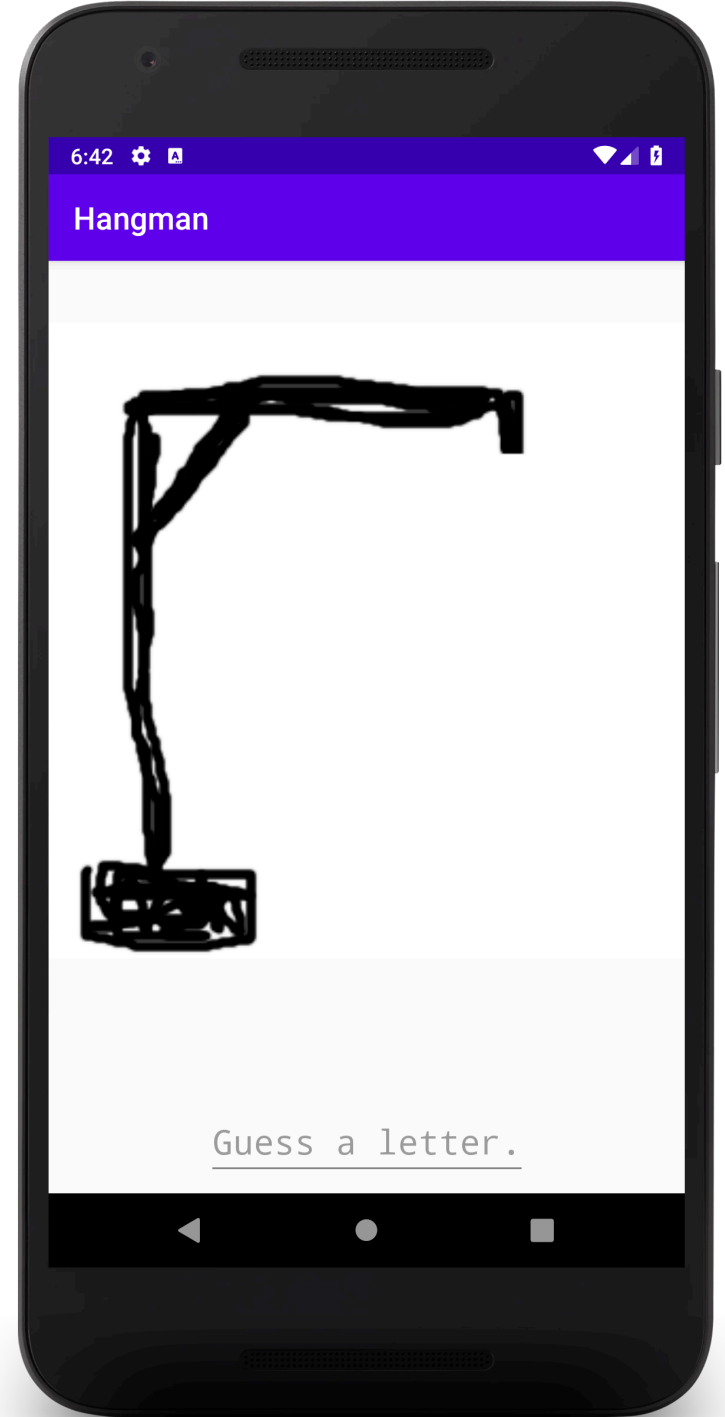


Hangman

Aplikácia – Obesenec



Logovanie

- <https://developer.android.com/studio/debug/am-logcat>
- Logcat (okno v Android Studio) – zobrazuje logy
- verbose, debug, info, warn, error, assert
- **Log.d(tag, message)**
- Tipy:
 - **if (BuildConfig.DEBUG)** – debug verzia aplikácie
 - **getClass().getName()** – názov triedy ako tag
 - filtrovanie správ v logcat
 - logcat screenshot nástroj

Životný cyklus aktivity

onCreate()

onStart()

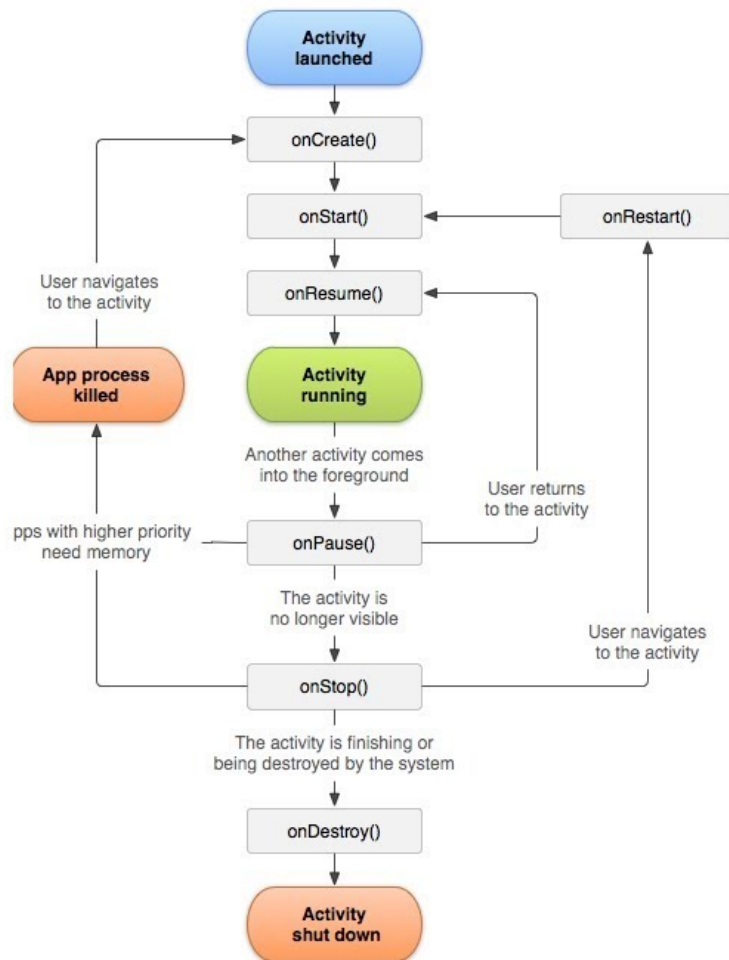
onRestart()

onResume()

onPause()

onStop()

onDestroy()



Životný cyklus aktivity - stav

- **viditeľná** - proces v popredí, má *focus*
- **čiasťočne viditeľná** - proces v pozadí, ak aktivita stratila *focus* napr. pri prekrytí lišty s prichádzajúcim hovorom alebo pri multiview zobrazení (focus má iná aktivita, ale daná pozastavená aktivita je stále viditeľná). Po zavolaní metódy **onPause()**
- **skrytá** - proces v pozadí, nie je viditeľná. Po zavolaní metódy **onStop()**
- **zničená** - proces je prázdny. Po zavolaní metódy **onDestroyed()**

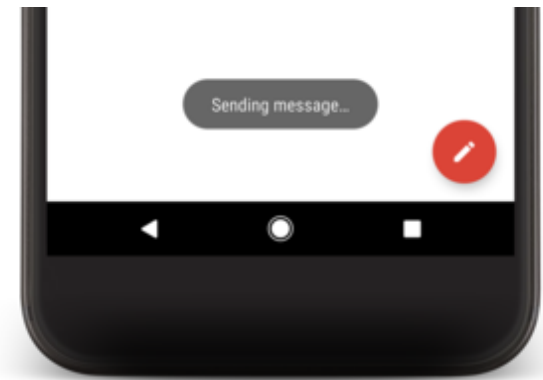
Android Jetpack



Jetpack is a **suite of libraries** to help developers follow best practices, reduce boilerplate code, and write code that works consistently across Android versions and devices so that developers can focus on the code they care about.

- dependency - **build.gradle (Module:app)**
- **ConstraintLayout**
<https://developer.android.com/training/constraint-layout/index.html>

Toast



- ostáva viditeľné aj keď aktivita skončí
- **`Toast.makeText(context, text, duration).show();`**
- **context** = aktivita (nie vždy, ale pre nás zatiaľ áno)
- **duration** = **`Toast.LENGTH_SHORT`** (alebo **`_LONG`**)
- `setGravity`, vlastný `view`...
- alternatíva – *snackbar*



Bundle

- reštart aktivity - otočenie obrazovky a iné
<https://developer.android.com/guide/topics/manifest/activity-element.html#config>
- mapa, key=**String**, value=**Parcelable**
- **save** - **onSaveInstanceState()**
 - parameter Bundle
 - volá sa pred zaniknutím aktivity – pred onStop()
 - špeciálne prípady v dokumentácii (prekrytie inou akt.)
- **load** – **onCreate()**
 - cez parameter Bundle (môže byť **null**)

Rozhrania na zabalenie objektov

- **Serializable**

- jednoduchá implementácia (žiadne metódy)
- object -> byte stream
- všetky inštančné premenné musia byť Serializable

- **Parcelable**

- rýchlejšie ako Serializable (nevytvára temp objekty)
- object -> byte stream
- vyžaduje implementáciu

Ako merať čas?



- **System.currentTimeMillis**
 - čas od 1.1.1970
 - nastaviteľný používateľom/sieťou – nevhodný na meranie uplynutého času
- **SystemClock.uptimeMillis** (System.nanoTime)
 - čas od bootovania systému
- **SystemClock.elapsedRealtime**
 - rovnako ako uptimeMillis + deep sleep

Kde uložiť dáta?

- súbor vs. databáza
- **app-specific** - internal/external storage
- **shared storage** – prístup aj z iných aplikácií
- **preferences** – key-value
- **database** - Android Room (súčasť Jetpack), SQLite
lokálna databáza

Shared Preferences

- mapa – klúč & hodnota
- **getPreferences ()** – ak stačí 1 súbor pre aktivitu
 - **getSharedPreferences ()** – súbory s menom
- READ – getX() + default hodnota
- WRITE – Editor + putX() + commit/apply
 - editor zapisuje po dávkach
 - **apply** - v pamäti ihneď, na disku asynchrónne
 - **commit** – synchrónny zápis (nevhodné do UI vlákna)