



# NTI/PAA - PROGRAMOVÁNÍ MOBILNÍCH APLIKACÍ

1. Platforma Android, prostředky, vývoj, oprávnění

Ing. Igor Kopetschke – TUL, NTI

<http://www.nti.tul.cz>

# Android – co to vlastně je

- Operační systém pro mobilní zařízení
- Založen na jádře Linuxu + úpravy respektující odlišnosti od klasických PC
- Výdrž baterie, menší výkonnost, paměť, rozlišení atd.
- Platforma Android poskytuje nejen OS, ale i
  - Uživatelské prostředí pro koncové uživatele
  - Specifikace ovladačů pro výrobce HW
  - Nástroje pro vývoj (SDK)
- Z převážné části otevřený systém, některé části uzavřené od Google

# Android – trochu historie

- 2003 – Android Inc. – Andy Rubin, Nick Sears, Chris White, Rich Miner
- 2005 – jako startup odkoupen Googlem, ve vedení Andy Rubin, Android na linuxovém jádře
- 2007 – konsorcium Open Handset Alliance
  - Google + cca 87 technologických a mobilních společností
  - Otevřený standard pro mobilní zařízení
  - Různý chipset, velikost a rozlišení obrazovky ...
  - Oficiální první Android na jádře Linux 2.6
- 2008 – první komerční telefon (HTC Dream) + SDK

# Android – trochu historie (verze)

- Verze 1.x
  - 1.5 (Cupcake), 1.6 (Donut)
- Verze 2.x
  - 2.0/2.1 (Eclair), 2.2 (Froyo), 2.3/2.4 (Gingerbread)
- Verze 3.x – určeno pro tablety (velké obrazovky)
  - 3.0/3.1/3.2 (Honeycomb)
- Verze 4.x – telefony i tablety
  - 4.0 (Ice Cream Sandwich), 4.1-4.3 (Jelly Bean), 4.4 (Kitkat)
- Verze 5.x (Lollipop)
  - 32 i 64 bit, nejpopulárnější v roce 2017
  - Podpora více SIM

# Android – trochu historie (verze)

## ■ Verze 6.x (Mashmallow)

- Změny řízení oprávnění aplikace (dodatečně požádat ...)
- Uživatelsky definované skupiny oprávnění

## ■ Verze 7.x (Nougat)

- Více oken – 2 aplikační okna na ½ displeje
- Nové 3D API Vulkan, nová podpora VR
- Uživatelské přidělení výkonu pro aplikace

## ■ Verze 8.0 (Oreo)

- Nové UI
- Android Go – light verze pro slabý HW
- Zvýšení bezpečnosti

# Android – trochu historie (verze)

## ■ Verze 9.x (Pie)

- Vylepšené Always-On, UI
- Wifi-RTT pro zjištění polohy
- atd.

## ■ Verze 10.x (původně Q – zrušeno užití sladkostí)

- Nativní podpora skládacích telefonů
- Novinky pro fotky a zvuk

## ■ Verze 11.x – 13.x

- Poslední verze vydána v srpnu 2022

# Android – architektura

- Architektura Androidu se skládá z 5 vrstev
  - Linux Kernel – jádro OS
  - Libraries – knihovny, napsány v C/C++
  - Android Runtime – aplikační virtuální stroj
  - Application Framework – aplikační vrstva
  - Applications – samotné uživatelské aplikace
- Toto rozdělení platí v podstatě napříč všemi verzemi Androidu
- Jednotlivé odlišnosti si představíme u popis vrstev

# Android – Linux Kernel

- Android < 4.1 – Linux 2.6, později 3.x, dnes 5.x
- HAL – Hardware Abstract Layer – vrstva mezi HW a SW kernelu nebo knihoven
- Správa paměti
- Ovladače zařízení
- Power management
- Správa procesů
- Naopak chybí
  - X Windows System
  - Neúplná sada GNU knihoven



# Android – knihovny

- Sada knihoven (C/C++)
- Využívány komponentami systému
- Poskytují funkce pro **Android Application Framework**
  - důležité pro vývojáře
    - Systémová knihovna C (libc) – upravená pro Android
    - Media Libraries (multimédia)
    - LibWebCore – webový prohlížeč
    - SQLite – relační DB knihovna
    - FreeType – rendering fontů
    - 3D, SGL (2D), WebKit, OpenSSL, OpenGL, aj.
    - Vulkan API – náhrada za Open GL

# Android – Android Runtime

- Aplikační virtuální stroj (analogie s JVM)
- Původně Dalvik Virtual Machine (DVM)
- .java -> .class -> Dalvik code -> DVM
- Z důvodů compatibility stále přítomen Dalvik Turbo
  - Vznik kvůli licencování JVM od Oracle
  - Samotná Java je volně šiřitelná
  - Každá aplikace má vlastní proces a vlastní instanci DVM

# Android – Android Runtime

- Od verze 4.4 nový Android RunTime (ART)
  - Používá dopřednou kompilaci (AOT, Ahead-of-time)
  - .java -> .class (Java kompilace)
  - .class -> .dex (byte code Dalviku)
  - .dex -> při instalaci kompilován do nativního kódu procesoru, DVM není potřeba pro spouštění
  - Mnohem rychlejší aplikace, delší výdrž baterie
  - Instalace je díky kompilaci pomalejší, ale je to jednorázová záležitost



# Android – Application Framework

- Nejdůležitější vrstva z pohledu vývojáře
- Poskytuje služby a funkce knihoven a nižších vrstev
- Základní sada služeb obsahuje např.
  - Sada View prvků – UI aplikací (button, checkbox, list ..)
  - Content providers – sdílení dat mezi aplikacemi
  - Resource manager – zdroje jako lokalizace, grafika, design
  - Notification manager – zasílání notifikací do stavového řádku
  - Activity manager – životní cyklus aplikací, zásobník aplikací



# Android – Applications

- Nejdůležitější vrstva z pohledu uživatele
- Tvoří ji samotné aplikace nainstalované v zařízení
- Jejich zdroje:
  - Předinstalováno systémem
  - Předinstalováno poskytovatelem (bundled SW)
  - Google play – oficiální market aplikací
  - 3. strana – od kamaráda, z webu atd ..
  - Vlastní vývoj

# Android – všech 5 vrstev



# Android – nástroje pro vývoj

- Základ tvoří Android SDK pro jednotlivé verze
- SDK Tools
  - Debugging (ddms) + testování
  - Android Virtual Devices (AVD) + AVD manager
  - Android emulátor pro vývoj na PC (SDK + AVD)
  - USB drivers pro testování na zařízení
  - Příklady kódů + dokumentace
  - Doplnky 3. stran (Google API, Market Licencing package..)
  - Drivery pro virtualizaci na Intel atd..
- IDE – donedávna Eclipse + pluginy, nyní Android Studio

# Android – nástroje pro vývoj

- Květen 2017, konference Google I/O 2017
  - Další oficiální jazyk pro vývoj aplikací pro Android
  - **Kotlin od JetBrains**
  - Jedná se o OpenSource pragmatický programovací jazyk
  - Někdy je nazýván "Swiftem pro Android"
  - Inteoperabilita mezi Kotlinem, Javou a JavaScriptem
  - Nativně podporován od Android Studio 3.0
  - Pro starší verze již existuje plugin



# Android – základní komponenty aplikace

- Aplikace pro Android se skládá z:
  - **Activity**
  - Fragments
  - Views
  - **Service**
  - **Content Provider**
  - Intents
  - **Broadcast Receiver**
  - Widgets
  - Resources ( layout, menu, styly, stringy, values..)
  - **AndroidManifest.xml**

# Android – Activity

- Základní komponenta pro zobrazení aplikace
- Bez existující aktivity nelze spustit aplikaci !
- Aplikace může mít i několik aktivit
- Prezentační vrstva aplikace
- Analogie s MVC/P – Android ale nemá View
- Data zobrazená z nižších vrstev je prezentována uživateli
- Životní cyklus je řízen pomocí **ActivityManager**, který pracuje se zásobníkem a zodpovídá za konkrétní stav aktivit
- Je vždy potomkem třídy `android.app.Activity`

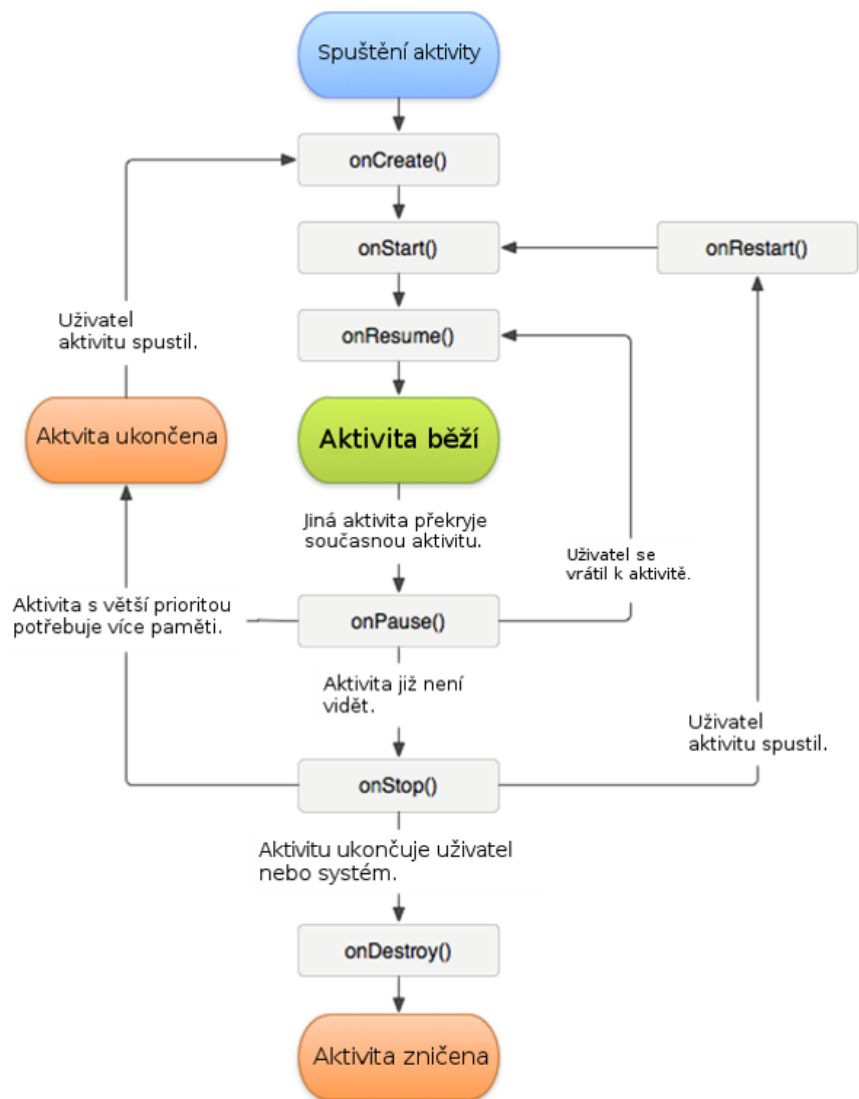
# Android – životní cyklus aktivit

- Aktivita má 4 fáze životního cyklu, v každém cyklu se volají metody zděděné z třídy **Activity**

## Jednotlivé fáze životního cyklu

- **Aktivita spuštěna** – došlo ke spuštění aplikace
- **Aktivita běží** – aktivita spuštěna a je v popředí
- **Aktivita v pozadí** – je vidět, ale překryta jinou aplikací (příchozí SMS, hovor, jiná notifikace ..)
- **Aktivita zastavená** – není vidět, bez přístupu, ale není úplně zničena
- **Aktivita ukončená** – úplné ukončení aktivity

# Android – životní cyklus aktivit





# Android – Manifest

- U Javy se spouštění a konfigurace aplikace řídí manifestem. U Androidu je to podobné.
- Každá aplikace má **AndroidManifest.xml**
  - Informace o package a verzi aplikace
  - Název, ikona, theme aplikace
  - Komponenty aplikace (Activity, Service aj.)
  - Permissions (práva)
  - Minimální a maximální Android API level
  - Použité knihovny
  - Další konfigurace

# Android – Manifest a oprávnění

- Manifest mj. řeší i oprávnění aplikace a jednotlivých komponent
- U Android verzích se při instalaci / spuštění zkontroloval úplný definovaný strom požadovaných oprávnění.
- Pokud některé z oprávnění nebylo povoleno, aplikace se nenainstaluje / nespustí
- Verze 6 má systém oprávnění přepracován.
- Aplikace může dodatečně požádat o oprávnění
- Aplikaci může být oprávnění dodatečně odmítnuto
- Řeší se tak situace, že aplikace získávala zbytečně příliš rozsáhlá práva
- <http://developer.android.com/reference/android/Manifest.permission.html>



.. A to je pro dnešek vše

DĚKUJI ZA POZORNOST