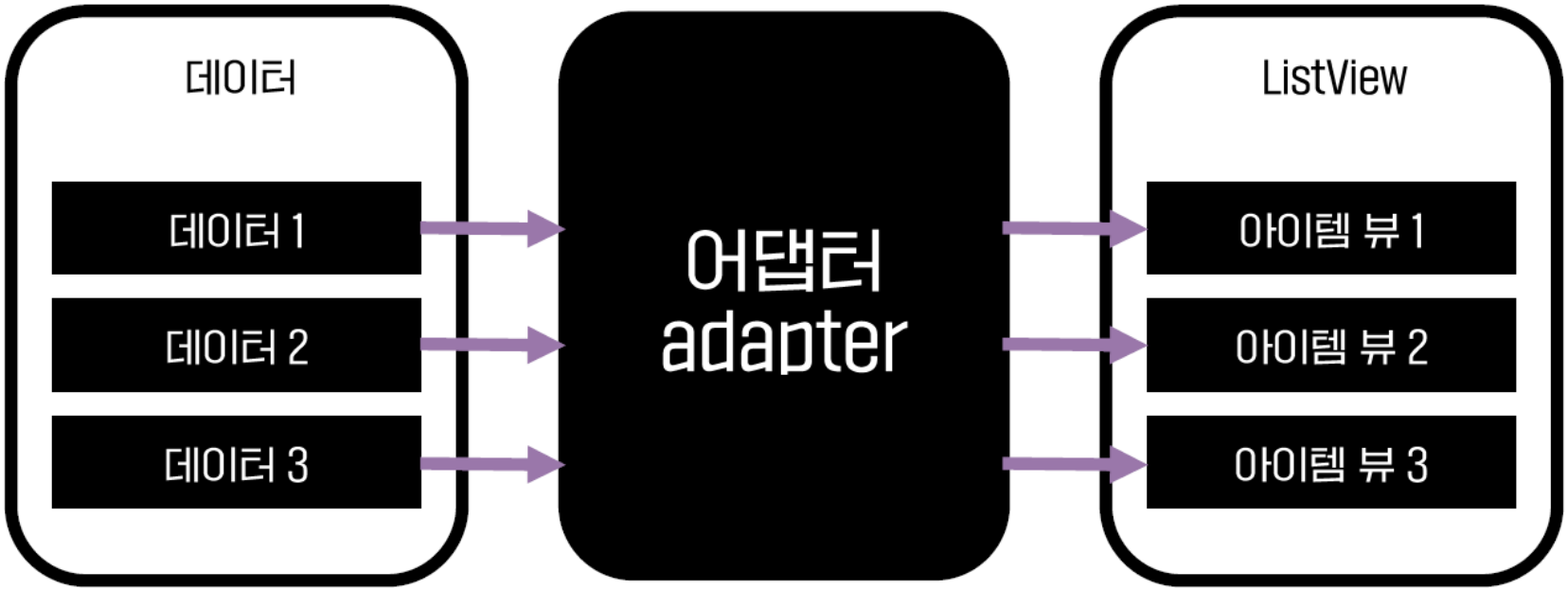
List View / Recycler View

ListView

- 화면에 데이터들을 리스트의 형태로 보여주는 View
- 데이터들을 리스트로 만들기 위해서는 어댑터를 만들어 줘야 한다.

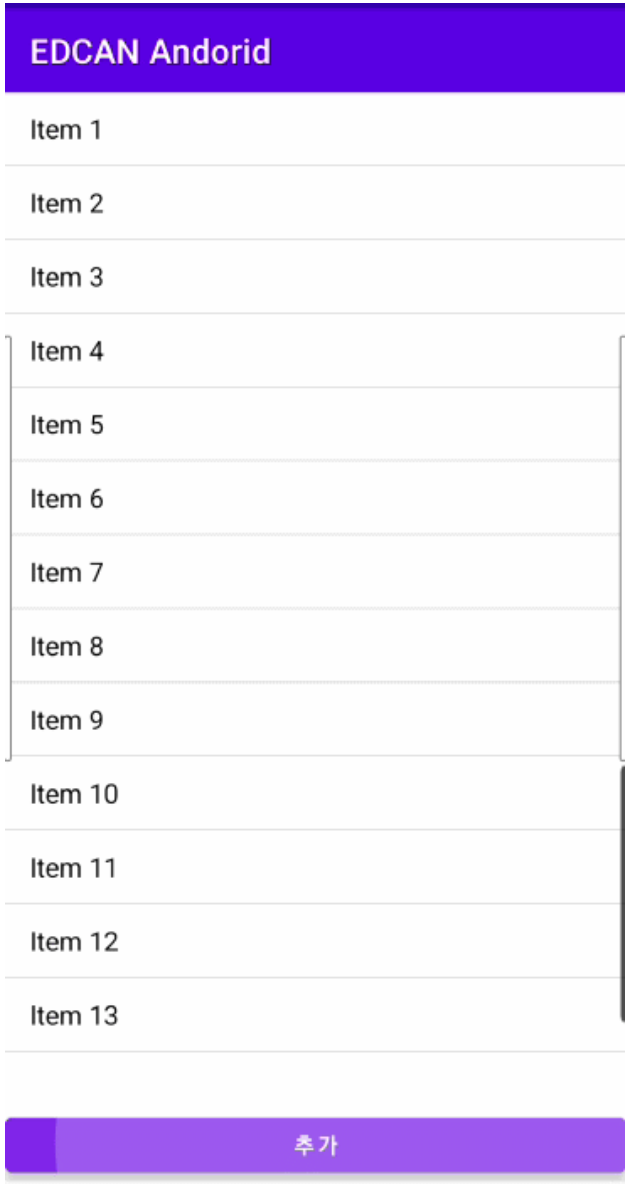


```xml
<ListView
    android:id="@+id/list_main_club"
    android:layout_width="0dp"
    android:layout_height="0dp"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toTopOf="@+id/btn_main_addItem"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    />
<Button
    android:id="@+id/btn_main_addItem"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

    android:text="추가"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    />
```

XML

```kotlin
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val dataList = mutableListOf<String>("Item 1", "Item 2", "Item 3", "Item 4", "Item 5")
        val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, dataList)
        findViewById<ListView>(R.id.list_main_club).adapter = adapter


        findViewById<Button>(R.id.btn_main_addItem).setOnClickListener {
            adapter.add("Item ${dataList.size + 1}")
            adapter.notifyDataSetChanged()
        }
    }
}
```

Kotlin



```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val dataList = mutableListOf<String>("Item 1", "Item 2", "Item 3", "Item 4", "Item 5")
        val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, dataList)
        findViewById<ListView>(R.id.list_main_club).adapter = adapter



        findViewById<Button>(R.id.btn_main_addItem).setOnClickListener {
            adapter.add("Item ${dataList.size + 1}")
            adapter.notifyDataSetChanged()
        }
    }
}
```

화면에 리스트로 보여줄 데이터

어댑터를 만들어준다

만들어준 어댑터를 연결시켜준다

리스트에 아이템을 추가한다.

리스트를 새로 그려준다.

ListView 어댑터 기능

add() : 데이터 추가

clear() : 데이터 전체 삭제

remove(item) : 데이터 삭제

getPostion(item) : 해당 아이템의 위치

notifyDataSetChanged() :리스트 새로 그리기

ListView 단점

- 리스트 아이템 커스텀을 할 수 없다.
- 리스트 아이템이 많아지면 화면에 보여주기 위한 시간이 오래 걸린다.
- 구글이 ListView 보다는 RecyclerView를 쓰라고 한다.

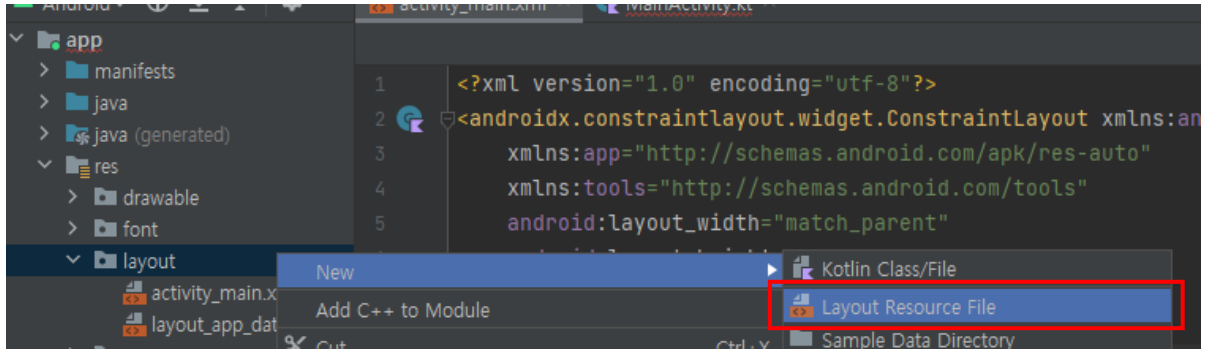
여러가지 이유 때문에 Recycler View를 사용한다.

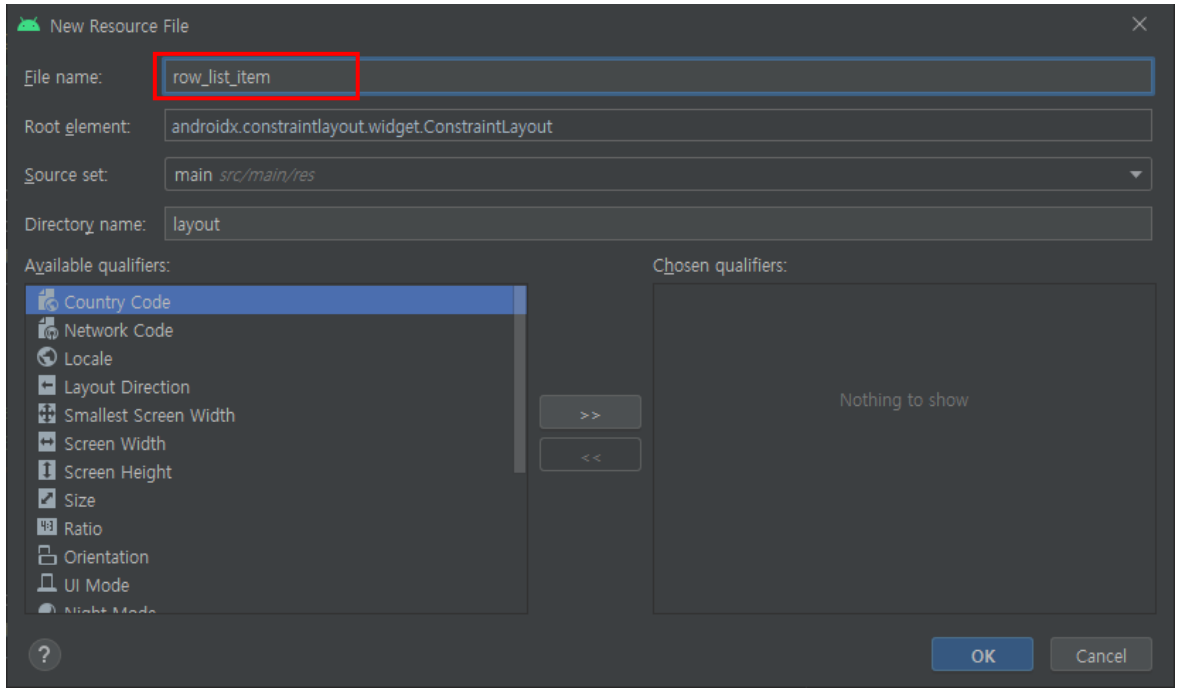
Recycler View

- 리스트 아이템 커스텀을 할 수 있다.
- 어댑터를 직접 만들어서 사용한다.
- ViewHolder를 만들어 줘야 한다.
- 리스트 아이템 레이아웃을 직접 만들어 줘야 한다.

Recycler View 사용법

1. 리스트 아이템 레이아웃 만들기





Recycler View 사용법

1. 리스트 아이템 레이아웃 만들기

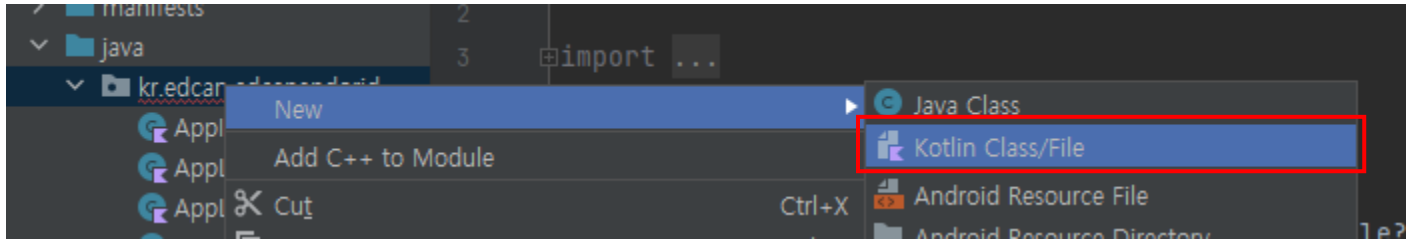
```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:paddingVertical="16dp" >
    <TextView
        android:id="@+id/txt_listItem_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="기본 값"
        android:textSize="20dp"

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        />
</androidx.constraintlayout.widget.ConstraintLayout>
```

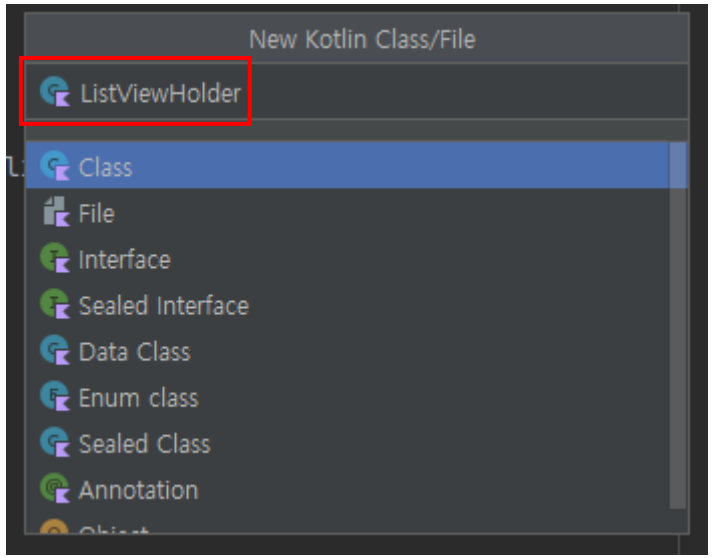
Recycler View - ViewHolder

- 리스트에서 아이템 하나를 클래스로 표현한 것
- 하나의 리스트 아이템에 있는 뷰들이 모여있다.
- 모든 ViewHolder은 **RecyclerView.ViewHolder** 추상 클래스를 상속 받는다.
- 객체 생성시 View를 하나 인자로 받는다.

Recycler View 사용법

2. ViewHolder 만들기





Recycler View 사용법

2. ViewHolder 만들기

```
class ListViewHolder(val row : View) : RecyclerView.ViewHolder(row) {
    val number = row.findViewById<TextView>(R.id.txt_listItem_number)
}
```

리스트 아이템의 뷰들을 가져와준다.

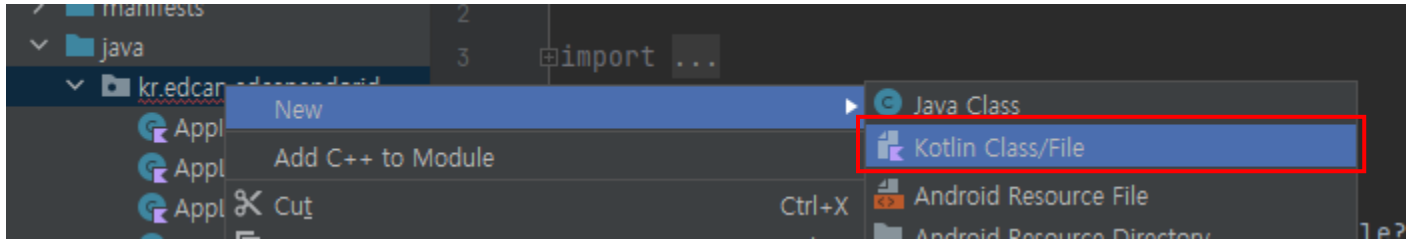
Recycler View - Adapter

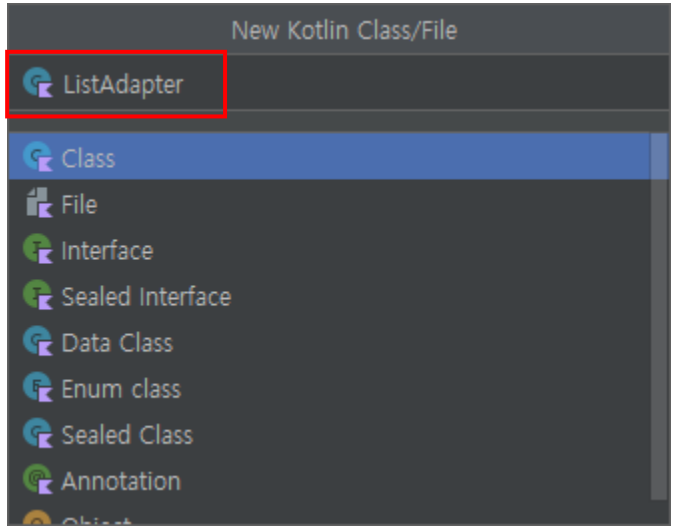
- 데이터들을 각각의 리스트 아이템으로 만들어 준다.
- RecyclerView.Adapter 추상 클래스를 상속 받아준다.

  이때 만들어준 ViewHolder**클래스**를 제네릭으로 넣어준다.

Recycler View 사용법

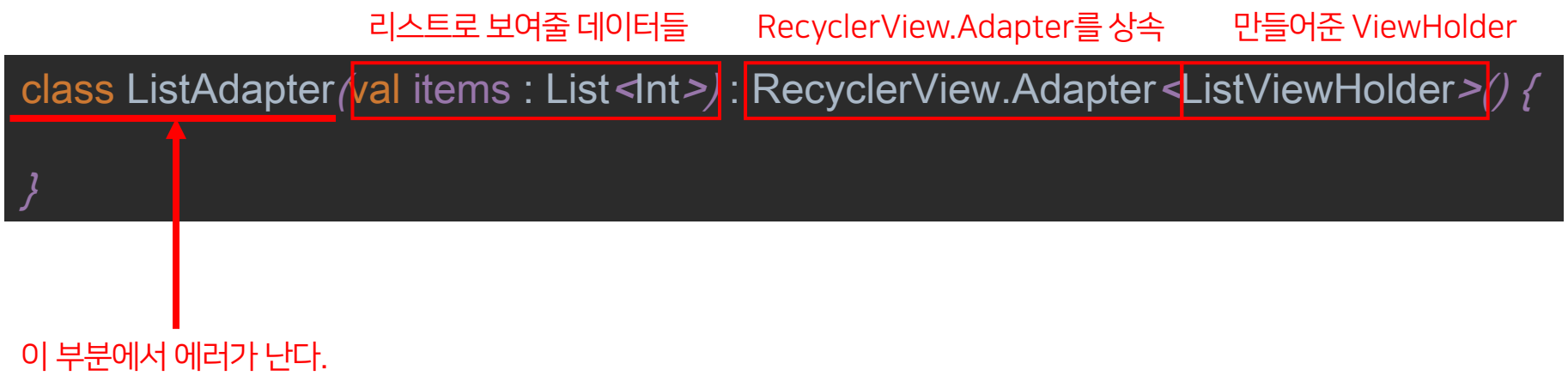
2. Adapter 만들기





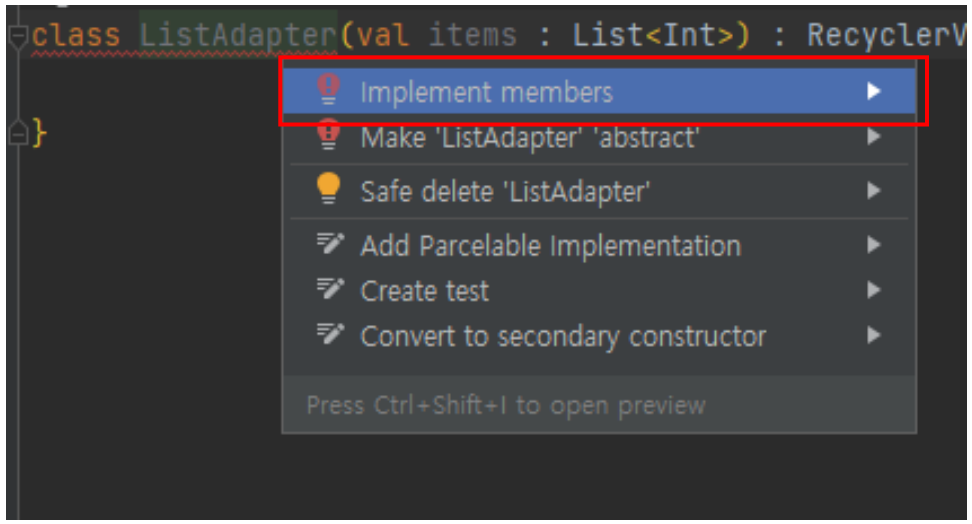
Recycler View 사용법

2. Adapter 만들기

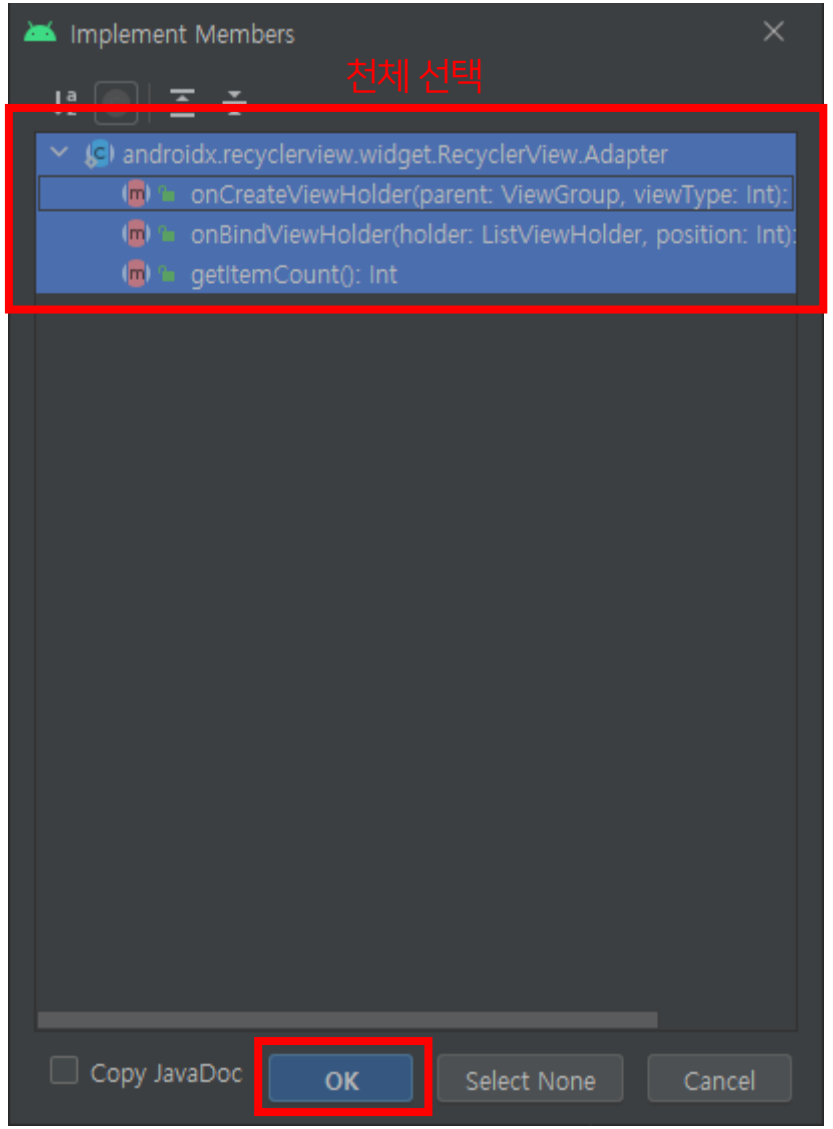


Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기



에러 난 부분에서 Alt + Enter



Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
class ListAdapter(val items : List<Int>) : RecyclerView.Adapter<ListViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
        TODO("Not yet implemented")
    } 리스트 아이템별 ViewHolder 만드는 메서드

    override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
        TODO("Not yet implemented")
    } 리스트 아이템에 데이터를 채워넣어주는 메서드

    override fun getItemCount(): Int {
        TODO("Not yet implemented")
    } 아이템의 전체 개수를 반환하는 메서드

}
```

Recycler.Adapter 추상 클래스는 3개의 메서드를 구현 해줘야 한다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {

    val view = LayoutInflater.from( parent.context ).inflate(R.layout.row_list_item, parent, false)

    return ListViewHolder(view)
}
```

만들어준 레이아웃 파일

LayoutInflater는 뷰를 동적으로 만들어 주는 역할을 한다.

만들어준 뷰를 ViewHolder로 만들어서 반환한다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
    holder.number.text = items[position].toString()
}
```

ViewHolder에 데이터를 채워 넣어 준다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
override fun getItemCount(): Int {
    return items.size
}
```

데이터들의 전체 개수를 반환하다.

Recycler View 사용법

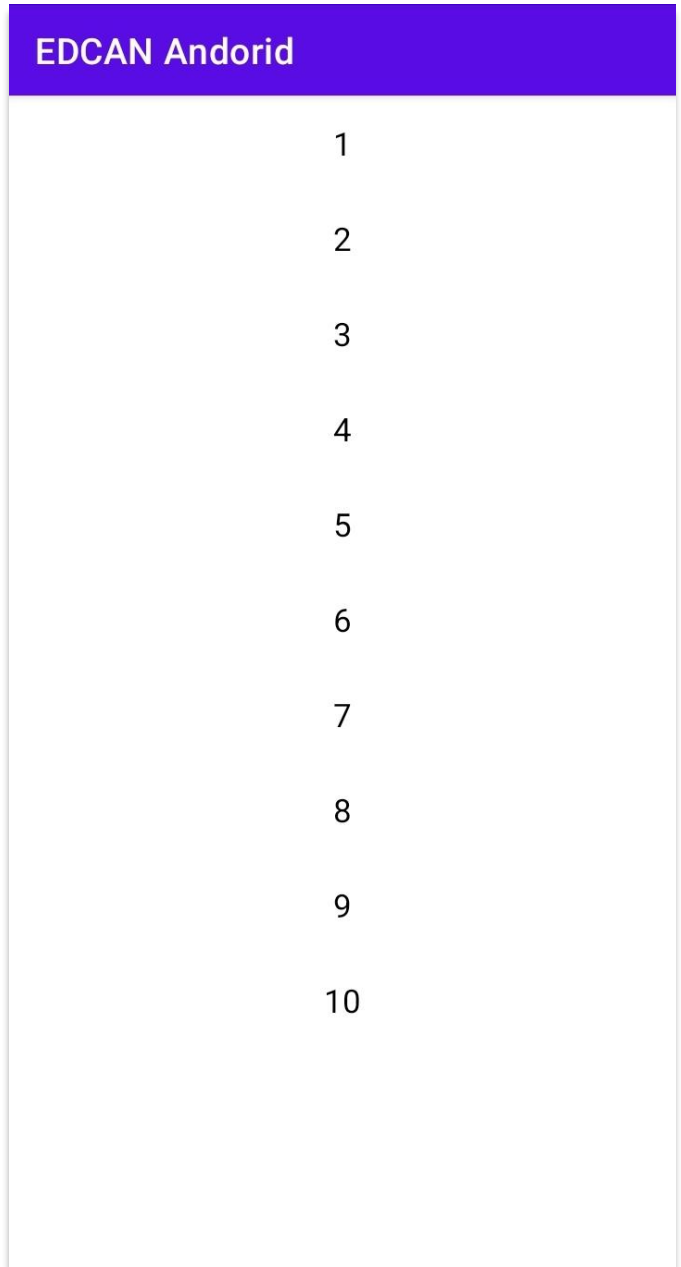
3. 리사이클러뷰와 어댑터를 연결해주기

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/list_main_club"
    android:layout_width="0dp"
    android:layout_height="0dp"

    tools:listitem="@layout/row_list_item"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    />
```

activity_main.xml

```
val recyclerView : RecyclerView = findViewById(R.id.list_main_club)

val datas = listOf(1, 2, 3, 4, 5, 6, 7, 8, ,9, 10)
val adapter = ListAdapter(datas)

recyclerView.adapter = adapter
recyclerView.layoutManager = LinearLayoutManager(this)
```

리스트가 보여지는 형식


EDCAN Andorid
1
2
3
4
5
6
7
8
9
10

Recycler View 사용법

리사이클러뷰 아이템 클릭 이벤트

```
class ListViewHolder(val row : View) : RecyclerView.ViewHolder(row) {
    val number = row.findViewById<TextView>(R.id.txt_listItem_number)

    init {
        row.setOnClickListener {
            Toast.makeText( row.context, "${number.text} 클릭", Toast.LENGTH_LONG ).show()
        }
    }
}
```

ViewHolder에서 들어온 view에 이벤트를 정의 해준다.

Recycler View 사용법

리사이클러뷰 데이터 추가하기

```
adapter.items.add(11)
adapter.notifyDataSetChanged()
```

이때 items가 mutable list일 때 만 추가 가능하다.

리스트를 새로 그린다.