

Block 9 – Activities und Intents (Navigation I)



- Interaktion (Intra-App)
 - ▶ Intent (ohne Parameter/ohne Rückgabe)
 - ▶ Intent (mit Parameter/ohne Rückgabe)
 - ▶ Intent (ohne Parameter/mit Rückgabe)
 - ▶ Intent (mit Parameter/mit Rückgabe)
 - ▶ Intent (mit Parameterobjekt/ohne Rückgabe)
- Interaktion (Inter-App)
 - ▶ Intent (mit Parameter/ohne Rückgabe)
 - ▶ Intent (mit Parameter/mit Rückgabe)
- Activity Stack
 - ▶ Standardverhalten
 - ▶ New Task/Clear Task
 - ▶ Clear Top
 - ▶ Single Top

Block 9 – Lernziele

In diesem Block werden Sie lernen . . .

- wie die Navigation zwischen verschiedenen Activities möglich ist.
- wie Parameter und Rückgabewerte zwischen Activities übermittelt werden können.
- wie sich der Activity Stack durch den Aufruf/das Verlassen von Activities auf- und abbaut.
- wie komplexe Objekte zwischen Activities ausgetauscht werden können.
- wie externe Anwendungen bzw. deren Activities aus einer eigenen Activity aufgerufen werden können.

9.1 Interaktion (Intra-App)

Zwischen Activities einer mobilen Anwendung gibt es folgende Interaktionsmuster:

- Aufruf einer Activity ohne Parameter ohne Rückgabewert
- Aufruf einer Activity mit Parameter ohne Rückgabewert
- Aufruf einer Activity ohne Parameter mit Rückgabewert
- Aufruf einer Activity mit Parameter mit Rückgabewert
- Aufruf einer Activity mit komplexen Datentypen (Objekten)

Für alle Aufrufe werden `Intents` (dt. Absichten) genutzt.

9.1.1 Intent (ohne Parameter/ohne Rückgabe)

Aufruf einer Activity ohne Parameter ohne Rückgabewert:

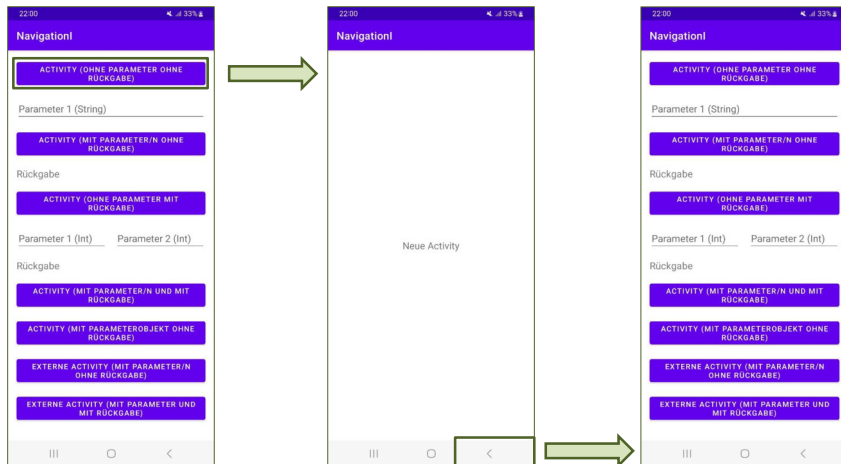
```
val intent = Intent(applicationContext ,  
    Activity_noParams_noResult :: class.java)  
startActivity(intent)
```

Gibt an, dass aus einer Activity (z.B. MainActivity.kt) die Activity Activity_noParams_noResult.kt aufgerufen werden soll.

Der eigentliche Aufruf wird dann mit startActivity(intent) ausgelöst.

9.1.1 Intent (ohne Parameter/ohne Rückgabe)

Beispiel (Aufruf einer Activity ohne Parameter ohne Rückgabewert):



9.1.2 Intent (mit Parameter/ohne Rückgabe)

Aufruf einer Activity mit Parameter ohne Rückgabewert:

```
val intent = Intent(applicationContext ,  
    Activity_params_noResult :: class.java)  
intent.putExtra(R.id.editTextParameter11.toString() ,  
    editText.text.toString());  
startActivity(intent)
```

Hier wird mit `putExtra()` ein Parameter übergeben.

Der Parametername wird mit der Konstanten `R.id.editTextParameter11.toString()` beschrieben, der Parameterwert ist ein String (Wert des Textfeldes).

9.1.2 Intent (mit Parameter/ohne Rückgabe)

Aufruf einer Activity mit Parameter ohne Rückgabewert:

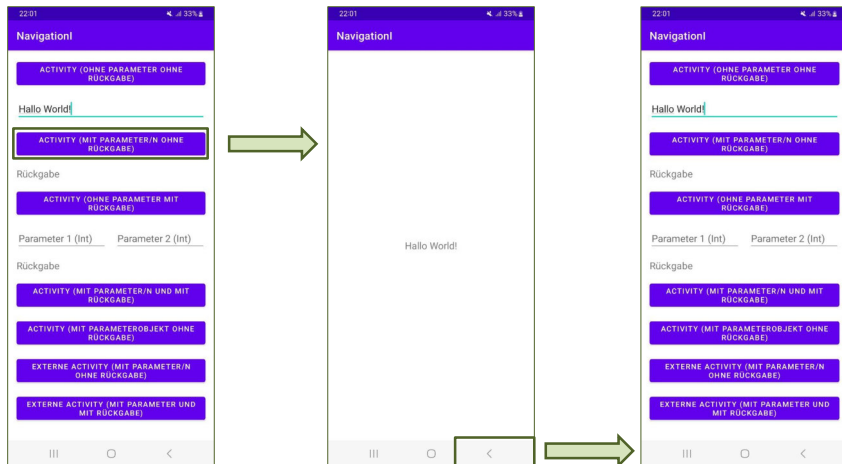
```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
    if (intent.hasExtra(R.id.editTextParameter11.toString())) {  
        val parameter1: String? = intent.getStringExtra(  
            R.id.editTextParameter11.toString())  
        textView = findViewById<TextView>(R.id.textView2)  
        textView.setText(parameter1)  
    }  
    ...  
}
```

In der Klasse `Activity_params_noResult.kt` muss der Parameterwert noch entgegengenommen werden.

Hier wird mit `hasExtra()` geprüft ob ein entsprechender Wert vorliegt. Falls dies der Fall ist wird die typspezifische Methode `getStringExtra()` zum Auslesen genutzt.

9.1.2 Intent (mit Parameter/ohne Rückgabe)

Beispiel (Aufruf einer Activity mit Parameter ohne Rückgabewert):



9.1.3 Intent (ohne Parameter/mit Rückgabe)

Aufruf einer Activity ohne Parameter mit Rückgabewert:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
    launch_Activity_noparams_result = registerForActivityResult(  
        ActivityResultContracts.StartActivityForResult(),  
        object : ActivityResultCallback<ActivityResult> {  
            override fun onActivityResult(result: ActivityResult?) {  
                if (result!!.resultCode == Activity.RESULT_OK) {  
                    val data : Intent? = result.data  
                    // Data-type specific i. e. getCharSequenceExtra()  
                    val result : String = data?.getCharSequenceExtra(  
                        Activity_noParams_result.RETURN_VALUE).toString()  
                    val textView = findViewById<TextView>(  
                        R.id.textViewReturn11)  
                    textView.text = result  
                }  
            }  
        }) ...  
}
```

In der Activity (z.B. MainActivity.kt) wird zunächst ein `ActivityResultLauncher` erstellt, an den später der Aufruf (Intent) gebunden wird.

9.1.3 Intent (ohne Parameter/mit Rückgabe)

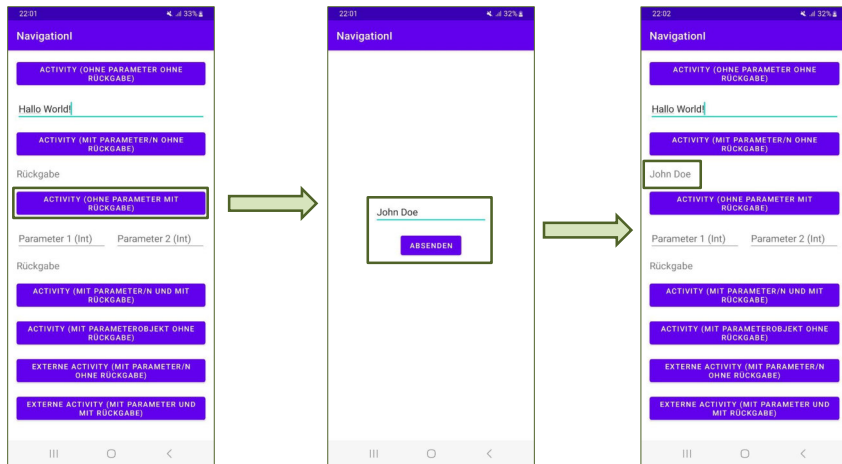
Aufruf einer Activity ohne Parameter mit Rückgabewert:

```
val intent = Intent(applicationContext ,  
    Activity_noParams_result :: class.java)  
(lauch_Activity_noparams_result as  
    ActivityResultLauncher<Intent >).launch(intent);
```

Hier wird der Intent über den `ActivityResultLauncher` gestartet und so nach dem Beenden der aufgerufenen Activity der Rückgabewert verarbeitet.

9.1.3 Intent (ohne Parameter/mit Rückgabe)

Beispiel (Aufruf einer Activity ohne Parameter mit Rückgabewert):



9.1.4 Intent (mit Parameter/mit Rückgabe)

Aufruf einer Activity mit Parameter mit Rückgabewert:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
    launch_Activity_params_result = registerForActivityResult(  
        ActivityResultContracts.StartActivityForResult(),  
        object : ActivityResultCallback<ActivityResult> {  
            override fun onActivityResult(result: ActivityResult?) {  
                if (result!!.resultCode == Activity.RESULT_OK) {  
                    val data : Intent? = result.data  
                    // Data-type specific i. e. getCharSequenceExtra()  
                    val result : String = data?.getIntExtra(  
                        Activity_params_result.RETURN_VALUE,0).toString()  
                    val textView = findViewById<TextView>(  
                        R.id.textViewReturn12)  
                    textView.text = result  
                }  
            }  
        }) ...  
}
```

In der Activity (z.B. MainActivity.kt) wird zunächst ein `ActivityResultLauncher` erstellt, an den später der Aufruf (Intent) gebunden wird.

9.1.4 Intent (mit Parameter/mit Rückgabe)

Aufruf einer Activity mit Parameter mit Rückgabewert (inkl. Argumentvalidierung):

```
...
val intent = Intent(applicationContext ,
    Activity_params_result :: class.java)
try {
    if (editTextParameter21.text.isNotEmpty() &&
        editTextParameter22.text.isNotEmpty()) {
        val parameter1 : Int =
            editTextParameter21.text.toString().toInt()
        intent.putExtra(R.id.editTextParameter21.toString(), parameter1)
        ...
        (lauch_Activity_params_result as
            ActivityResultLauncher<Intent>).launch(intent);
    } else {
        displayMessage(baseContext, "Parameter 1 or 2 is not given")
    }
} catch (e : java.lang.NumberFormatException) {...}
```

Hier wird der Intent über den ActivityResultLauncher gestartet und so nach dem Beenden der aufgerufenen Activity der Rückgabewert verarbeitet.

9.1.4 Intent (mit Parameter/mit Rückgabe)

Aufruf einer Activity mit Parameter mit Rückgabewert:

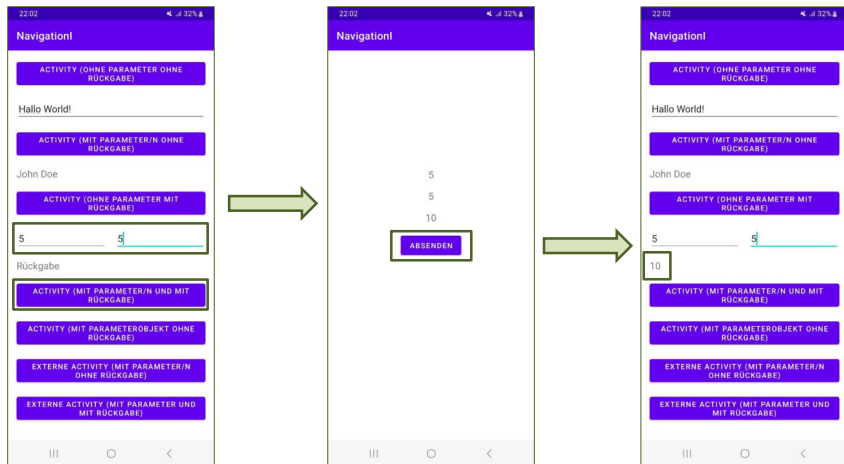
```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
    if (intent.hasExtra(R.id.editTextParameter21.toString())) {  
        parameter1 = intent.getIntExtra(  
            R.id.editTextParameter21.toString(), 0)  
        textView3.setText(parameter1.toString())  
    }  
    ...  
}
```

In der Klasse `Activity_params_result.kt` müssen die Parameterwerte noch entgegengenommen werden.

Hier wird mit `hasExtra()` geprüft ob ein entsprechender Wert vorliegt. Falls dies der Fall ist wird die Methode `getIntExtra()` zum Auslesen genutzt.

9.1.4 Intent (mit Parameter/mit Rückgabe)

Beispiel (Aufruf einer Activity mit Parameter mit Rückgabewert):



9.1.5 Intent (mit Parameterobjekt/ohne Rückgabe)

Aufruf einer Activity mit Parameterobjekt ohne Rückgabewert:

Definition eines Objekts vom Typ Person.

```
data class Person(  
    val vorname: String?,  
    val nachname: String?) {  
}
```


9.1.5 Intent (mit Parameterobjekt/ohne Rückgabe)

Die Klasse der übergebenen bzw. erhaltenen Objekte muss das Interface `Parcelable` implementieren.

Die Methode `writeToParcel` schreibt dann alle Attributwerte in das `Parcel parcel`.

```
data class Person(val vorname: String?,
    val nachname: String?) : Parcelable {

    constructor(parcel: Parcel) : this (parcel.readString(),
        parcel.readString()) {}

    override fun writeToParcel(parcel: Parcel, flags: Int) {
        parcel.writeString(vorname)
        parcel.writeString(nachname)
    }

    ...
}
```

9.1.5 Intent (mit Parameterobjekt/ohne Rückgabe)

Die Methode `describeContent` kann anzeigen, ob in dem Parcel weitere Metadaten (z.B. ein File-Descriptor) enthalten sind.

```
...  
  
override fun describeContents(): Int {  
    return 0  
}  
  
companion object CREATOR : Parcelable.Creator<Person> {  
    override fun createFromParcel(parcel: Parcel): Person {  
        return Person(parcel)  
    }  
    override fun newArray(size: Int): Array<Person?> {  
        return arrayOfNulls(size)  
    }  
}  
}
```

9.1.5 Intent (mit Parameterobjekt/ohne Rückgabe)

Aufruf einer Activity mit Parameterobjekt ohne Rückgabewert:

Definition eines Objekts vom Typ Person.

```
val person : Person = Person ("John", "Doe")  
val intent = Intent(applicationContext ,  
    Activity_paramObject::class.java)  
intent.putExtra("object", person)  
startActivity(intent)
```

9.1.5 Intent (mit Parameterobjekt/ohne Rückgabe)

Aufruf einer Activity mit Parameterobjekt ohne Rückgabewert:

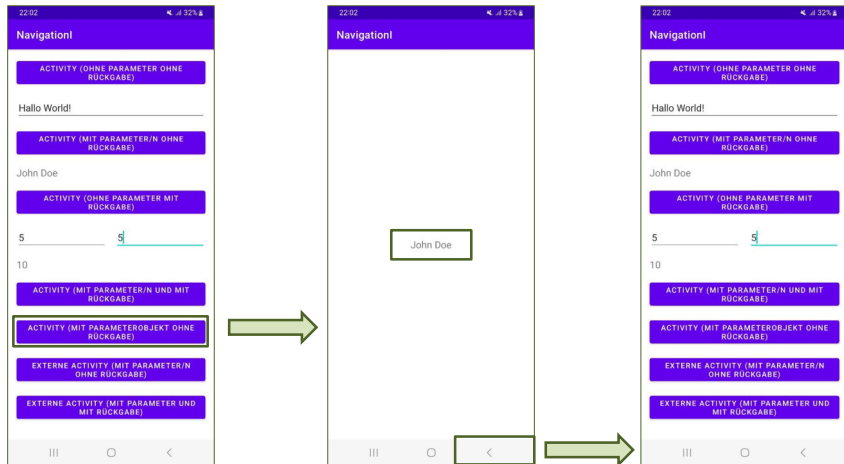
```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
    if (intent.hasExtra("object")) {  
        val person_object : Person = intent.getParcelableExtra(  
            "object", Person::class.java)!!  
        val textView = findViewById<TextView>(R.id.textView6)  
        textView.setText(  
            person_object.vorname + " " + person_object.nachname)  
        }  
        ...  
    }
```

In der Klasse `Activity_paramObject.kt` muss der Parameterwert noch entgegengenommen werden.

Hier wird mit `hasExtra()` geprüft ob ein entsprechender Wert vorliegt. Falls dies der Fall ist wird die Methode `getParcelableExtra()` zum Auslesen genutzt.

9.1.5 Intent (mit Parameterobjekt/ohne Rückgabe)

Aufruf einer Activity mit Parameterobjekt ohne Rückgabewert:



9.2 Interaktion (Inter-App)

Zwischen Activities unterschiedlicher mobiler Anwendungen kann auch eine Interaktion erfolgen:

- Aufruf einer Activity mit Parameter ohne Rückgabewert
- Aufruf einer Activity mit Parameter mit Rückgabewert

Für alle Aufrufe werden Intents (dt. Absichten) genutzt.

9.2.1 Ext. Intent (mit Parameter/ohne Rückgabe)

Aufruf einer externen Activity mit Parameter ohne Rückgabewert:

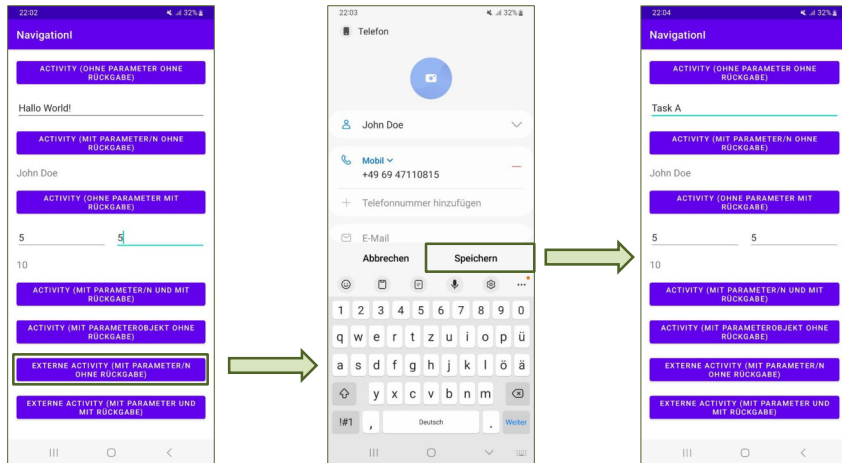
```
val intent = Intent(ContactsContract.Intents.Insert.ACTION)
intent.type = ContactsContract.RawContacts.CONTENT_TYPE
intent.putExtra(ContactsContract.Intents.Insert.NAME,
    "John Doe")
intent.putExtra(ContactsContract.Intents.Insert.PHONE,
    "+49 69 4711 0815")
startActivity(intent)
```

Gibt an, dass aus einer Activity (z.B. MainActivity.kt) die Activity `ContactsContract.Intents.Insert.ACTION` (Insert) aufgerufen werden soll.

Der eigentliche Aufruf wird dann mit `startActivity(intent)` ausgelöst.

9.2.1 Ext. Intent (mit Parameter/ohne Rückgabe)

Aufruf einer externen Activity mit Parameter ohne Rückgabewert:



9.2.2 Ext. Intent (mit Parameter/mit Rückgabe)

Aufruf einer externen Activity mit Parameter mit Rückgabewert:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
    launch_Activity_params_resultObject = registerForActivityResult(  
        ActivityResultContracts.StartActivityForResult(),  
        object : ActivityResultCallback<ActivityResult> {  
            override fun onActivityResult(result: ActivityResult?) {  
                val intent = Intent(applicationContext ,  
                    DialogActivity::class.java)  
                intent.putExtra("path", photoFile!!.absolutePath)  
                startActivity(intent)  
            }  
        }) ...  
}
```

In der Activity (z.B. MainActivity.kt) wird zunächst ein `ActivityResultLauncher` erstellt, an den später der Aufruf (Intent) gebunden wird.

9.2.2 Ext. Intent (mit Parameter/mit Rückgabe)

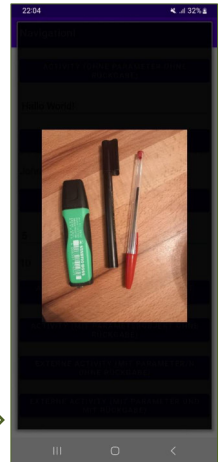
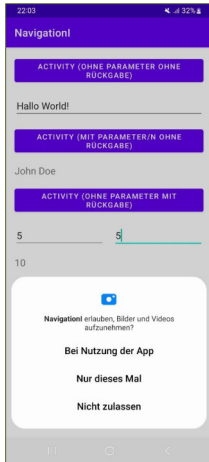
Aufruf einer externen Activity mit Parameter mit Rückgabewert:

```
val photoURI = FileProvider.getUriForFile(  
    this, "com.example.navigation1.fileprovider", photoFile!!)  
takePictureIntent.putExtra(  
    MediaStore.EXTRA_OUTPUT, photoURI)  
(launch_Activity_params_resultObject as  
    ActivityResultLauncher<Intent>).launch(takePictureIntent);
```

Hier wird der Intent über den `ActivityResultLauncher` gestartet und so nach dem Beenden der aufgerufenen externen Activity der Rückgabewert verarbeitet.

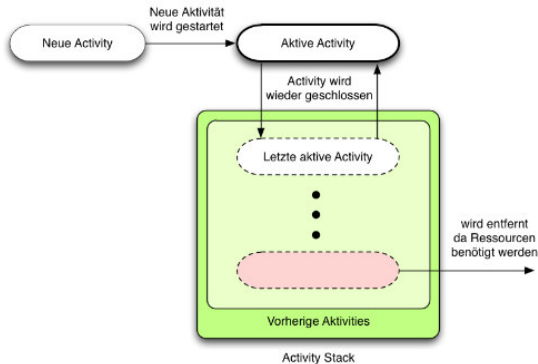
9.2.2 Ext. Intent (mit Parameter/mit Rückgabe)

Aufruf einer externen Activity mit Parameter mit Rückgabewert:



9.3 Activity Stack

Der Aufruf von Activities bewirkt, dass sich ein Activity Stack auf- u. abbaut:



Grafik: Markku D. Lammerz (2010)

Achtung: Das System kann auch Activities entfernen, wenn die Ressourcen knapp werden. Das kann unerwünschte Folgen haben.

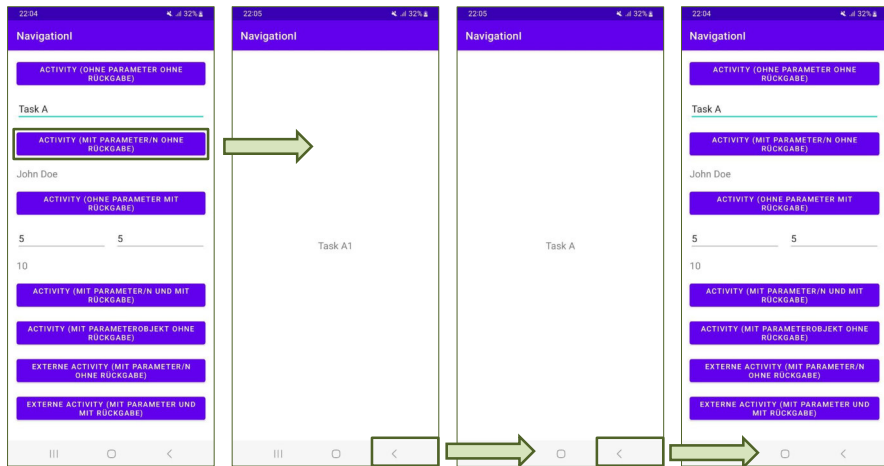
9.3.1 Activity Stack (Standardverhalten)

Der Aufruf der Activity `Activity_params_noResult.kt` mit dem Parameterwert `“Task A”` leitet einen (einfachen) rekursiven Aufruf derselben Activity ein.

```
if (textView.text.toString().equals("Task A")) {  
    val intent = Intent(applicationContext,  
        Activity_params_noResult::class.java)  
    intent.putExtra(R.id.editTextParameter11.toString(),  
        "Task A1");  
    // MainActivity -> A -> A1  
    startActivity(intent)  
}
```

9.3.1 Activity Stack (Standardverhalten)

Rekursiver Aufruf einer Activity (Standardverhalten):



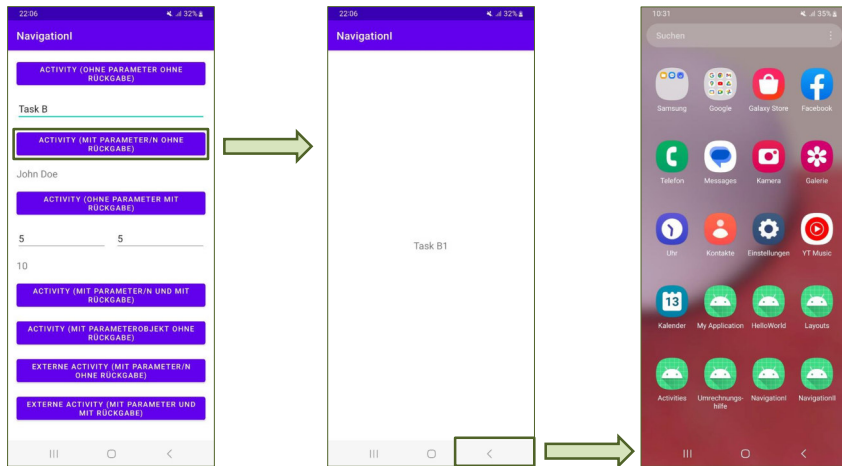
9.3.2 Activity Stack (New Task/Clear Task)

Der Aufruf der Activity `Activity_params_noResult.kt` mit dem Parameterwert `‘Task B’` leitet einen (einfachen) rekursiven Aufruf derselben Activity ein. Dabei wird die Activity mit den Optionen `NEW_TASK` u. `CLEAR_TASK` gestartet.

```
if (textView.text.toString().equals(" Task B")) {  
    val intent = Intent(applicationContext ,  
        Activity_params_noResult:: class .java)  
    intent.putExtra(R.id.editTextParameter11.toString(),  
        "Task B1");  
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK  
        or Intent.FLAG_ACTIVITY_CLEAR_TASK)  
    // B1  
    startActivity(intent)  
}
```

9.3.2 Activity Stack (New Task/Clear Task)

Rekursiver Aufruf einer Activity (New Task/Clear Task):



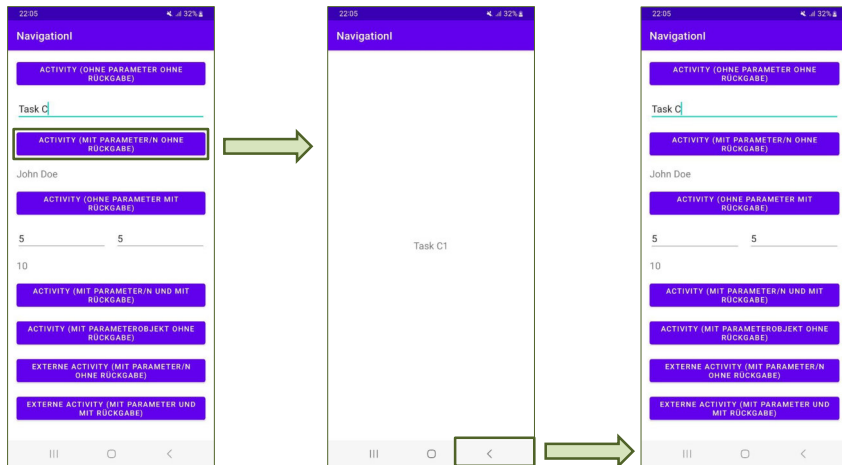
9.3.3 Activity Stack (Clear Top)

Der Aufruf der Activity `Activity_params_noResult.kt` mit dem Parameterwert `‘Task C’` leitet einen (einfachen) rekursiven Aufruf derselben Activity ein. Dabei wird die Activity mit der Option `CLEAR_TOP` gestartet.

```
if (textView.text.toString().equals("Task C")) {  
    val intent = Intent(applicationContext ,  
        Activity_params_noResult::class.java)  
    intent.putExtra(R.id.editTextParameter1.toString(),  
        "Task C1");  
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)  
    // MainActivity -> C1  
    startActivity(intent)  
}
```

9.3.3 Activity Stack (Clear Top)

Rekursiver Aufruf einer Activity (Clear Top):



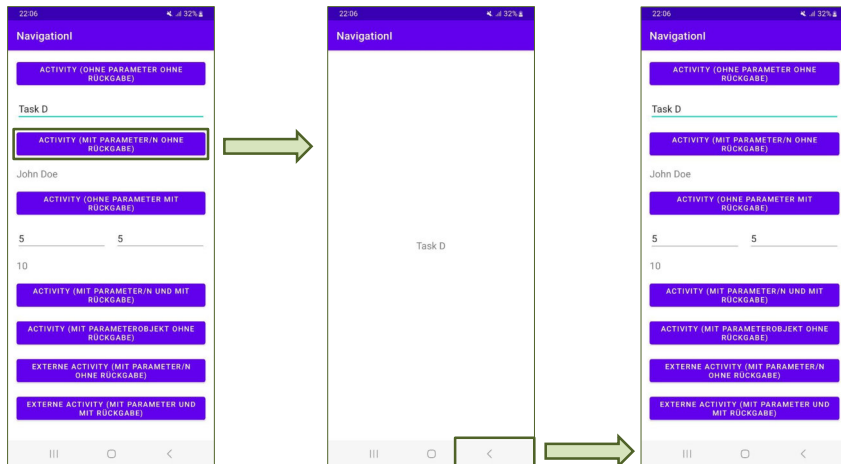
9.3.4 Activity Stack (Single Top)

Der Aufruf der Activity `Activity_params_noResult.kt` mit dem Parameterwert `‘Task D’` leitet einen (einfachen) rekursiven Aufruf derselben Activity ein. Dabei wird die Activity mit der Option `SINGLE_TOP` gestartet.

```
if (textView.text.toString().equals("Task D")) {  
    val intent = Intent(applicationContext ,  
        Activity_params_noResult::class.java)  
    intent.putExtra(R.id.editTextParameter1.toString(),  
        "Task D1");  
    intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP)  
    // MainActivity -> D  
    startActivity(intent)  
}
```

9.3.4 Activity Stack (Single Top)

Rekursiver Aufruf einer Activity (Single Top):



Block 9 – Zusammenfassung

- Activities können über `Intents` andere (oder dieselbe) Activity aufrufen.
- Dabei können sowohl Argumente (Parameter) an die aufgerufene Activity übergeben werden, als auch Ergebnisse nach dem Aufruf zurückerhalten werden.
- Argumente (Parameter) sind meist Basisdatentypen, können aber auch komplexe Objekte sein (`Parcelable`).
- Activities können mit unterschiedlichen Optionen aufgerufen werden, die sich auf den Aufbau des Activity-Stacks auswirken.
- Eine interne Activity (einer selbsterstellten mobilen Anwendung) kann auch externe Activities (z. B. Kamera-App) aufrufen und deren Funktionen nutzen.

Block 9 – Weitere Aufgaben

- Beschreiben Sie anhand der Beispiele zum Stackauf- und -abbau, welche Zustände der einzelnen Activities durchlaufen werden.
- Erzeugen Sie programmatisch (z. B. durch eine Schleife) eine hohe Anzahl von Intents einer neuen Activity. Beobachten Sie das Verhalten des Systems.
- Versuchen Sie durch die Umprogrammierung (Ergänzung) einer Activity, einen Aufruf einer zweiten Activity zu vermeiden.

Block 9 – Literatur I