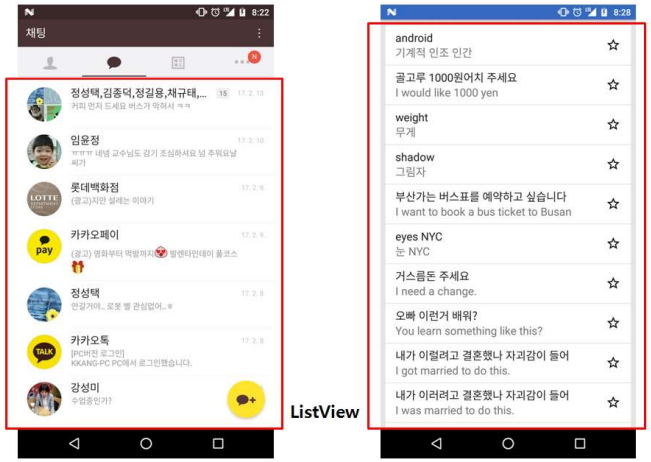
06

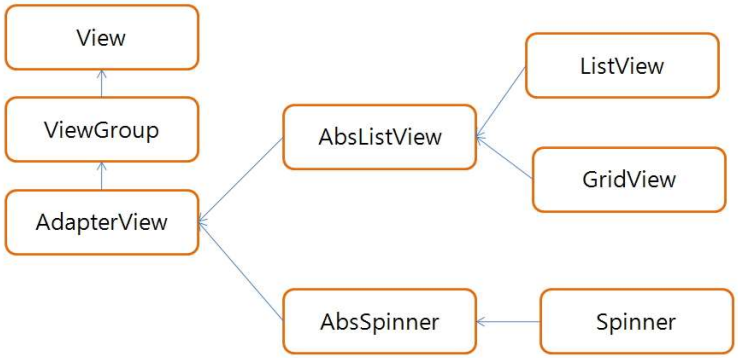
06-1. RecyclerView

다양한 뷰 활용

Adapter 와 AdapterView

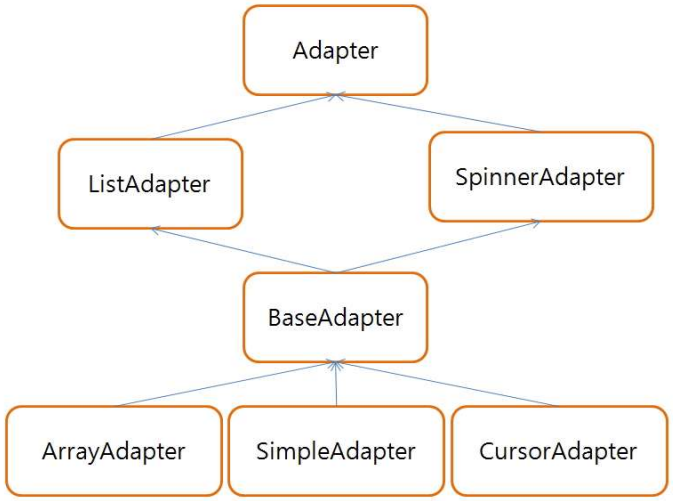
- AdapterView는 항목을 나열하는 뷰





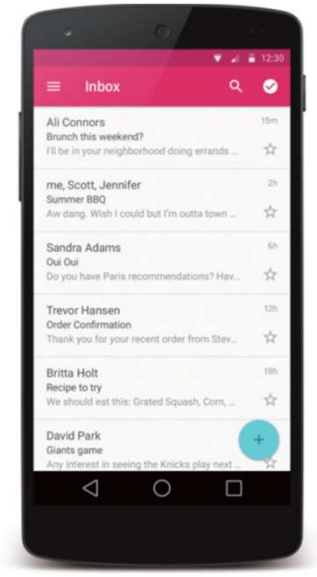
Adapter 와 AdapterView

- AdapterView에 항목이 나열되어 화 면에 데이터가 나오려면 꼭 Adapter라는 클래스를 이용
- Adapter에게 일을 시 키고 Adapter가 AdapterView를 완성해주는 구조



리사이클러 뷰

- 구성 요소
  - ViewHolder(필수) : 항목에 필요한 뷰 객체를 가진다
  - Adapter(필수) : 항목을 구성
  - LayoutManager(필수) : 항목을 배치
  - ItemDecoration(옵션) : 항목을 꾸민다



리사이클러 뷰

- 뷰 홀더 준비
  - 각 항목에 해당하는 뷰 객체를 가지는 뷰 홀더는 RecyclerView.ViewHolder를 상속받아 작성

• 뷰 홀더 준비

```
class MyViewHolder(val binding: ItemMainBinding): RecyclerView.ViewHolder(binding.root)
```

- 어댑터 준비
  - 각 항목을 만들어 주는 역할

• 어댑터 준비

```
class MyAdapter(val binding: ItemMainBinding):
    RecyclerView.Adapter<RecyclerView.ViewHolder>() {
    override fun getItemCount(): Int {
        TODO("Not yet implemented")
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
    RecyclerView.ViewHolder {
        TODO("Not yet implemented")
    }
    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
        TODO("Not yet implemented")
    }
}
```

리사이클러 뷰

- getItemCount() : 항목 개수를 판단하려고 자동으로 호출
- onCreateViewHolder(): 항목의 뷰를 가지는 뷰 홀더를 준비하려고 자동으로 호출
- onBindViewHolder(): 뷰 홀더의 뷰에 데이터를 출력하려고 자동으로 호출

• 뷰에 데이터 출력

```
override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
    Log.d("kkang","onBindViewHolder : $position")
    val binding = (holder as MyViewHolder).binding
    // 뷰에 데이터 출력
    binding.itemData.text = datas[position]
    // 뷰에 이벤트 추가
    binding.itemRoot.setOnClickListener {
        Log.d("kkang", "item root click : $position")
    }
}
```

• 항목의 개수 구하기

```
override fun getItemCount(): Int = datas.size
```

• 항목 구성에 필요한 뷰 홀더 객체 준비

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder =
    MyViewHolder(ItemMainBinding.inflate(LayoutInflater.from(parent.context),
        parent, false))
```

리사이클러 뷰

- 리사이클러 뷰 출력

• 리사이클러 뷰 출력

```
class RecyclerViewActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val binding = ActivityRecyclerViewBinding.inflate(layoutInflater)
        setContentView(binding.root)
        val datas = mutableListOf<String>()
        for(i in 1..10){
            datas.add("Item $i")
        }
        binding.recyclerView.layoutManager = LinearLayoutManager(this)
        binding.recyclerView.adapter = MyAdapter(datas)
        binding.recyclerView.addItemDecoration(DividerItemDecoration(this,
        LinearLayout Manager.VERTICAL))
    }
}
```

리사이클러 뷰

- 항목을 동적으로 추가·제거
  - 항목을 구성하는 데이터에 새로운 데이터를 추가하거나 제거한 후 어댑터의 notifyDataSetChanged() 함수를 호출

• 항목 추가

```
datas.add("new data")
adapter.notifyDataSetChanged()
```

리사이클러 뷰

- 웃 매니저는 어댑터로 만든 항목을 리사이클러 뷰에 배치
  - LinearLayoutManager : 항목을 가로나 세로 방향으로 배치
  - GridLayoutManager : 항목을 그리드로 배치
  - StaggeredGridLayoutManager : 항목을 높이가 불규칙한 그리드로 배치

- 항목을 가로·세로 방향으로 배치
  - LinearLayoutManager를 사용

• 항목을 세로로 배치

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

▶ 실행 결과

리사이클러 뷰

- LinearLayoutManager의 orientation값을 Linear LayoutManager.HORIZONTAL로 지정

• 항목을 가로로 배치

```
val layoutManager = LinearLayoutManager(this)
layoutManager.orientation =
    LinearLayoutManager.HORIZONTAL
binding.recyclerView.layoutManager = layoutManager
```

▶ 실행 결과

리사이클러 뷰

- 그리드로 배치하기
  - GridLayoutManager를 이용
  - 생성자의 숫자는 그리드에서 열의 개수를 뜻
  - 방향을 설정할 수 있다

  - 가로로 설정하려면 생성자에 GridLayoutManager.HORIZONTAL을 지정

• 그리드에서 항목을 가로로 배치

```
val layoutManager = GridLayoutManager(this, 3,
GridLayoutManager.HORIZONTAL, false)
binding.recyclerView.layoutManager = layoutManager
```

▶ 실행 결과

리사이클러 뷰

- GridLayoutManager 생성자의 네 번째 매개변수에 Boolean값을 설정

• 그리드에서 항목을 오른쪽부터 배치

```
val layoutManager = GridLayoutManager(this, 3,
GridLayoutManager.HORIZONTAL, true)
binding.recyclerView.layoutManager = layoutManager
```

▶ 실행 결과

리사이클러 뷰

- 높이가 불규칙한 그리드로 배치하기
  - StaggeredGridLayoutManager는 뷰의 크기가 다르면 지그재그 형태로 배치

• 지그재그 그리드 형태로 배치

```
val layoutManager = StaggeredGridLayout
Manager(2, StaggeredGridLayoutManager.
VERTICAL)
binding.recyclerView.layoutManager =
layoutManager
```

▶ 실행 결과

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare