1. 어댑터(Adapter)

어댑터(Adapter): 외부 데이터 소스와 어댑터뷰(AdapterView) 사이의 연결 수단.

여러 계층으로 구성: 리스트어댑터, 스피너어댑터, 베이스어댑터 등.

데이터 소스: DB, XML, 배열 등.

어댑터뷰: ViewGroup의 서브클래스 (GridView, ListView, Gallery, Spinner 등).

어댑터를 이용하여 데이터로 레이아웃 구성 및 사용자 선택 이벤트 처리.

2. ArrayAdapter

ArrayAdapter: 자바 배열이나 java.util.List의 인스턴스를 사용하여 어댑터를 제공.

데이터 소스가 배열인 경우 사용.

어댑터 설정:

String[] items = {"item 1", "item 2", "item 3"};

ArrayAdapter<String> adapter = new ArrayAdapter<String>(

this,

android.R.layout.simple\_list\_item\_1,

items

);

3. 레이아웃 리소스 ID

다양한 레이아웃 리소스 ID:

simple\_list\_item\_1: 하나의 텍스트 뷰.

simple\_list\_item\_checked: 체크 표시 레이아웃.

simple\_list\_item\_single\_choice: 라디오 버튼 표시.

simple\_list\_item\_multiple\_choice: 체크 버튼 표시.

4. ListView

목적: 어댑터 사용 방법 습득 및 어댑터뷰(리스트뷰, 그리드뷰, 스피너 등) 사용법 습득.

ListView: 안드로이드에서 리스트박스를 구현하기 위한 클래스.

진행 순서:

레이아웃에 ListView 추가.

ListView 객체 얻기.

setAdapter() 메소드 호출로 데이터 지정.

리스너 연결 (setOnItemClickListener()).

5. Spinner

목적: AdapterView 가운데 스피너(Spinner) 사용법 습득.

Spinner: 드롭다운 선택 기능을 구현하는 위젯.

실행 방법: XML에서 android:entries 속성으로 배열 리소스를 지정하여 사용.

6. 리소스 준비 및 코드

배열 자료 준비: arrays.xml 파일 추가.

리스너 연결:

선택한 리스트 항목을 알아내기 위한 리스너 연결 (OnItemSelectedListener).

항목 선택 시 호출되는 메소드와 연결 해제 시 호출되는 메소드.

GridView

목적: 2차원 격자 형태로 항목을 배치하고, 사용자가 항목을 선택할 수 있도록 함.

속성:

android:verticalSpacing: 항목 간의 상하 여백을 결정.

android:horizontalSpacing: 항목 간의 좌우 여백을 결정.

android:columnWidth: 각 컬럼의 너비를 지정.

android:numColumns: 한 줄에 표시되는 컬럼 수. auto\_fit으로 설정하면 자동 조절.

메소드:

setAdapter(): GridView 객체와 커스텀 어댑터를 연결. 커스텀 어댑터인 ImageAdapter와 연결.

getCount(): Adapter에 포함된 항목의 개수를 반환.

getItem(): 지정한 위치의 항목을 반환.

getItemId(): 지정한 위치의 ID를 반환. 인덱스 자체가 고유 ID이므로 인덱스를 그대로 반환.

getView(): 항목 하나를 출력하기 위한 뷰를 생성하여 반환. 커스텀 어댑터 정의 시 재정의하여 커스텀 레이아웃을 생성.

Android:theme=”@style/NoActionTheme”

BaseAdapter

목적: GridView에 표시할 데이터의 커스텀 어댑터를 정의.

핵심 메소드:

getCount(), getItem(), getItemId(), getView()를 오버라이드하여 어댑터의 동작을 정의.

ImageView

목적: GridView와 BaseAdapter 사용법 연습 및 이미지 출력 방법 습득.

기능: 아이콘 등 임의의 이미지를 출력. TextView 클래스를 확장한 형태로, 이미지를 표시할 수 있는 TextView라고 생각할 수 있음.

속성:

android:src, app:srcCompat: 출력할 이미지의 이름을 입력. drawable 폴더에 있는 이미지를 지정.

android:tint: 이미지의 배경 색상 설정.

**질문 1: 어댑터(Adapter)란 무엇인가요?**

* **정의와 역할**: 어댑터는 외부 데이터 소스와 어댑터뷰(AdapterView) 사이의 연결 역할을 합니다. 어댑터는 데이터 소스를 어댑터뷰에 표시할 수 있는 형식으로 변환하여, 데이터의 표현 및 사용자 상호작용을 가능하게 합니다.
* **종류와 데이터 소스**: 여러 종류의 어댑터가 있으며, 대표적으로 리스트어댑터(ListAdapter), 스피너어댑터(SpinnerAdapter), 베이스어댑터(BaseAdapter) 등이 있습니다. 데이터 소스에는 데이터베이스(DB), XML 파일, 배열(Array) 등이 포함됩니다.

**질문 2: ArrayAdapter를 어떻게 사용하나요?**

* **정의와 용도**: ArrayAdapter는 자바 배열이나 java.util.List의 인스턴스를 사용하여 어댑터를 제공합니다. 주로 데이터 소스가 배열인 경우에 사용됩니다.
* **설정 방법 예시 코드**:

String[] items = {"item 1", "item 2", "item 3"};

ArrayAdapter<String> adapter = new ArrayAdapter<String>(

this,

android.R.layout.simple\_list\_item\_1,

items

);

위의 코드에서 items 배열의 데이터를 simple\_list\_item\_1 레이아웃을 사용해 표시하는 ArrayAdapter를 설정하고 있습니다.

**질문 3: ListView의 목적과 설정 방법에 대해 설명하세요.**

* **목적**: ListView는 어댑터를 사용하여 목록 형태로 데이터를 표시하는데 사용됩니다. 사용자 상호작용을 통해 항목을 선택할 수 있도록 지원합니다.
* **설정 방법**:
  1. 레이아웃 XML 파일에 ListView를 추가합니다.
  2. ListView 객체를 Java 코드에서 얻습니다.
  3. setAdapter() 메소드를 호출하여 데이터를 지정합니다.
  4. setOnItemClickListener()를 사용하여 사용자 클릭 이벤트를 처리합니다.

**질문 4: Spinner의 기능은 무엇인가요?**

* **정의와 목적**: Spinner는 드롭다운 선택 기능을 구현하는 위젯입니다. 사용자가 여러 옵션 중에서 하나를 선택할 수 있게 해줍니다.
* **XML 속성**: Spinner를 설정할 때 XML에서 android:entries 속성을 사용하여 배열 리소스를 지정할 수 있습니다. 예를 들어:

<Spinner

android:id="@+id/spinner"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:entries="@array/my\_array" />

**질문 5: GridView의 주요 속성과 메소드를 설명하세요.**

* **역할**: GridView는 항목을 2차원 격자 형태로 배치하여 사용자가 항목을 선택할 수 있도록 합니다.
* **주요 속성**:
  + android:verticalSpacing: 항목 간의 상하 여백을 설정합니다.
  + android:horizontalSpacing: 항목 간의 좌우 여백을 설정합니다.
  + android:numColumns: 한 줄에 표시되는 컬럼 수를 설정합니다. auto\_fit으로 설정하면 자동으로 조정됩니다.
* **주요 메소드**:
  + setAdapter(): GridView와 어댑터를 연결합니다.
  + getCount(): 어댑터에 포함된 항목의 개수를 반환합니다.
  + getItem(): 지정한 위치의 항목을 반환합니다.
  + getView(): 항목 하나를 출력하기 위한 뷰를 생성하여 반환합니다.

**질문 6: BaseAdapter란 무엇인가요?**

* **정의와 목적**: BaseAdapter는 GridView와 같은 어댑터뷰에 사용할 커스텀 어댑터를 정의할 때 사용하는 클래스입니다. 개발자가 원하는 방식으로 데이터를 표시할 수 있도록 지원합니다.
* **오버라이드해야 하는 핵심 메소드**: getCount(), getItem(), getItemId(), getView() 메소드를 오버라이드하여 어댑터의 동작을 정의해야 합니다.

**질문 7: ImageView의 역할은 무엇인가요?**

* **정의와 기능**: ImageView는 이미지를 표시하는 뷰입니다. 아이콘이나 사진과 같은 이미지를 출력할 때 사용됩니다. TextView 클래스를 확장한 형태로 생각할 수 있으며, 이미지를 출력할 수 있는 TextView라고 볼 수 있습니다.
* **주요 속성**:
  + android:src: 출력할 이미지의 이름을 입력하여 drawable 폴더에 있는 이미지를 지정합니다.
  + android:tint: 이미지의 배경 색상을 설정합니다.