S7 Inf A3 — Java for Graphical and Mobile Programming · Centrale Lyon

Stéphane Derrode

CM7

Layouts, Navigation & Intents

- LinearLayout with weights — flexible proportions
- ConstraintLayout — the modern layout
- RecyclerView — scrollable lists
- Explicit Intent — launching a second Activity
- Passing data — putExtra / getIntent().getStringExtra()

Module Overview

Part 1 — Java Fundamentals	Part 2 — GUI with Swing	Part 3 — Android
CM1 From Python to Java	**CM4** Swing: Components & Layouts	**CM6** Android Architecture
CM2 Collections, Exceptions & I/O	**CM5** Graphics 2D & Events	**CM7** Layouts, Navigation & Intents
CM3 Inheritance, Interfaces		**CM8** Persistence & Threads

Projects: ★ Spider Game (Part 2) ★ Calculator / Currency Converter (Part 3)

Exam 50% + BEs 50%

Stéphane Derrode — Centrale Lyon · S7 Inf A3

CM7 — Agenda

01 LinearLayout — weights & gravity

02 ConstraintLayout — modern layout

03 RecyclerView — scrollable lists

04 Explicit Intent — second Activity

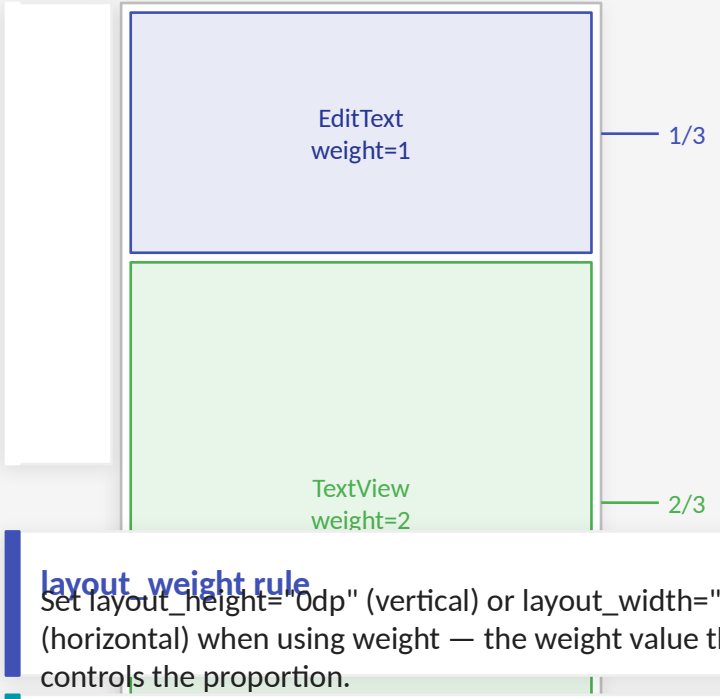
05 putExtra / getExtra — passing data

06 NotesPad v2 — full app

LinearLayout — Weights & Gravity

```
<!-- Vertical LinearLayout with weights -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- Takes 1/3 of available height -->
    <EditText
        android:id="@+id/noteInput"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:hint="@string/hint_note"
        android:gravity="top"/>

    <!-- Takes 2/3 of available height -->
    <TextView
        android:id="@+id/preview"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:gravity="start|top"/>

    <!-- Fixed height — not weighted -->
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_save"/>
```

Result



layout_weight rule

Set layout_height="0dp" (vertical) or layout_width="0dp" (horizontal) when using weight — the weight value then controls the proportion.

gravity vs layout_gravity

gravity — aligns content INSIDE the view (text alignment).
layout_gravity — aligns the view INSIDE its parent.

ConstraintLayout — The Modern Layout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <EditText
        android:id="@+id/noteInput"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/hint_note"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toStartOf="@id/saveButton"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginEnd="8dp"/>

    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btn_save"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

Constraint rules

`constraintStart_toStartOf`

Left edge of this view → left edge of target

`constraintEnd_toEndOf`

Right edge → right edge of target

`constraintTop_toTopOf`

Top edge → top edge of target

`constraintBaseline_to BaselineOf`

Text baseline aligned with target's baseline

`layout_width="0dp"`

Width fills the space between constraints (match_constraint)

Use the visual editor

In Android Studio: drag the blue handles to create constraints. The XML is generated automatically. Every view needs at least one horizontal AND one vertical constraint, or it

RecyclerView — Scrollable Lists

RecyclerView = Adapter pattern — 3 moving parts

Your data

`List<String> notes`

→

Adapter

`NoteAdapter extends RecyclerView.Adapter`

pro

ViewHolder

`Holds one row's views (TextView…)`

```
// 1. item_note.xml — one row layout
// (res/layout/item_note.xml)
// <LinearLayout>
//   <TextView android:id="@+id/noteText"
//             android:layout_width="match_parent"
//             android:layout_height="wrap_content"/>
// </LinearLayout>

// 2. NoteAdapter.java
public class NoteAdapter extends
        RecyclerView.Adapter<NoteAdapter.NoteHolder> {

    private List<String> notes;
    private OnNoteClick  listener;

    public NoteAdapter(List<String> notes,
                       OnNoteClick listener) {
        this.notes    = notes;
        this.listener = listener;
    }
    // Interface for click callback
    public interface OnNoteClick {
        void onClick(int position);
    }
```

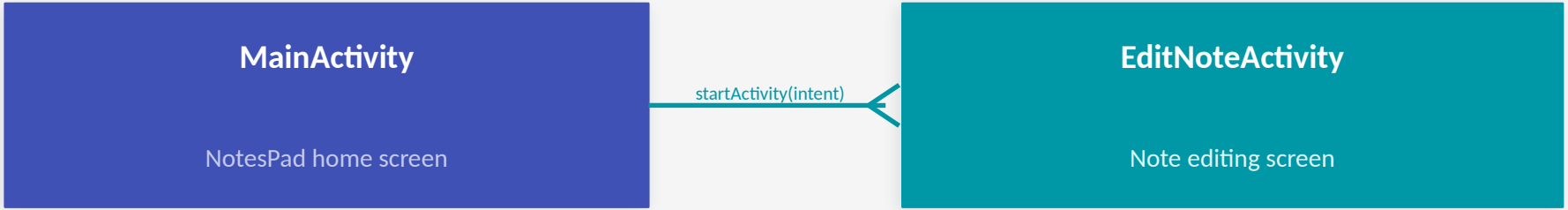
```
    @Override  // called for each row
    public NoteHolder onCreateViewHolder(
            ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(
            parent.getContext()).inflate(
            R.layout.item_note, parent, false);
        return new NoteHolder(v);
    }

    @Override  // bind data to a row
    public void onBindViewHolder(
            NoteHolder holder, int position) {
        holder.noteText.setText(
            notes.get(position));
        holder.itemView.setOnClickListener(
            v -> listener.onClick(position));
    }

    @Override
    public int getItemCount() {
        return notes.size();
    }
}
```

RecyclerView — XML & item_note Layout

```
<!-- activity_main.xml — add RecyclerView -->
<androidx.constraintlayout.widget.ConstraintLayout ...>

    <EditText
        android:id="@+id/noteInput"
        ...
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toStartOf="@id/addButton"/>

    <Button
        android:id="@+id/addButton"
        android:text="@string/btn_add"
        ...
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintBaseline_toBaselineOf="@id/noteInpu
t"/>

    <!-- The scrollable list -->
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/notesList"
        android:layout_width="0dp"
        android:layout_height="0dp"

app:layout_constraintTop_toBottomOf="@id/noteInput"
```

```
<!-- res/layout/item_note.xml -->
<!-- One row of the list -->
<LinearLayout
    xmlns:android=

"http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="12dp"

android:background="?attr/selectableItemBackgroun
d">

    <TextView
        android:id="@+id/noteText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="16sp"/>

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_edit"
        android:contentDescription=
          "@string/edit_note"/>
```

Explicit Intent — Launching a Second Activity




```
// Step 1: Create the Intent
//   (this = current Context, target class)
Intent intent = new Intent(
    this, EditNoteActivity.class);

// Step 2: Attach data (optional)
intent.putExtra("NOTE_TEXT", noteText);
intent.putExtra("NOTE_INDEX", position);

// Step 3: Launch it
startActivity(intent);


// ─── In EditNoteActivity.java ──────────

@Override
protected void onCreate(Bundle saved) {
    super.onCreate(saved);
    setContentView(R.layout.activity_edit_note);
```

Register in Manifest!

Every Activity must be declared in AndroidManifest.xml. Android Studio adds it automatically when you use File → New → Activity. If missing → app crashes with ActivityNotFoundException.

```
<!-- AndroidManifest.xml -->
<activity
    android:name=".EditNoteActivity"
    android:label="@string/edit_note_title"
    android:parentActivityName=
        ".MainActivity"/>
<!-- parentActivityName adds the back arrow
-->
```

putExtra key convention

Use SCREAMING_SNAKE_CASE for keys and prefix with your class name: "NOTE_TEXT" not just "text". Avoids conflicts when passing intents between apps

Getting a Result Back — ActivityResultLauncher

When the launched Activity must return data (e.g. saved note text)

```
// ── In MainActivity.java ──────────────────────────
// 1. Register the launcher (in the Activity field)
private ActivityResultLauncher<Intent> editLauncher =
    registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if (result.getResultCode() == RESULT_OK) {
                Intent data = result.getData();
                if (data != null) {
                    String updated = data.getStringExtra(
                        "UPDATED_NOTE");
                    int index = data.getIntExtra(
                        "NOTE_INDEX", -1);
                    if (index >= 0) {
                        notes.set(index, updated);
                        adapter.notifyItemChanged(index);
                    }
                }
            }
        });

// 2. Launch the Activity
private void editNote(int position) {
    Intent intent = new Intent(
        this, EditNoteActivity.class);
    intent.putExtra("NOTE_TEXT",
        notes.get(position));
```

```
// ── In EditNoteActivity.java ──────────────
// When the user clicks Save:
private void saveNote() {
    EditText editor = findViewById(
        R.id.editor);
    String text = editor.getText()
                        .toString().trim();

    Intent result = new Intent();
    result.putExtra("UPDATED_NOTE", text);
    result.putExtra("NOTE_INDEX",
        getIntent().getIntExtra(
            "NOTE_INDEX", -1));

    // Send result back to MainActivity
    setResult(RESULT_OK, result);
    finish(); // close this Activity
}

// If user presses back without saving:
// setResult(RESULT_CANCELED) is called
// automatically by Android.
```

strings.xml — Never Hardcode Text

```xml
<!-- res/values/strings.xml (default — French) -->
<resources>
    <string name="app_name">NotesPad</string>
    <string name="hint_note">
        Saisissez une note…</string>
    <string name="btn_add">Ajouter</string>
    <string name="btn_save">Enregistrer</string>
    <string name="edit_note_title">
        Modifier la note</string>
    <string name="error_empty">
        La note ne peut pas être vide</string>
    <string name="note_deleted">
        Note supprimée</string>
</resources>

<!-- res/values-en/strings.xml (English) -->
<resources>
    <string name="app_name">NotesPad</string>
    <string name="hint_note">
        Enter a note…</string>
    <string name="btn_add">Add</string>
    <string name="btn_save">Save</string>
    <string name="error_empty">
        Note cannot be empty</string>
</resources>
```

Using string resources in Java

```java
// In any Activity:
String text = getString(R.string.btn_add);

// With format arguments:
// <string name="count">%d note(s)</string>
String msg = getString(
    R.string.count, notes.size());

// In Toast:
Toast.makeText(this,
    R.string.error_empty,
    Toast.LENGTH_SHORT).show();

// In XML (automatic — no Java needed):
android:text="@string/btn_add"
```

How Android picks the right file

Android reads the device locale and looks for values-xx/ first. If not found, falls back to values/. values/ is your default language — if only one language needed, only values/ is required.

Folder structure

res/values/strings.xml ← default (FR)
res/values-en/strings.xml ← English
res/values-es/strings.xml ← Spanish
Same principle for layouts: res/layout-land/ for landscape

NotesPad v2 — Full App Architecture

Two-Activity app

MainActivity

- `EditText noteInput`
- `Button addButton`
- `RecyclerView notesList`
- `List<String> notes`
- `NoteAdapter adapter`

Intent + putExtra

result ← back

EditNoteActivity

- `EditText editor`
- `Button saveButton`
- `Button deleteButton`

What's new in v2

✓ RecyclerView with NoteAdapter

✓ EditNoteActivity (second screen)

✓ Intent + putExtra to pass note

✓ ActivityResultLauncher for result

AndroidManifest.xml declares MainActivity + EditNoteActivity, app name, minSdk=26, targetSdk=33

CM7 Quick Reference

Task	Code / XML
Weighted layout	`android:layout_weight="1"  android:layout_height="0dp"  (vertical)`
ConstraintLayout ref	`app:layout_constraintStart_toStartOf="parent"  (and end, top, bottom)`
RecyclerView XML	`<androidx.recyclerview.widget.RecyclerView  android:id="@+id/list" .../>`
Setup RecyclerView	`rv.setLayoutManager(new LinearLayoutManager(this));  rv.setAdapter(adapter);`
Launch Activity	`Intent i = new Intent(this, OtherActivity.class);  startActivity(i);`
Pass data	`intent.putExtra("KEY", value);  /  getIntent().getStringExtra("KEY");`
Return result	`setResult(RESULT_OK, intent);  finish();  (in the launched Activity)`
Notify list update	`adapter.notifyItemInserted(pos);  notifyItemChanged(pos);` `notifyItemRemoved(pos);`
String resource	`XML: @string/name  /  Java: getString(R.string.name)`

CM7 — Key Takeaways

LinearLayout	layout_weight with 0dp size — flexible proportions; gravity vs layout_gravity
ConstraintLayout	Every view needs horizontal + vertical constraints; 0dp = fill between constraints
RecyclerView	Adapter + ViewHolder pattern; onCreateViewHolder / onBindViewHolder / getItemCount
Intent	new Intent(this, Target.class); startActivity(i) — register in Manifest
putExtra	intent.putExtra("KEY", value); getIntent().getStringExtra("KEY") in target
Result back	registerForActivityResult(); setResult(RESULT_OK, data); finish()

Up next → TD7: NotesPad v2 — RecyclerView + navigation