간만 사용하는 wrap\_content를 일반적으로 사용.

## 20장. XML 레이아웃 리소스로 GridLayout 사용하기

## GridLayout에서 사용할 수 있는 속성들

* android:rowCount : 그리드를 구성하는 행(row)의 개수
* android:columnCount : 그리드를 구성하는 열(column)의 개수
* android:orientation : 방향 정의 (예: horizontal, vertical)

## 자식 뷰에서 사용 할 수 있는 속성들

* android:layout\_row : 자식 뷰가 위치할 부모 그리드의 행 번호
* android:layout\_column : 자식 뷰가 위치할 부모 그리드의 열 번호
* android:layout\_gravity : 부모 컨테이너의 여유 공간에 뷰가 모두 채워지지 않아 여유 공간 안에서 뷰를 정렬할 때(가로|세로 방향 값)
* android:layout\_rowSpan : 행 병합할 개수
* android:layout\_columnSpan : 열 병합할 개수
* android:layout\_margin : 자식 뷰의 둘레의 마진을 설정