# Paso de Datos y Menús

Dra. Mª del Puerto Paule Ruiz

# Ejercicio

- Añadir animaciones entre activities
  - pasodatos12MT →Material Design(cont.)

# Paso de datos entre dos activities

- Hacemos uso de la clase Bundle
  - Clase que “envuelve” los tipos básicos de datos en Android
- Para “envolver” un tipo de dato se utiliza el método putTipodeDato:

  void putTipoDeDato (String key, TipoDeDato value)

key: Indentifica el dato a pasar

Value: Dato a pasar

# Paso de datos entre dos activities

- TipoDeDato= String, Boolean, Char, Double…
  - Por ejemplo, para envolver Strings utilizaríamos putString

  mbundle.putString(String key, String value)
- Por último, se debe añadir el envoltorio al intento:

  mIntent.putExtras(mBundle)

# Recoger los datos pasados

- Es necesario acceder al intento que creó la actividad. Una vez obtenido el intento, obtenemos el “envoltorio” para recupera los datos

```
Bundle mBundleRecibido = getIntent().getExtras();
```

- Para acceder a los datos, lo haremos a través de la key en el método getTipoDeDato:

```
String mString = mBundleRecibido.getString(key);
```

# Ejercicio (PasoDatos31MT)

- Pasar una cadena de texto de una activity a otra.
- Esta segunda la muestre en un textView

# Paso de datos (Serializable)

- Android propone interface **Parcelable**.
  - **Parcelable** es una interface específica donde implementar la serialización.
  - **Objetivo:** Ser más eficientes que Serializable
- Métodos @Override:
  - describeContents: Tipo especial de objeto
  - writeToParcel: Escribe el objeto parcelable
- Más métodos:
  - Creación de varias instancias del objeto parcelable

```
public static final Parcelable.Creator<libro>
CREATOR = new Parcelable.Creator<libro>()
```

# Envío y recuperación objeto Parcelable

- Paso y envío de objeto parcelable (MainActivity):
  - public static String OBJETO_KEY = "OBJETO_KEY";
  - mIntent.putExtra(OBJETO_KEY, parcelable);
  - startActivity (mIntent)
- Recuperación objeto parcelable:
  - Bundle b = getIntent().getExtras();
  - objeto o = b.getParcelable(MainActivity.OBJETO_KEY);

# Ejercicio (PasoDatos12MTParcelable)

- Crear un objeto parcelable libro con las propiedades de título,  autor (y opcionalmente fecha de publicación)
- MainActivity: Recoge los datos de título y autor y libro
- SecondActivity: Muestra los datos del libro concreto

# Menus

- Existen tres tipos de menu:
  - OptionsMenu
  - ContextMenu
  - PopupMenu

# Options Menu

- Dentro de /res hay un directorio “menu”
- Dentro de “menu” hay fichero xml que contiene la definición de menú:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   tools:context="com.example.android.pasodatos31.SecondActivity">
   <item android:id="@+id/action_settings" android:title="@string/action_settings"
      android:orderInCategory="100" app:showAsAction="never" />
   <item android:id="@+id/action_fin"
      android:title="@string/Finalizar"
      android:orderInCategory="100"
      android: icon=“@android:drawable/ic-delete
      app:showAsAction=“ifRoom" />
</menu>
```

# Options Menu

- El método onCreateOptionsMenu (Menu menu) es llamado en la creación del menú.
- En la Activity, donde queremos ver el menú:

```
public boolean onCreateOptionsMenu(final Menu menu) {
        // Inflate the menu
        getMenuInflater().inflate(R.menu.menu, menu);
        return true;     }
```

# Options Menu

- Para añadir funcionalidad a los elementos del menú se utiliza el evento:

public boolean onOptionsItemSelected(final MenuItem item)

- Se activa cuando el usuario selecciona un elemento
- El método recibe como parámetro el elemento seleccionado
- Según el identificador del elemento debemos de realizar la opción adecuada

# Options Menu

```
public boolean onOptionsItemSelected(MenuItem item) {

    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
```

# Ejercicio (PasoDatos31MT)

- Añadir en la **segunda actividad** una opción de menú que permita finalizarla
  - Explicar las diferentes opciones del menú

# Retornar resultado (Actividad principal)

- Se puede lanzar una Activity de manera que ésta devuelva un resultado a la Activity que la lanzó
- Para ello debe utilizarse el método:

startActivityForResult(Intent mIntent,  int RequestCode)

Intent que lanza la nueva Activity

RequestCode: Indentifica la Activity a lanzar (por si se lanzan varias)

# Devolver un resultado (Actividad principal)

- Una vez finalizada la Activity lanzada, el evento invocado para recibir los datos es:

```
protected void onActivityResult(int requestCode,  int resultCode,  Intent data)
```

requestCode: Indentifica la actividad lanzada

resultCode: código para conocer si son correctos los datos devueltos

data: Datos devueltos por la actividad

# Devolver un resultado (Ejemplo onActivityResult)

```
protected void onActivityResult(int requestCode,  int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        /*Recogemos datos*/
                if (resultCode == RESULT_OK) {
                        final Bundle mBundle = data.getExtras();
                        /*Código necesario*/
                }
                else if (resultCode == RESULT_CANCELED) {
        /*Código necesario*/
                }

}
```

# Devolver un resultado (Actividad secundaria)

- Los datos se pasan a través del Bundle

  void putTipoDeDato (String key, TipoDeDato value)

- El método setResult estable el código resultante y el Intent que contiene los datos:

  ```
  final Intent resultIntent = new Intent();
  resultIntent.putExtras(mBundle);
  setResult(RESULT_OK, resultIntent);
  ```

- Se finaliza la actividad con finish()

# Ejercicio (PasoDatos41MT)

- Dada dos activities:
  - En la primera introducir una cadena en un editText
  - En la segunda escribir la cadena introducida en la primera
  - Devolver a la primera activity la longitud de la cadena obtenida en la segunda