

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**VOLNĚ DOSTUPNÉ NÁSTROJE PRO VÝVOJ ANDROID
APLIKACÍ**
BAKALÁŘSKÁ PRÁCE

Miroslav Soukup
Informatika se zaměřením na vzdělávání

Vedoucí práce: Mgr. Denis Mainz
Plzeň, 2016

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 15. dubna 2016

.....
vlastnoruční podpis

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce, kterým byl Mgr. Denis Mainz. A to za vedení při tvorbě práce a užitečné rady, které mi pomohli práci zkompletovat.

ZDE SE NACHÁZÍ ORIGINÁL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

SEZNAM ZKRATEK	3
ÚVOD	4
1 SYSTÉM ANDROID	5
1.1 CO JE TO ANDROID?	5
1.2 HISTORIE	5
1.3 HISTORIE VERZÍ SYSTÉMU ANDROID	6
1.3.1 Android 1.0	6
1.3.2 Android 1.1	7
1.3.3 Android 1.5 (Cupcake)	7
1.3.4 Android 1.6 (Donut)	7
1.3.5 Android 2.0 (Eclair)	8
1.3.6 Android 2.2 (Froyo)	8
1.3.7 Android 2.3 (Gingerbread)	9
1.3.8 Android 3.0 – 3.1 (Honeycomb)	9
1.3.9 Android 4.0 (Ice Cream Sandwich)	10
1.3.10 Android 4.1 – 4.3 (Jelly Bean)	11
1.3.11 Android 4.4 (KitKat)	12
1.3.12 Android 5.0 (Lollipop)	12
1.3.13 Android 6.0 (Marshmallow)	13
1.4 ANDROID AKTUÁLNĚ	13
2 VÝVOJ APLIKACÍ PRO ANDROID	16
2.1 JAZYKY PRO VÝVOJ APLIKACÍ	16
2.1.1 Java	16
2.1.2 XML	18
2.1.3 C/C++	18
2.1.4 Lua	19
2.1.5 HTML, CSS a JavaScript	19
2.1.6 Metoda „drag and drop“	20
2.2 VÝVOJOVÁ PROSTŘEDÍ	20
2.2.1 Android Studio	20
2.2.2 Stencyl	21
2.2.3 MIT App Inventor	21
2.2.4 Corona SDK	22
2.2.5 PhoneGap	22
2.3 KLASIFIKACE NÁSTROJŮ A MOŽNOSTÍ	22
2.4 KRITÉRIA PRO VÝBĚR VÝVOJOVÉHO PROSTŘEDÍ	24
3 ZÁKLADY ANDROID APLIKACE	26
3.1 TYPY APLIKACÍ	26
3.2 SLOŽENÍ APLIKACÍ	26
3.2.1 Aktivity (Activity)	26
3.2.2 Služby (Services)	27
3.2.3 Poskytovatelé obsahu (Content Providers)	27
3.2.4 Záměry (Intents)	27
3.2.5 Přijímače (Broadcast Receivers)	27
3.2.6 Oznámení (Notifications)	28
3.3 CO MÁ VÝVOJÁŘ PŘI VÝVOJI APLIKACE K DISPOZICI	30

3.3.1	Úložiště	30
3.3.2	Síť	30
3.3.3	Multimédia	30
3.3.4	Global Positioning systém (GPS).....	31
3.3.5	Telefonní služby	31
3.4	OPRÁVNĚNÍ PRO APLIKACE	31
3.5	ANDROID MANIFEST	33
4	PŘEDSTAVENÍ VÝVOJOVÉHO PROSTŘEDÍ ANDROID STUDIO	34
4.1	INSTALACE POTŘEBNÝCH NÁSTROJŮ	34
4.2	PRVNÍ SPUŠTĚNÍ	35
4.3	SDK MANAGER	37
4.3.1	SDK Platforms	38
4.3.2	SDK Tools	38
4.3.3	SDK Update Sites	39
4.4	VYTVOŘENÍ PRVNÍHO PROJEKTU	40
4.5	NASTAVENÍ EMULÁTORU A ZAŘÍZENÍ	46
4.6	APLIKACE.....	51
4.6.1	Představení aplikace.....	51
4.6.2	Aplikace zevnitř	53
	ZÁVĚR.....	59
	RESUMÉ	60
	SUMMARY	61
	SEZNAM LITERATURY	62
	SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ	65
	PŘÍLOHY	I
	PŘÍLOHA 1 – ZDROJOVÝ KÓD ACTIVITY_MAIN.XML	II
	PŘÍLOHA 2 – ZDROJOVÝ KÓD MAINACTIVITY.JAVA	III
	PŘÍLOHA 3 – CD.....	IV

SEZNAM ZKRATEK

JDK	Java Development Kit – Balíček nástrojů pro vývoj Java aplikací
SDK	Software Development Kit – Sada vývojových nástrojů
ADT	Android Developing Tools – doplňkový modul pro prostředí Eclipse umožňující vývoj aplikací pro platformu Android
XML	Extensible Markup Language – Rozšiřitelný značkový jazyk
HTML	HyperText Markup Language – Značkový jazyk pro tvorbu webových stránek
CSS	Kaskádové styly určující vzhled webových aplikací
WiFi	Technologie umožňující připojení zařízení k bezdrátové síti
Gmail	Emailová služba od společnosti Google
RAM	Operační paměť
API	Application Programming Interface – rozhraní pro programování aplikací

Úvod

Mobilní zařízení jako tablety či smartphony neboli chytré telefony, jsou fenoménem, bez něž si dnešní svět dokáže představit jen málokdo. Technologický vývoj takových zařízení je zdánlivě nezastavitelný, o čemž svědčí i fakt, že výkony dnešních mobilních telefonů by se daly srovnávat s výkony některých stolních počítačů. S tímto výkonem souvisí nejen hardwarové specifikace, ale i softwarové vybavení od operačního systému po další podpůrné programové aplikace.

Práce je zaměřena na softwarové vybavení mobilních zařízení, zejména pak chytrých telefonů, konkrétně na operační systém Android a na vývoj aplikací pro něj. Nejprve je představen systém Android z hlediska historie jednotlivých verzí včetně těch aktuálně nejpoužívanějších. Následně je tento systém porovnán s ostatními operačními systémy a to podle podílu mobilních platforem na trhu.

Další část je věnována teoretickému úvodu a základním znalostem, potřebným k vývoji aplikací pro tento operační systém. Představena budou také volně dostupná vývojová prostředí i jednotlivé jazyky, které jsou v těchto prostředích používány. Jedním segmentem této části je i určitá kategorizace představených vývojových prostředí a s tím související kritéria pro správný výběr jednoho z nich.

Následující kapitola je zaměřena na to, jak Android aplikace funguje a z jakých komponent se může skládat.

Poslední část se věnuje vývojovému prostředí Android Studio. Je vysvětleno nainstalování všech potřebných komponent, první spuštění, nastavení emulátoru. Práce v tomto prostředí je ukázána na vytvoření jednoduché aplikace, kde jsou nám přiblíženy i jazyky, které se zde používají.

Cílem práce je seznámit potenciálního vývojáře s možnostmi, které může zvolit pro vytváření aplikací pro systém Android a představením jednoho z nejpoužívanějších vývojových nástrojů demonstrovat, jakým směrem se může ubírat vývoj aplikací pro systém Android.

1 SYSTÉM ANDROID

1.1 CO JE TO ANDROID?

Android je operační systém vyvíjený zejména pro mobilní telefony a tablety. V dnešní době se ale tato softwarová platforma rozšiřuje do více a více zařízení jako např. televize nebo automobily.

Android je založený na Linuxovém jádře a je prezentován jako open-source software. To znamená, že by jeho zdrojový kód měl být volně k dispozici k dalšímu upravování a neomezené distribuci. Některé části jsou však uzavřené společností Google a některé patentované části systému Android si musí výrobci, kteří chtějí systém Android použít ve svých zařízeních, licencovat. (1)

Systém je dále navržený tak, aby mohl běžet nezávisle na hardwaru, např. různé rozlišení nebo velikost displejů.

1.2 HISTORIE

Společnost Android Inc. byla založena v Kalifornii roku 2003 Andym Rubinem, Richem Minerem, Nickem Searsem a Christopherem Whitem. (2) Původním záměrem této společnosti bylo vyvíjet pokročilý operační systém pro digitální kamery. Po čase ale zjistili, že trh s kamerami není dostatečně velký a odklonily směr vývoje operačního systému k mobilním telefonům. (3)

V roce 2005 byl Android Inc. odkoupen společností Google přibližně za 50 milionů dolarů. Klíčový zaměstnanci, včetně zakladatelů koupené společnosti, zde nadále zůstali. (4) Doposud se o společnosti Android Inc. moc nevědělo, ale předpokládalo se, že chce Google tímto tahem vstoupit na trh mobilních telefonů. (5) Následně tým pracovníků, s Andym Rubinem ve vedení, vyvinul platformu pro mobilní telefony založenou na Linuxovém jádře. Google platformu prezentoval výrobcům mobilních telefonů jako flexibilní a jako platformu, kterou lze snadno vylepšit. (6)

Spekulace okolo vstupu Googlu na mobilní trh se stále stupňovaly. Byl vyvinut prototyp, který ale nepodporoval dotykový displej, pouze fyzickou QUERTY klávesnici. Tento prototyp byl později předělán tak, aby dotykový displej podporoval a mohl konkurovat dalším společnostem, jejichž koncept dotykový displej obsahoval. (7)

Dne 5. listopadu 2007 bylo společností Google založeno konsorcium technologických firem Open Handset Alliance, pro vyvíjení otevřených standardů pro mobilní telefony. Součástí tohoto konsorcia jsou například Samsung, HTC, T-Mobile, Qualcomm a mnoho dalších. Tohoto dne byl představen systém Android. (5) První mobilní telefon, uvedený na veřejný trh dne 22. října 2008, bylo HTC Dream. (8)

Od roku 2008 Android získával velké množství aktualizací, které postupně vylepšovaly operační systém. Příkladem může být přidávání nových funkcí a opravování chyb verzí předchozích. Každá z majoritních aktualizací nese název po nějaké „cukrovince“ např. „Donut“.

Roku 2010 začala společnost Google vydávat telefony nesoucí název Nexus. Tato série zařízení je specifická tím, že obsahuje „čistý“ systém Android. To znamená, že i když je telefon vyráběn různými výrobci např. LG či HTC, neobsahuje žádnou jejich nadstavbu.

1.3 HISTORIE VERZÍ SYSTÉMU ANDROID

1.3.1 ANDROID 1.0

Android 1.0 je první verze tohoto operačního systému, uvedená na trh dne 23. září 2008. Běžela na telefonu HTC Dream a byla závislá na hardwaru tohoto telefonu. (9)

Hlavní funkce (10):

- Podpora fotoaparátu.
- Stahování a aktualizace aplikací z Android Marketu.
- Webový prohlížeč, který dokázal pracovat s HTML a XHTML.
- Hlasové vytáčení.
- Youtube video přehrávač.
- Notifikace ve stavové liště.
- Podpora WiFi a Bluetooth.
- Multimediální přehrávač.
- Umísťování widgetů (miniaplikací) na hlavní obrazovku.
- Aplikace Gmail, synchronizovaná s Gmailem.

1.3.2 ANDROID 1.1

Aktualizace byla uvolněna 9. února 2009. Byly přidány nové funkce, ale především tato verze řešila nedostatky té předchozí. (10)

Hlavní změny (10):

- Detailnější informace o podnicích v mapách.
- Prodloužen interval zhasnutí obrazovky při příchozím hovoru a možnost zobrazení číselníku během hovoru.
- Možnost ukládání příloh ze zpráv.

1.3.3 ANDROID 1.5 (CUPCAKE)

Tato verze byla uvolněna dne 27. dubna 2009 a jako první byla oficiálně pojmenována po „cukrovince“ z čehož se stal trend a byla tak pojmenovaná každá následující velká aktualizace. Tato verze byla označena jako „Cupcake“ neboli „Košíček“. (10)

Hlavní změny (10):

- Virtuální klávesnice s podporou predikce textu a vlastního slovníku.
- Přiřazení obrázku k oblíbeným kontaktům.
- Kopírování a vkládání ve webovém prohlížeči.
- Animované přechody jednotlivých obrazovek.
- Nová animace při zavádění systému.
- Možnost nahrávat videa na YouTube.
- Automatické otáčení obrazovky vzhledem k natočení telefonu.
- Nahrávání a přehrávání videí ve formátu 3GP.

1.3.4 ANDROID 1.6 (DONUT)

Byl vydán 15. září 2009 a pojmenován „Donut“ neboli „Kobliha“. (10)

Hlavní změny (10):

- Podpora různých rozlišení a poměrů stran obrazovky.
- Vylepšené hlasové i textové vyhledávání.

- Snazší vyhledávání v Android Marketu a zobrazování screenshotů dané aplikace.
- Při mazání fotografií přidána možnost označit jich více najednou.

1.3.5 ANDROID 2.0 (ECLAIR)

Po necelých dvou měsících byla dne 26. října 2009 vydána další aktualizace nazvaná „Eclair“. (10)

Hlavní změny (10):

- Možnost přidat více Google účtů, pro synchronizaci emailů a kontaktů.
- Podpora Bluetooth 2.1.
- Vylepšení funkcí fotoaparátu jako například digitální zoom, podpora blesku nebo vyvážení bílé.
- Prohlížeč podporující HTML5 a přiblížení obrazovky po dvojitém poklepání.

1.3.6 ANDROID 2.2 (FROYO)

Další verze systému Android, „Froyo“, byla vydána dne 20. května 2010. Většina změn u této verze byla provedena na pozadí a jejich cílem bylo zrychlení a zjednodušení systému. Společně s touto verzí byl vydán i první telefon z řady Nexus. Telefony z této řady jsou dodávány s Androidem bez jakýchkoliv nadstaveb a jsou uváděny na trh přímo společnostmi Google. Jednotlivá zařízení jsou však vyráběna různými výrobci. První z těchto zařízení měla na starost firma HTC a neslo označení Nexus One. (10)

Hlavní změny (10):

- Optimalizace rychlosti, paměti a výkonu.
- Podpora sdílení internetového připojení přes USB a WiFi.
- Možnost zakázat připojení k internetu přes mobilní data.
- Podpora automatických aktualizací aplikací skrze Android Market.
- Rychlé přepínání mezi jazyky klávesnice.
- Podpora instalace aplikací na rozšiřující paměť.
- Vylepšená celková funkcionality webového prohlížeče.

- Podpora push notifikací (notifikace, posílané ze serveru, na kterém se ukládají data z aplikace).

1.3.7 ANDROID 2.3 (GINGERBREAD)

Verze „Gingerbread“ byla vydána dne 6. prosince 2010 společně s dalším zařízením z řady Nexus. Tentokrát byl tento telefon vyráběn společností Samsung a nesl označení Nexus S. (10)

Hlavní změny (10):

- Aktualizovaný vzhled uživatelského rozhraní se zvýšenou jednoduchostí a rychlostí.
- Rychlejší a více intuitivní psaní na klávesnici, se zvýšenou přesností, lepší předvídání následujícího textu.
- Podpora NFC (bezdrátová komunikace na krátké vzdálenosti).
- Podpora více fotoaparátů na jednom zařízení. Například přední a zadní fotoaparát.
- Vylepšené řízení spotřeby energie.
- Vylepšení zvuku, grafiky a vstupů pro vývojáře her.
- Garbage collector běžící na pozadí. Jeho úkolem je vyřazovat z paměti data, která aplikace již nepotřebuje.
- Nativní podpora senzorů jako jsou Gyroskop nebo Barometr.

1.3.8 ANDROID 3.0 – 3.1 (HONEYCOMB)

Verze „Honeycomb“ byla vydána dne 22. února 2011 a jako první verze systému Android, byla určena pouze pro tablety. (10)

Hlavní změny (10):

- Přidán systémový panel pro rychlý přístup k notifikacím a některým nastavením. Dále se zde nachází systémová tlačítka. Panel je umístěn na dolní části obrazovky.
- Přidán akční panel na horní část obrazovky, který umožňuje rychlý přístup ke kontextovým možnostem, widgetům a dalšímu nastavitelnému obsahu.

- Zjednodušení multitaskingu tzn., že po kliknutí na nabídku posledních použitých aplikací na systémovém panelu se zobrazí náhledy těchto aplikací. Uživatel mezi nimi může libovolně přecházet.
- Předělaná klávesnice pro snazší a rychlejší psaní.
- Zjednodušené rozhraní pro kopírování a vkládání.
- Práce v prohlížeči je možná ve více panelech. Prohlížeč nabízí procházení webu v anonymním režimu.
- Podpora procesorů s více jádry.
- Možnost šifrovat data uživatele.
- Možnost připojení externích USB zařízení.
- Možnost měnit velikost widgetů na domovské stránce.
- Podpora joysticků a gamepadů (ovladače pro ovládání her).

1.3.9 ANDROID 4.0 (ICE CREAM SANDWICH)

Android 4.0 („Ice Cream Sandwich“) byl vydán dne 19. října 2011. Google uvedl, že tato verze by měla být kompatibilní s kterýmkoliv zařízením, které bylo vyrobeno v době Androidu 2.3.x. (10)

Hlavní změny (10):

- Možnost použití softwarových tlačítek z předchozí verze i u telefonů (nahrazení tlačítek hardwarových).
- Velká změna vzhledu systému.
- Vylepšená hlasová schránka s možností zrychlit nebo zpomalit hlasovou zprávu.
- Integrovaná funkce zachycení snímku obrazovky za pomoci tlačítek hlasitosti a vypnutí/zapnutí telefonu.
- Možnost přístupu k aplikacím přímo ze zamčené obrazovky.
- Možnost odemkat telefon pomocí softwaru na rozpoznání obličeje.
- Automatická synchronizace záložek v prohlížeči Chrome na počítači a v telefonu.

- Přehled využití mobilních dat. Možnost nastavení limitu, kdy se mají data vypnout.
- Zabudovaný editor fotografií.
- Ukončování aplikací v nabídce posledních použitých aplikací pomocí přetažení dané aplikace do strany.

1.3.10 ANDROID 4.1 – 4.3 (JELLY BEAN)

Další verze – „Jelly Bean“ byla Vydána dne 9. července 2012 a jejím primárním záměrem bylo vylepšení funkčnosti a výkonu celého uživatelského rozhraní.

„Project Butter“ je pojem související se zlepšením výkonu. Využívá předvídání doteků trojitého načítání dat do mezipaměti pro vykreslení animace a dále stabilní obnovovací frekvenci displeje s hodnotou 60 snímků za sekundu.

Tyto aspekty byly využity k tomu, aby systém působil plynuleji. Android 4.2 byl vydán dne 13. listopadu 2012 a Android 4.3 dne 24. července 2013. (10)

Hlavní změny (10):

- Plynulejší prostředí díky projektu „Butter“.
- Rozšiřitelné notifikace.
- Možnost vypnutí notifikací u konkrétních aplikací.
- Přehrávání audia bez mezer mezi jednotlivými stopami.
- Podpora více uživatelských účtů na jednom zařízení (pouze u tabletů).
- Skupinové posílání zpráv.
- Doručky u prémiových SMS zpráv.
- Podpora využití Bluetooth jako dálkového ovládání.
- Vylepšení aplikace fotoaparátu.
- Widgety a zástupci na domovské obrazovce mohou být automaticky přesunuty nebo může být změněna jejich velikost tak, aby se na domovskou obrazovku vešly nové prvky.

1.3.11 ANDROID 4.4 (KITKAT)

Verze „Kitkat“ byla vydána dne 31. října 2013 a byla optimalizována tak, aby mohla fungovat na větším počtu zařízení než předchozí uvedené verze. Pro tuto aktualizaci byla doporučena minimální paměť RAM 512MB i přesto, že Android potřebuje ke svému běhu pouze 340MB. Zařízení, která nedisponovala doporučenou minimální pamětí, se musela prezentovat jako zařízení s malou pamětí RAM. (10)

Hlavní změny (10):

- Vylepšené grafické uživatelské rozhraní s bílými prvky namísto modrých.
- Aplikace mohou zasahovat do průhlednosti navigačního a stavového panelu. Mohou-též tyto panely úplně skrýt, pokud je aplikace na popředí.
- Aplikace mají zakázaný přístup k externím uložistům (např. paměťové kartě), pokud se nejedná o práci se složkou téže aplikace.
- Podpora bezdrátového tisku.
- V nastavení je nyní možné změnit výchozí aplikaci pro komunikaci prostřednictvím SMS.
- Podpora senzorů pro počítání kroků.
- Optimalizovaný výkon na zařízeních s horšími hardwarovými specifikacemi.
- Rozšířená funkčnost služeb pro odposlouchávání oznámení.

1.3.12 ANDROID 5.0 (LOLLIPOP)

Android 5.0 „Lolipop“ byl Vydán dne 12. listopadu 2014 a jeho opětovnou charakteristikou byl předělaný vzhled uživatelského rozhraní, nazvaný „Material design“. (10)

Hlavní změny (10):

- Vylepšený Garbage Collector.
- Podpora procesorů s 64 bitovou architekturou.
- Podpora náhledu tisku.
- Zrušena podpora widgetů na zamčené obrazovce.

- U notifikací je možnost nastavení jejich priority.
- V systémovém nastavení přibyla možnost vyhledávání konkrétní položky, pro rychlejší přístup k tomuto nastavení.
- Optimalizace systému tak, aby baterie měla delší životnost na jedno nabití.
- Na uzamčenou obrazovku mohou být umístěni zástupci aplikací.
- Obnoven přístup aplikací k externím uložištím.
- Aplikace v nabídce posledních spuštěných zde zůstávají i po restartu zařízení.
- Vytvoření více uživatelských účtů je nyní možné i v telefonu.

1.3.13 ANDROID 6.0 (MARSHMALLOW)

Prozatím poslední velká aktualizace operačního systému Android s názvem „Marshmallow“ byla vydána 15. Října 2015.

Následovala ještě jedna aktualizace, označovaná 6.0.1, ta ale jen opravila majoritní chyby původní verze a přidala velké množství emotikon neboli „smajlíků“. (10)

Hlavní změny (10):

- Snížení rychlosti práce procesoru pokud je vypnutá obrazovka a tím zvýšení životnosti baterie na jedno nabití. Tato funkce se nazývá „Doze“.
- Vyhledávací panel v menu aplikací.
- Nativní podpora čtečky otisků prstů.
- Podpora USB typu C.
- Automatické zálohování a obnovování dat aplikací.

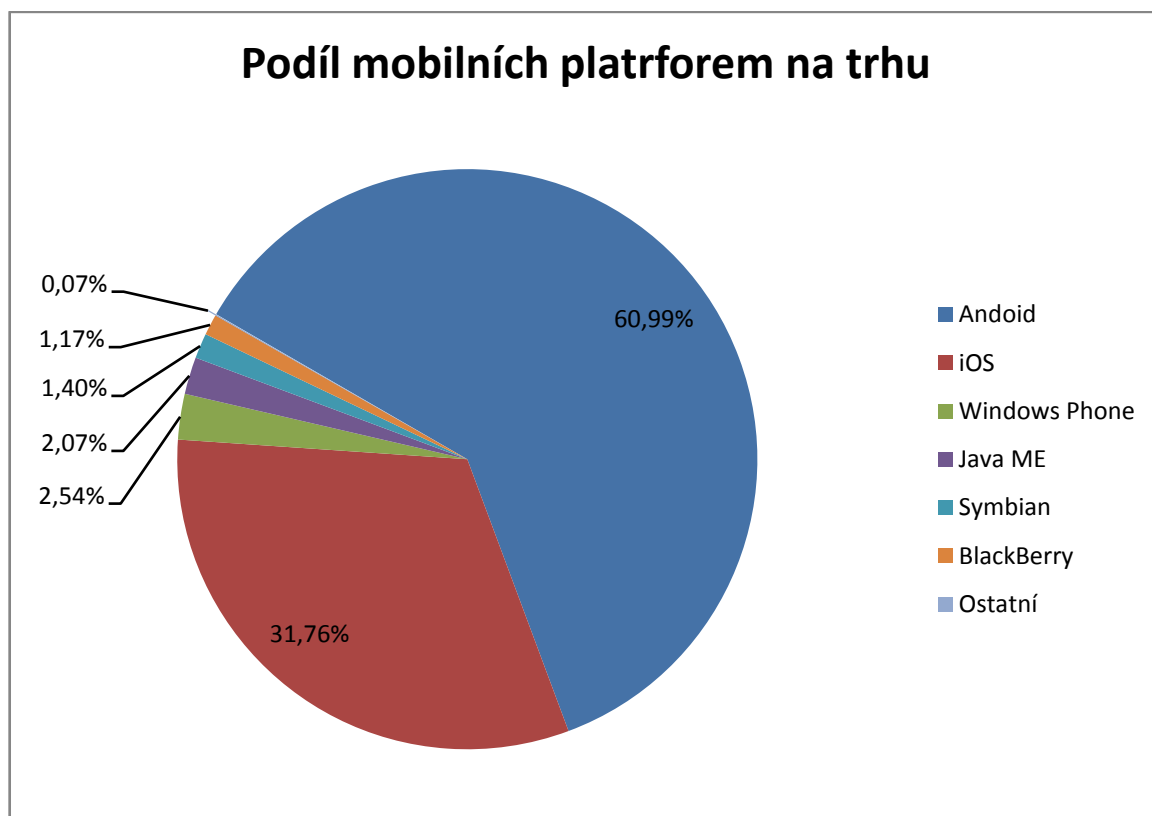
Možnost udělovat aplikacím jednotlivá oprávnění. K čemu mohou přistupovat (např. přístup ke kontaktům) a co mohou využívat (např. nákupy v aplikaci).

1.4 ANDROID AKTUÁLNĚ

V dnešní době je systém Android nejrozšířenější mobilní platformou na trhu. Údaje o tom, jaké má Android aktuální zastoupení na trhu, je možné zjistit ze stránky <http://marketshare.hitslink.com/>. Tato stránka zaznamenává statistiky například o tom, jaké operační systémy jsou nejvíce využívány, jak v mobilních zařízeních, tak i na osobních

počítačích v rámci zadaného časového období. Dále je zde možné zjistit, jaký internetový prohlížeč nebo jaký vyhledávač je nejvíce používán v rámci daných zařízení.

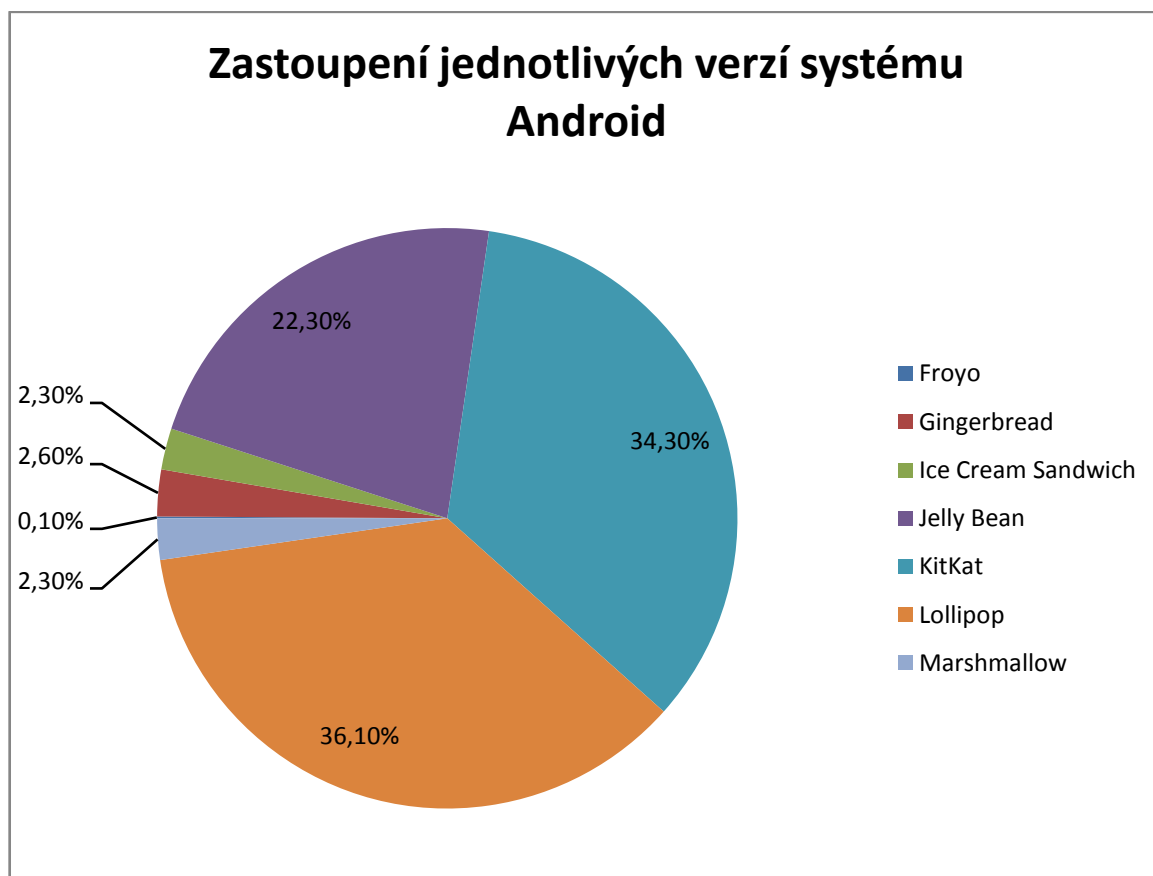
Následující graf (Graf 1) ukazuje zastoupení jednotlivých mobilních platform v měsíci březnu 2016. Jak je možné vidět, Android na trhu převládá. A jeho procentuální zastoupení dále roste. Na druhém místě je konkurenční platforma iOS. Procenta ostatních výrobců jsou na trhu mizivá. (11)



Graf 1: Podíl mobilních platform na trhu (zdroj: vlastní)

Výhodou, ale zároveň i nevýhodou operačního systému Android mohou být rychle vycházející aktualizace. Jak je patrné z grafu (Graf 2), verze, která byla vydána na podzim roku 2015, běží pouze na 2,3 % zařízení. Verze téměř rok a půl stará běží přibližně na třetině zařízení s tímto operačním systémem, což není mnoho. Největší problém v tomto ohledu je, že většina výrobců vytváří nadstavby systému. Když potom společnost Google vydá novou verzi, výrobci musí tuto nadstavbu opět předělat a většina těchto nadstavb se váže přímo ke konkrétnímu zařízení. Proto aktualizace mají tak dlouhou prodlevu a k zařízením z nižší cenové kategorie se většinou ani nedostanou. (12)

Řešením tohoto problému by mohla být distribuce systému pouze v čisté podobě, bez jakýchkoliv nadstaveb. To funguje u zařízení zmíněných výše, řady telefonů a tabletů Nexus. Výrobcům telefonů toto ale Google bohužel nařídít nemůže, protože je Android prezentován jako open-source software. Tento problém, ale nemění nic na tom, že systém Android je nejrozšířenější platformou a proto má programování aplikací pro něj velký potenciál.



Graf 2: Zastoupení jednotlivých verzí systému Android (zdroj: vlastní)

2 VÝVOJ APLIKACÍ PRO ANDROID

K vyvíjení aplikací je obvykle potřeba mít několik nástrojů a komponent (13 str. 15):

- Java SE Development Kit (JDK).
- Android Software Development Kit (SDK).
- Vývojové prostředí.

Java SE Development Kit je balíček nástrojů pro vývojáře, který umožňuje vytvářet a kompilovat Java aplikace. Balíček obsahuje též debugger a Java Runtime Environment (JRE), což je v podstatě virtuální stroj, který nám umožňuje Java aplikaci spustit. Tento balíček je dostupný z oficiálních stránek jazyka Java společnosti Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/>). (14)

Android Software Development Kit je sada nástrojů, která umožňuje vyvíjet aplikace pro platformu Android. Obsahuje knihovny, debugger, emulátor (pro spuštění vytvořené aplikace na virtuálním zařízení přímo v počítači), dokumentaci. S každou novou aktualizací systému Android vychází i nová verze Android SDK z důvodu toho, aby vývojář mohl při stavění nové aplikace využít všech funkcí, které aktuální verze systému nabízí. Android SDK je dostupná na stránkách pro vývojáře Android aplikací (<http://developer.android.com/tools/sdk/>). Stránky jsou produkovány společností Google. (15)

2.1 JAZYKY PRO VÝVOJ APLIKACÍ

2.1.1 JAVA

Jazyk Java byl vytvořen firmou Sun Microsystems. Původně byl určen pouze pro spotřební elektroniku. První verze tohoto jazyka byla vydána v roce 1995 a firmu v témže roce zároveň získala společnost Oracle, která Javu nadále spravuje. (13 str. 15)

Java je programovací jazyk objektově orientovaný. Jeho syntaxe je podobná jazykům C a C++, ale nesetkáváme se zde například s ukazatli. Zde se o správu paměti stará „garbage collector“, který automaticky zajišťuje, aby byly nepoužívané objekty odstraněny z paměti. Dále se jedná o jazyk interpretovaný, což znamená, že program není překládán do strojového kódu, ale do Java bajtkódu. Ten je následně interpretován virtuálním strojem. Z toho vyplývá, že jazyk Java je multiplatformní, což představuje velikou výhodu,

protože můžeme aplikaci tohoto typu spustit nezávisle na operačním systému a architektuře zařízení. (13 stránky 15-16)

Java tedy zajišťuje funkční stránku vyvíjené aplikace a patří mezi nejpoužívanější jazyky pro programování Android aplikací.

Java v současné době existuje minimálně v pěti edicích a každá z těchto edic je zaměřená na různá prostředí, či zařízení.

- Java Card
- Java ME (Micro Edition)
- Java TV
- Java SE (Standard Edition)
- Java EE (Enterprise Edition)

Technologie Java Card poskytuje bezpečné prostředí pro aplikace běžící na „chytrých“ čipových kartách nebo na jiných důvěryhodných zařízeních s velmi malou pamětí a limitovaným výpočetním výkonem. Na jedno takové zařízení může být umístěno více aplikací. (16)

Java Micro Edition je výkonné a přizpůsobivé prostředí pro aplikace běžící na vestavěných a mobilních zařízeních s přístupem k internetu. Například mobilní telefony, různé senzory, PDA zařízení, set-top boxy nebo tiskárny. Java ME má flexibilní uživatelské rozhraní, silné zabezpečení, zabudované internetové protokoly a podporu pro síťové i offline aplikace, které mohou být staženy dynamicky. (17)

Java TV je technologie vycházející z Java ME. Je určena pro vyvíjení aplikací, které budou fungovat na televizích a set-top boxech. Java TV poskytuje vývojářům jednoduše naprogramovat například elektronické programové průvodce (doplňující informace k pořadům běžícím v televizi), hry a vzdělávací aplikace nebo aplikace s přístupem na internet (klienti pro sociální sítě atd.). (18)

Java Standard Edition je standardní verze Javy a je běžně používaná. Umožňuje vyvíjet a spouštět aplikace na osobních počítačích a také na serverech.

Java Enterprise Edition je verze určená pro rozsáhlejší počítačové systémy, například pro firmy.

Hlavní vlastnosti jazyku Java (19):

- Jednoduchost – zjednodušená syntaxe.
- Objektivě orientovaný jazyk.
- Distribuovatelnost – navržený pro funkčnost aplikací v síti.
- Interpretovaný jazyk – multiplatformní.
- Robustnost – určen pro vyvíjení spolehlivého software.
- Správa paměti garbage collectorem.
- Víceúlohový jazyk.
- Dynamický jazyk.

2.1.2 XML

XML (Extensible Markup Language) je značkovací jazyk podobný HTML. Je to jednoduchý, na textu založený formát představující strukturované informace, například různé dokumenty, konfigurace, knihy atd. (20)

XML jako jazyk sám o sobě nic nedělá. Jsou to jen informace obsažené mezi jednotlivými značkami. XML potřebuje podpůrný software, který bude s těmito informacemi pracovat. (21)

V Android aplikacích XML popisuje, jak bude vypadat uživatelské rozhraní této aplikace.

2.1.3 C/C++

Jazyk C byl vyvinut Kenem Thompsonem a Dennisem Ritchiem začátkem sedmdesátých let dvacátého století. Jedná se o jazyk procedurální, nízkoúrovňový a kompilovaný. Procedurální znamená, že kód tohoto jazyka je strukturován do funkcí, které mezi sebou mohou být libovolně volány. Nízkoúrovňový jazyk je takový, který pracuje na úrovni strojových instrukcí. Kompilovaný představuje jazyk, který musí být nejdříve přeložen překladačem do strojového jazyka a poté může být teprve spuštěn. (22)

Jazyk C++ byl vyvinut Bjarnem Soustrupem na začátku osmdesátých let dvacátého století jako jazyk objektivě orientovaný. Je založený na jazyku C, na kterém v době vzniku C++

běžel v podstatě veškerý systémový software. Důvodem vzniku C++ byla potřeba lepšího strukturování větších softwarových projektů a zvýšení jejich výkonu. Byl vyvinut tak, aby byl spustitelný na jakémkoli operačním systému v někdejší době. (23)

C a C++ řídí většinu funkcí v chytrých mobilních telefonech, jako je například jádro operačního systému, které komunikuje s fyzickými prvky zařízení, nebo knihovny, které zajišťují chod zařízení. (23)

Programování Android aplikací v jazyce C/C++ není tolik rozšířené jako programování v Javě. Vývojáři ve společnosti Google se snaží o to, aby se tento jazyk používal co nejméně. Avšak mají pochopení pro to, že v některých případech se tomu nedá vyhnout. Je proto možné naprogramovat nezbytné části kódu v C/C++ a zbytek aplikace v jazyku Java. Použití jazyka C/C++ souvisí převážně s vývojem her. (24)

Pro vývoj aplikací v C/C++ je potřeba sada nástrojů, která se nazývá Android NDK (Native Development Kit). Tato sada je dostupná na stránkách <http://developer.android.com/ndk/downloads/>.

2.1.4 LUA

Lua je rychlý, odlehčený, výkonný a procedurální skriptovací jazyk. Stejně jako Java je to jazyk interpretovaný a též má automatickou správu paměti, která probíhá za pomoci garbage collectoru. (25)

Jazyk Lua si vysloužil reputaci díky jeho výkonu. Několik benchmarků (testy srovnávající výkon tohoto jazyka s konkurenčními) ukazuje, že v oblasti interpretovaných jazyků, je nejrychlejší. Pro ještě vyšší výkon existuje LuaJIT (Lua Just-In-Time), která funguje na takovém principu, že kód kompiluje až tehdy, když je ho potřeba. (25)

2.1.5 HTML, CSS A JAVASCRIPT

HTML (Hyper Text Markup Language) je značkovací jazyk pro vytváření webových dokumentů. Obsahuje sadu značek. Každá značka popisuje jinou část obsahu stránky např. odstavec, nadpisy atd. (26)

CSS (Cascading Style Sheets) neboli kaskádové styly je jazyk který popisuje, jak budou dokumenty v HTML nebo XML vizuálně prezentovány. Tyto styly mohou být zařazené přímo do kódu HTML nebo mohou být uloženy v externím souboru. Jeden externí soubor potom může ovlivňovat vzhled více stránek. (27)

JavaScript je vysokoúrovňový, dynamický a interpretovaný skriptovací jazyk využívaný při tvorbě webových dokumentů. Stejně jako CSS se může zapisovat přímo do HTML kódu nebo může být v samostatném externím souboru. Skripty tohoto jazyka jsou umístěny na serveru, ale provádí se na straně klienta. To znamená, že JavaScript obsažený na navštívené stránce se celý posílá na klientský počítač. Tam se teprve provede.

Spojením těchto tří jazyků lze také vytvářet aplikace pro systém Android ve vývojovém prostředí zvaném PhoneGap.

2.1.6 METODA „DRAG AND DROP“

Tato metoda není jazyk jako takový, ale umožňuje programovat aplikace bez nutnosti znát jakýkoli programovací jazyk. Funguje na principu přetahování hotových bloků kódu, kterým se pouze nastavují parametry a události které mají nastat při průběhu daného bloku.

Tato metoda je vhodná zejména pro výukové účely a pochopení základních logických principů programování.

2.2 VÝVOJOVÁ PROSTŘEDÍ

2.2.1 ANDROID STUDIO

Android Studio je oficiální nástroj od společnosti Google pro vyvíjení Android aplikací, jehož předchůdcem byl Eclipse s pluginem ADT (Android Development Tools). Je založeno na softwaru IntelliJ IDEA, který slouží k vývoji aplikací v mnoha jazycích. Android studio převzalo editor kódu, vývojové nástroje a celkový vzhled prostředí. Dále nabízí ještě další funkce, které by měli zvýšit celkovou produktivitu při budování Android aplikací. K těmto funkcím patří například našeptávač kódu, nebo šablony kódu pro běžné aplikace. (28)

V Android Studiu je možné programovat jak v jazyce Java, tak i v jazyce C/C++. Je tedy nutností znát alespoň jeden z nich. Vzhledovou stránku aplikace je zde možno realizovat metodou „drag and drop“ u nepřiliš složitých aplikací nebo přímo psaním kódu XML. Při použití metody „drag and drop“ se u každé komponenty dají nastavit různé parametry, bez psaní jakéhokoliv kódu.

Android Studio ve svém instalačním balíku také obsahuje emulátor, který umožní vyzkoušet aplikaci rovnou na virtuálním zařízení či ji nahrát přímo do mobilního zařízení. Obě možnosti vyžadují k vyzkoušení (testování) aplikace nástroj Android Debug Bridge

(adb), mobilní zařízení navíc příslušné ovladače. Nástroj adb umožňuje komunikaci počítače s virtuálním i připojeným zařízením.

Tento program lze využít k vyvíjení základních i složitějších aplikací, stejně tak jako k vývoji her.

2.2.2 STENCIL

Stencil je prostředí pro vývoj herních aplikací. Je prezentován jako sada nástrojů, která zvýší výkon vývojáře. (29)

Je volně dostupný avšak publikování aplikací na některé platformy se musí platit. Mezi tyto platformy bohužel patří i systém Android.

Platformy, pro které je možné aplikace publikovat:

- iOS
- Android
- Windows
- Mac
- Linux
- Flash
- HTML5 (publikování pro tuto platformu je v experimentální fázi)

Celé toto vývojové prostředí funguje na principu „drag and drop“, takže vývojář nemusí znát žádný programovací jazyk pro to, aby mohl vytvořit svojí vlastní hru.

2.2.3 MIT APP INVENTOR

MIT App Inventor je online vývojové prostředí, které by mělo uvést začínající programátory do problematiky programování aplikací pro systém Android. Lze jej stáhnout a používat lokálně a offline. Má jednoduché uživatelské rozhraní a opět se jedná o nástroj, ve kterém funguje vývoj aplikací na základě metody „drag and drop“. (30)

Tato aplikace by mohla být dobře využitelná k výukovým účelům, neboť díky jejímu jednoduchému uživatelskému rozhraní se v tomto prostředí velice dobře orientuje. Ke každé komponentě, která může být umístěna do grafického rozhraní vyvíjené aplikace,

je připojena krátká nápověda, která stručně popisuje, co konkrétní komponenta dělá. Pokud nám nápověda nestačí, lze přejít na stránku, kde jsou veškeré informace.

2.2.4 CORONA SDK

Corona SDK je multiplatformní vývojové prostředí, které umožňuje vývojářům vytvářet 2D hry a aplikace pro mobilní telefony, televize a počítače. Prostředí je navrženo tak, aby vývoj probíhal velice rychle, a to až desetkrát rychleji než u ostatních. Dynamické zpracování kódu umožňuje vidět prováděné změny okamžitě. (31)

Platformy, pro které je možné aplikace publikovat:

- iOS
- Android
- Kindle
- Windows Phone 8
- Apple TV
- Android TV
- Mac OS X
- Windows

Corona SDK využívá programovacího jazyka Lua a je tedy nutná jeho znalost.

2.2.5 PHONEGAP

PhoneGap je v podstatě tunelové propojení počítače a mobilního telefonu za účelem promítání vytvářené aplikace přímo do telefonu. Vyvíjenou aplikaci lze upravovat v jakémkoliv textovém editoru, jelikož se jedná o soubory HTML, CSS a JavaScriptu. Dále je potřeba rozšíření do příkazového řádku PhoneGap CLI. Když je aplikace hotová, pomocí příkazového řádku se zkompiluje a vznikne soubor s příponou apk, který lze nainstalovat na jakémkoliv zařízení se systémem Android.

2.3 KLASIFIKACE NÁSTROJŮ A MOŽNOSTÍ

Základním rozřazením vývojových nástrojů je to, jestli pro vývoj aplikace potřebujeme znát nějaký programovací jazyk. Jak již bylo popsáno, ne každý nástroj toto vyžaduje.

Potřeba znalosti nějakého vývojového jazyka:

- Android Studio
- (Eclipse)
- Corona SDK
- PhoneGap

Vývojová prostředí, u kterých se bez této znalosti obejdeme:

- Stencyl
- MIT App Inventor

Důležitým faktorem klasifikace může být cena nástroje. Všechny výše popsané nástroje jsou volně dostupné, avšak publikování aplikací pro Android ve vývojovém prostředí Stencyl je placené. Abychom mohli aplikace publikovat, je potřeba zaplatit poplatek ve výši 200\$ na jeden rok. Mezi další nástroje, za jejichž chod musíme zaplatit, patří například GameSalad nebo GameMaker.

Nástroje lze klasifikovat také podle jazyku, který v nich lze využít.

Java:

- Android Studio
- (Eclipse)

C/C++:

- Android Studio
- (Eclipse)

Lua:

- Corona SDK

XML:

- Android Studio
- (Eclipse)

HTML, CSS a JavaScript

- PhoneGap

Drag and drop:

- Stencyl
- MIT App Inventor
- Částečně Android Studio

Potencionální vývojář si také musí rozmyslet, jestli chce vyvíjet programovou aplikaci (např. formulářovou) nebo hru. Pokud bude záměrem vývojáře vyvinout hru, nejrychlejším nástrojem pro vývoj jednodušších her bude Stencyl. Stencyl je prostředí, ve kterém se dají vyvíjet pouze hry. V ostatních vývojových nástrojích se dají vyvíjet jak programové aplikace, tak i hry.

MIT App Inventor by mohl být zařazen do kategorie výukových nástrojů, jak je popsáno v kapitole 2.2.3.

2.4 KRITÉRIA PRO VÝBĚR VÝVOJOVÉHO PROSTŘEDÍ

Kritéria pro výběr vývojového prostředí souvisí s jejich klasifikací a záleží na subjektivních potřebách potencionálního vývojáře. První věcí, nad kterou by se měl uživatel zamyslet je, jaký jazyk ovládá nebo jestli je ochotný se některý z uvedených jazyků naučit. Pokud ne, bude mít na výběr jen z omezeného množství vývojových nástrojů. U některých nástrojů se také bude muset zamyslet, jestli je ochoten do vývojového nástroje v budoucnu investovat nějaké peníze. Jak už bylo řečeno, Stencyl toto vyžaduje k publikaci aplikací.

Dalším kritériem bude, co chce uživatel vyvíjet. Pokud to bude aplikace, zvolí MIT App Inventor. Pokud bude chtít vyvíjet hru, může si vybrat mezi zmiňovaným Inventorem a Stencylem. Stencyl je zaměřený na vývoj her a bude tedy snazší pro jejich vývoj. Pokud bude chtít vyvíjet aplikaci a přitom se nechce učit žádný programovací jazyk, bude si muset vystačit s MIT App Inventorem.

Pokud se uživatel rozhodne naučit se některý z jazyků, musí se také dále rozhodnout, jestli je ochoten se jich naučit více nebo pouze jeden. Pokud bude ochoten, se učit jen jeden, bude muset zvolit programovací jazyk Lua a vývojové prostředí Corona SDK. Dále uživateli zbývá výběr mezi Javou s XML (v pozdější fázi, dle potřeby i C/C++) nebo mezi jazyky

z webových technologií HTML, CSS a JavaScript. První možnost bude v začátcích asi složitější, ale bude v ní určitě daleko více možností a volnosti jak aplikaci tvořit.

Předchozí kapitola (2.3 Klasifikace nástrojů a možností) poskytuje uživateli přehled o tom, co bude k vývoji potřebovat a určit si jeho vlastní kritéria pro výběr.

3 ZÁKLADY ANDROID APLIKACE

Mobilní programová aplikace je programové vybavení mobilního zařízení provádějící nějakou činnost. Typickým příkladem mobilní aplikace dodávané přímo s operačním systémem mobilního telefonu je SMS klient, který umožňuje zasílat SMS zprávy, navigace, správce souborů atd.

3.1 TYPY APLIKACÍ

Aplikace vytvořené pro systém Android spadají do jedné z následujících tří kategorií (32 str. 39):

- **Aplikace na popředí** – Jsou to takové aplikace, které jsou většinou použitelné pouze, když běží na popředí. Příkladem mohou být hry. Pokud je aplikace přenesena do pozadí, jejich činnost je pozastavena. (32 str. 39)
- **Aplikace na pozadí** – Tyto aplikace běží na pozadí, mimo hlavní běh zařízení. Na zařízení je například spuštěna hra a aplikace na pozadí stále běží, bez ovlivnění aplikace na popředí. Příkladem může být sledování příchozích SMS. (32 str. 39)
- **Aplikace s přerušovanou činností** – Jedná se o aplikace, které také většinou pracují na pozadí, ale je od nich očekávána komunikace s uživatelem. Mezi tyto aplikace je možné zařadit například hudební přehrávač. (32 str. 39)

3.2 SLOŽENÍ APLIKACÍ

Android aplikace se skládají z určitých komponent. Následující kapitola popisuje nejdůležitější a nejzákladnější komponenty.

3.2.1 AKTIVITY (ACTIVITY)

Aktivita je část aplikace, která zajišťuje interakci s uživatelem skrze grafické rozhraní aplikace. Například vytočení telefonního čísla. V jedné aplikaci se většinou nachází více aktivit a jedna z nich je určená jako hlavní. Hlavní aktivita provádí hlavní funkci celé aplikace a bývá spuštěna a zobrazena ihned po spuštění aplikace. Každá z aktivit potom může spustit další aktivitu. Při spuštění jiné aktivity se předchozí aktivita pozastaví a uloží do paměti typu zásobník neboli LIFO. Pokud tedy bude v aplikaci stisknuto tlačítko zpět, z paměti se vezme poslední uložená aktivita a bude obnovena její činnost. (32 stránky 39-40)

3.2.2 SLUŽBY (SERVICES)

Služby jsou částí aplikace, které mohou běžet na pozadí a neumožňují uživateli s nimi komunikovat skrze grafické rozhraní aplikace. Služby mohou běžet, i pokud uživatel přejde do jiné aplikace nebo tuto aplikaci uzavře. Příkladem služby může být přehrávání hudby na pozadí. (32 str. 40)

3.2.3 POSKYTOVATELÉ OBSAHU (CONTENT PROVIDERS)

Poskytovatelé obsahu spravují přístup k datům v zařízení. Aplikace potom tedy mohou ukládat a načítat různá data. Poskytovatelé obsahu také zajišťují určitou úroveň zabezpečení dat. Příkladem poskytovatelů obsahu pro běžné typy dat mohou být například audio a video soubory nebo kontakty. (32 str. 40)

3.2.4 ZÁMĚRY (INTENTS)

Záměr je v podstatě zpráva pro ostatní aktivity aplikace, která se vyvolá po splnění nějaké podmínky. Příkladem může být navigace. Pokud se budeme nacházet v určité vzdálenosti od místa odbočení, aplikace ohlásí, že máme po určité vzdálenosti zatočit. Obsah záměru tedy může být informace o tom, jaká akce se má vykonat, jaký parametr ovlivňuje vykonání akce a informace o tom, jaká aplikace má akci provést. (32 str. 40)

Záměry se dají rozdělit do dvou kategorií:

- **Explicitní** – Obsahuje akci a specifický název aktivity nebo aplikace, která má akci provést. Explicitní záměry se většinou používají u aplikací, které samy vyvíjíme, protože známe názvy námi vytvořených aktivit. (32 str. 40)
- **Implicitní** – Tento záměr také obsahuje akci, které se má provést, ale už neobsahuje specifický název aktivity či aplikace. Při spuštění záměru, je systémem Android uživateli umožněn výběr aplikace, kterou chce spustit. (32 str. 40)

3.2.5 PŘIJÍMAČE (BROADCAST RECEIVERS)

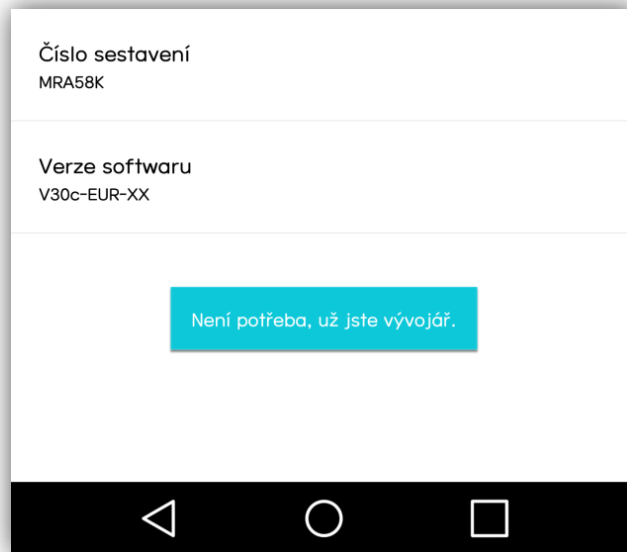
Přijímač je součástí aplikace, která má za úkol naslouchat oznámením zevnitř nebo z vnějšku aplikace. Podle toho jaké oznámení přijímač obdrží, následuje určitá reakce. Aplikace mohou přijímat oznámení buď systémová, nebo taková, jaká si uživatel sám naprogramuje. Přijímače nedisponují uživatelským rozhráním. (32 str. 41)

3.2.6 OZNÁMENÍ (NOTIFICATIONS)

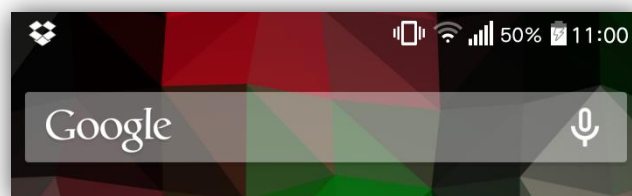
Oznámení mohou informovat uživatele o tom, že v aplikaci nastala nějaká určitá situace. Uživatel na toto oznámení reagovat může i nemusí, v některých případech ani nemůže.

Existují tři typy oznámení:

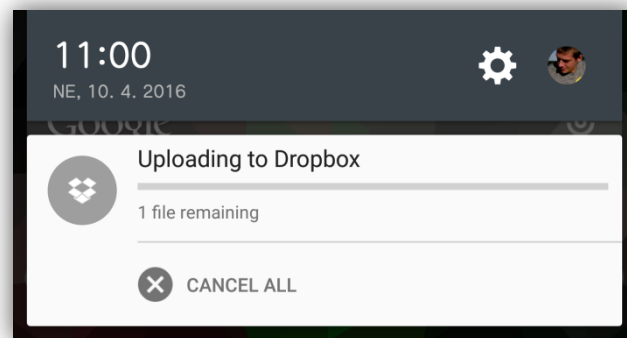
- **Toast oznámení** – Tato oznámení se zobrazují na aktuální obrazovce, na které uživatel právě pracuje. Většina těchto oznámení se vztahuje k tomu, co právě uživatel dělá. Například po odeslání SMS zprávy se může zobrazit krátké oznámení o doručení této zprávy nebo po odemčení nástrojů pro vývojáře, se po dalším pokusu o otevření těchto nástrojů zobrazí text „Není potřeba, už jste vývojář“, jak je ukázáno na obrázku (Obrázek 3). Oznámení tedy zabere tolik prostoru, jako zabere jeho text. Uživatel na toto oznámení zareagovat nemůže a oznámení automaticky zmizí. Oznámení tohoto typu můžou vyvolat aplikace běžící na pozadí i na popředí. (32 str. 41)
- **Oznámení ve stavovém řádku** – Tyto oznámení přidají do stavového řádku ikonu aplikace, od které oznámení přišlo (v levém horním rohu se nachází ikona „rozbalené krabice“, viz Obrázek 4) v tomto případě od aplikace Dropbox a po rozbalení stavového řádku se zobrazí celá zpráva tohoto oznámení, viz Obrázek 5. Tyto oznámení mohou být signalizovány zvukovým doprovodem, vibracemi nebo světelnou signalizací pomocí LED diody, pokud jí telefon disponuje. (32 str. 41)
- **Oznámení formou dialogového okna** – Oznámení tohoto typu reagují na uživatelem právě prováděnou akci a souvisí s právě aktivní aplikací, ve které je akce prováděna. Příklad oznámení můžeme vidět na obrázku (Obrázek 6). Toto oznámení se zobrazilo při pokusu o smazání jedné z fotografií. (32 str. 41)



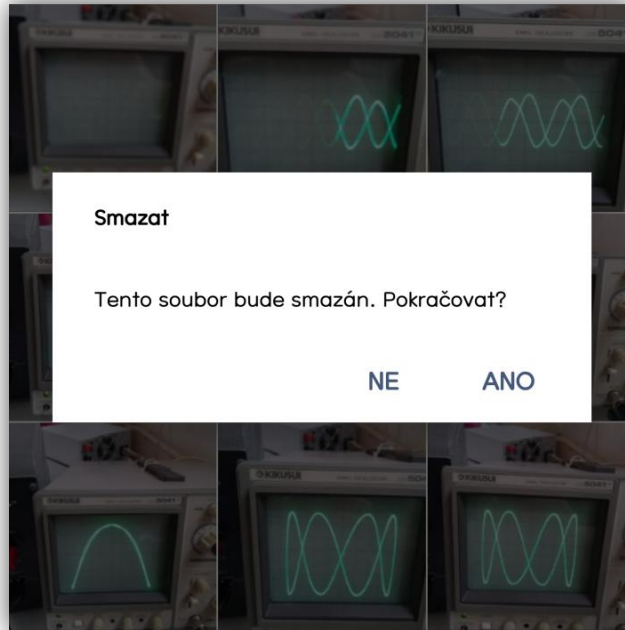
Obrázek 3: Toast oznámení (zdroj: vlastní)



Obrázek 4: Oznámení ve stavovém řádku 1 (zdroj: vlastní)



Obrázek 5: Oznámení ve stavovém řádku 2 (zdroj: vlastní)



Obrázek 6: Oznámení formou dialogového okna (zdroj: vlastní)

3.3 CO MÁ VÝVOJÁŘ PŘI VÝVOJI APLIKACE K DISPOZICI

3.3.1 ÚLOŽIŠTĚ

Úložiště slouží pro ukládání dat aplikace. Některé soubory, které zůstanou neměnné, mohou být zabaleny přímo v instalačním balíčku aplikace. Ty se poté zkopírují do paměti zařízení, nebo na externí zařízení (např. paměťová karta). Dále se dá využít online úložiště pro různá data ukládaná uživatelem a to tak, že se data budou ukládat do databází. To je přínosné zejména v případě, když si uživatel pořídí nové zařízení. Nemusí potom složitě kopírovat požadovaná data, pouze se připojí přes již vytvořený účet a data se mu automaticky do aplikace načtou. (33 str. 24)

3.3.2 SÍŤ

Veškerá zařízení v dnešní době mají přístup k internetu, jak skrze mobilní data, tak i přes připojení WiFi. I většina aplikací, které jsou k dispozici, komunikují online. Může to být například jen za účelem výdělků vložením reklam do aplikace. To ale neznamená, že každá aplikace vyžaduje internetové připojení. Toto připojení se dá využít různě, například již zmíněná reklama, či ukládání dat. (33 str. 24)

3.3.3 MULTIMÉDIA

Dalším typickým prvkem moderních zařízení je například pořizování fotografií, nebo nahrávání a přehrávání videozáznamů nebo audia. Každé zařízení se liší konkrétními

specifikacemi hardwaru, který tyto služby poskytuje. To znamená, že se budou pořízené multimediální záznamy a výstupy lišit kvalitou. Aplikace se může na tyto specifika zařízení zeptat a poté je uživateli prezentovat. Například v aplikaci fotoaparát máme možnost vybrat, v jakém rozlišení chceme fotografie pořizovat. Každý s těchto multimediálních prvků může být použit ve vyvíjené aplikaci, záleží pouze na uživateli, zda tam daný prvek použije či nikoli. (33 str. 24)

3.3.4 GLOBAL POSITIONING SYSTÉM (GPS)

GPS přijímač bývá také součástí každého novodobého chytrého zařízení. Slouží k určení polohy zařízení na celém světě. Programátor má k tomu to přijímači přístup. Využití záleží na tom, jakou aplikaci vyvíjí. Může jej využít například pro navigace nebo pro detekci polohy v případě ztráty nebo odcizení zařízení. (33 str. 24)

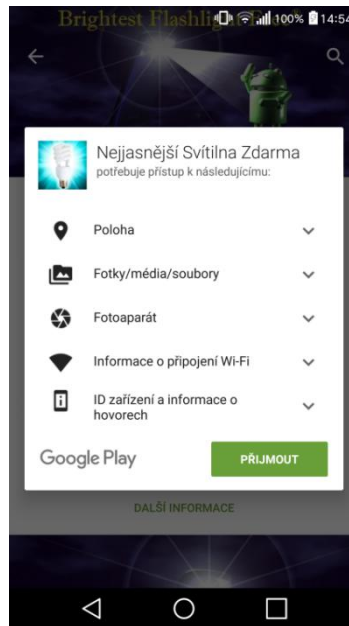
3.3.5 TELEFONNÍ SLUŽBY

Telefonní služby jsou nedílnou součástí každého mobilního telefonu. Je tedy možné, aby se v aplikacích dali posílat SMS zprávy nebo se skrze ně dalo telefonovat. (33 str. 24)

3.4 OPRÁVNĚNÍ PRO APLIKACE

Některé aplikace pro systém Android vyžadují speciální oprávnění, které musí potvrdit uživatel, aby aplikaci vůbec mohl nainstalovat. Pokud aplikace toto oprávnění nemají, nemohou přistupovat k určitým komponentám nebo údajům, které jsou potřebné pro správnou funkčnost aplikace. V nejnovější verzi systému Android, je možné některá oprávnění aplikaci odepřít, může to však zapříčinit nesprávnou funkčnost nebo úplnou nefunkčnost aplikace. Při instalaci musí být oprávnění potvrzena a aplikace se při pokusu například o pořízení fotografie musí dotázat, jestli jí to uživatel povolí. (13 stránky 61-62)

Někdy jsou ovšem oprávnění pro určitou aplikaci nesmyslná. Mohou být do aplikace zařazené za účelem získat od uživatele citlivé údaje. Například aplikace v Obchodě Play „Nejjasnější Svítilna Zdarma“, viz Obrázek 7, vyžaduje údaje o poloze, připojení k WiFi, hovorech a přístupu do úložiště. Přitom jedinou činností, kterou má tato aplikace vykonávat, je rozsvítit diodu blesku. To znamená, že jediné potřebné oprávnění je pro přístup k fotoaparátu.



Obrázek 7: Nesmyslná oprávnění aplikace (zdroj: vlastní)

Realita v dnešní době je taková, že uživatel, který si chce nainstalovat nějakou aplikaci, většinou oprávněním nevěnuje pozornost a jednoduše veškerá oprávnění potvrdí, přitom mohou být data zneužita.

Oprávnění, která aplikace vyžaduje, musí být uvedena v souboru AndroidManifest viz kapitola 3.5. (13 str. 62)

Oprávnění existuje velké množství, proto si zde uvedeme osm základních skupin:

- **CALENDAR** – Oprávnění související s kalendářem.
- **CAMERA** – Oprávnění umožňující aplikaci přistupovat k fotoaparátu a zachycovat fotografie či videa nebo rozsvítit diodu sloužící jako blesk.
- **CONTACTS** – Oprávnění umožňující přístup ke kontaktům a profilům nastaveným v telefonu.
- **LOCATION** – Oprávnění umožňující přistupovat k poloze zařízení.
- **MICROPHONE** – Oprávnění umožňující přístup k mikrofonu.
- **PHONE** – Oprávnění umožňující přístup k informacím o telefonu. Například stav nabití baterie.
- **SMS** – Oprávnění související s prací se SMS zprávami (odesílání, čtení atd.)
- **STORAGE** – Oprávnění pro přístup k externímu úložišti.

3.5 ANDROID MANIFEST

Manifest je soubor ve formátu XML (AndroidManifest.xml) a je umístěn v kořenovém adresáři, který je základem každé aplikace. Deklaruje se v něm obsah aplikace, jako jsou aktivity, služby, oprávnění atd. Dále jsou zde uloženy informace, o jakou verzi se jedná, jak se aplikace jmenuje nebo jaká je minimální verze systému Android, na kterém aplikace poběží. (32 stránky 45-49)

4 PŘEDSTAVENÍ VÝVOJOVÉHO PROSTŘEDÍ ANDROID STUDIO

Tato část je zaměřena prakticky na představení vývojového prostředí Android Studio 2.0 (teoretické základy jsou uvedeny v kapitole 2.2.1).

4.1 INSTALACE POTŘEBNÝCH NÁSTROJŮ

Jak již bylo řečeno, k vyvíjení aplikací v Android Studiu je potřeba více komponent než samotné vývojové prostředí. Tyto komponenty jsou Java Development Kit (JDK), Android Software Development Kit (SDK) a samotné Android studio. Kde získat tyto komponenty je popsáno v kapitole 2.

Je možné, že JDK je již na počítači nainstalováno. To je možné zjistit spuštěním příkazového řádku a zadáním příkazu `javac -version`. Výpis příkazového řádku po zadání tohoto příkazu by mohl vypadat takto: `javac 1.8.0_60`. Pokud příkazový řádek příkaz nerozezná, JDK není nainstalováno. Je tedy nutné stáhnout instalační soubor a JDK nainstalovat podle pokynů průvodce instalací. Verze JDK, kterou Android Studio momentálně požaduje, je verze poslední a to JDK8. (34)

Dalším krokem je instalace Android Studia. Výhodou je, že Android Studio v instalačním balíčku obsahuje nástroj Android SDK. Při instalaci se opět budeme řídit pokyny průvodce. Ve druhém kroku průvodce nabízí instalaci SDK a instalaci virtuálního zařízení. Nutné je zaškrtnout položku SDK, abychom vůbec mohli aplikace vyvíjet. Virtuální zařízení není podmínkou, ale pokud vývojář nevlastní zařízení, na kterém může vyvíjenou aplikaci testovat, je dobré ho nainstalovat též. V dalším kroku se průvodce ptá, kam nainstalovat Android Studio a sadu nástrojů SDK. Pro SDK je vhodné zvolit cestu, která v názvech jednotlivých složek neobsahuje diakritiku, protože by Android Studio po instalaci hlásilo cestu jako neplatnou. Další kroky je třeba provést dle instrukcí průvodce. V posledním kroku se zobrazí zaškrťovací políčko, které se ptá, jestli má být Android Studio rovnou spuštěno či nikoliv. Defaultně je zaškrtnuta možnost ano. Pokud se tedy spustit nemá, je třeba volbu-změnit.

4.2 PRVNÍ SPUŠTĚNÍ

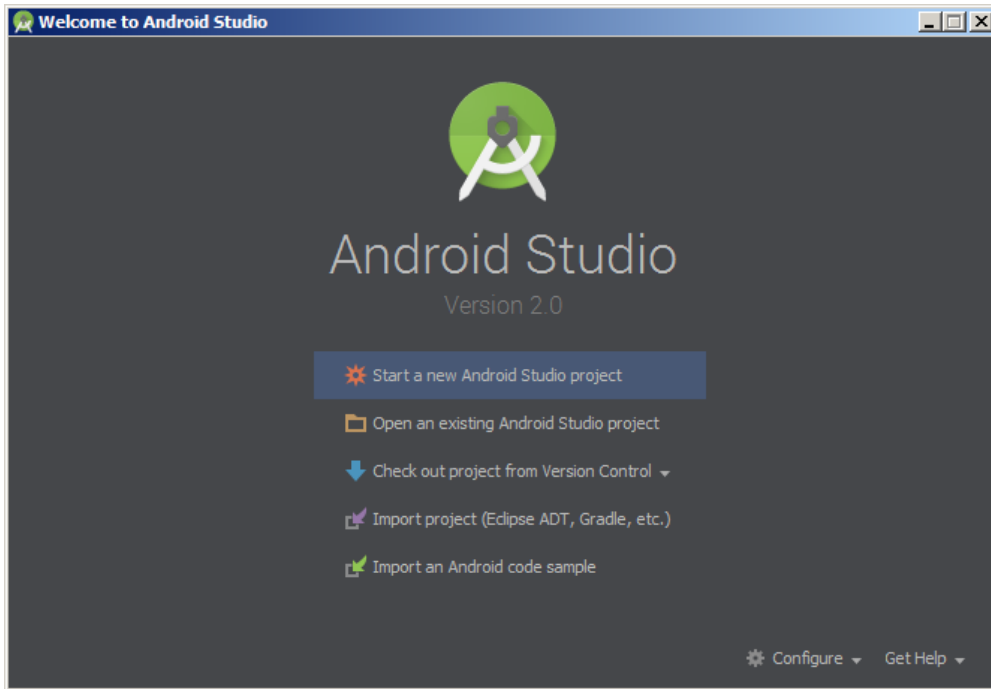
Při prvním spuštění se Android studio zeptá, zdali uživatel chce importovat nějaká předchozí nastavení. Pokud již bylo studio v předchozí době nainstalováno a z nějakého důvodu opět odinstalováno, poskytne uživateli cestu k souborům z předchozí instalace. Pokud instalace dříve neproběhla, vybere možnost, která říká, že se žádné předchozí nastavení v počítači nenachází nebo ho z nějakého důvodu importovat uživatel nechce.

V dalším kroku Android studio uživatele uvítá a nabídne doinstalování různých SDK. Dále také oznámí, že po dokončení konfigurace, bude možné otevřít již existující aplikaci nebo začít vytvářet novou.

Po kliknutí na tlačítko další se zobrazí nabídka nastavení Android Studia. To může být nastaveno standardně, to znamená, že nastavení bude provedeno podle nejpoužívanějšího nastavení většiny uživatelů. Tuto volbu Android Studio doporučuje. Druhou volbou je nastavení vlastní, kde uživatel může nastavit téma (vzhled) uživatelského rozhraní, jestli chce doinstalovat nějaké rozšiřující balíčky pro vytváření aplikací Android. Je zde i možnost nainstalovat přednastavené virtuální zařízení.

Dále pak průvodce spuštěním zjistí, zda může emulátor běžet ve zvýšeném výkonu a vyžádá si číselnou hodnotu, která pro něho může být rezervována z paměti RAM. Poté se zobrazí dokončovací obrazovka. Po odkliknutí dokončovací obrazovky průvodce začne stahování a instalace dodatečných komponent.

Po dokončení instalace se zobrazí úvodní nabídka Android Studia, viz Obrázek 8.



Obrázek 8: Úvodní obrazovka po spuštění Android Studia (zdroj: vlastní)

Na výběr je zde z několika možností:

- Start a new Android Studio project – Vytvoření nového projektu.
- Open an existing Android Studio project – Otevření existujícího projektu.
- Check out project from Version Control – Otevření projektu z cloudového úložiště jako je například Google Cloud nebo GitHub.
- Import project (Eclipse ADT, Gradle, etc.) – Umožňuje importovat projekt z dalších vývojových prostředí.
- Import an Android code sample – Umožňuje importovat existující vzorové projekty, které jsou uloženy online na úložišti GitHub.

Dále se v pravém dolním rohu nachází dvě nabídky – Configure a Get Help.

V nabídce Configure stojí za zmínku tyto možnosti:

- SDK Manager – viz kapitola 4.3.
- Plugins – Zde se nachází nabídka pluginů (rozšiřujících modulů), které je možné do Android Studia přidat.

- Export a Import settings – Slouží k uložení aktuálního nastavení Android Studia nebo k načtení uloženého.
- Check for Update – Kontrola nejnovějších aktualizací Android Studia.

V nabídce Get Help se nachází tyto možnosti:

- Help Topics – Tato záložka nás přesměruje na webové stránky, kde se nacházejí užitečné rady všeho druhu.
- Tip of the Day – Otevře okno s náhodným tipem, který nám poradí užitečnou funkci Android Studia.
- Default Keymap Reference – Otevře online PDF soubor s klávesovými zkratkami.
- Plugin Development – Otevře webovou stránku s pluginy, kde se nachází jejich zdrojové kódy a také informace o pluginech.

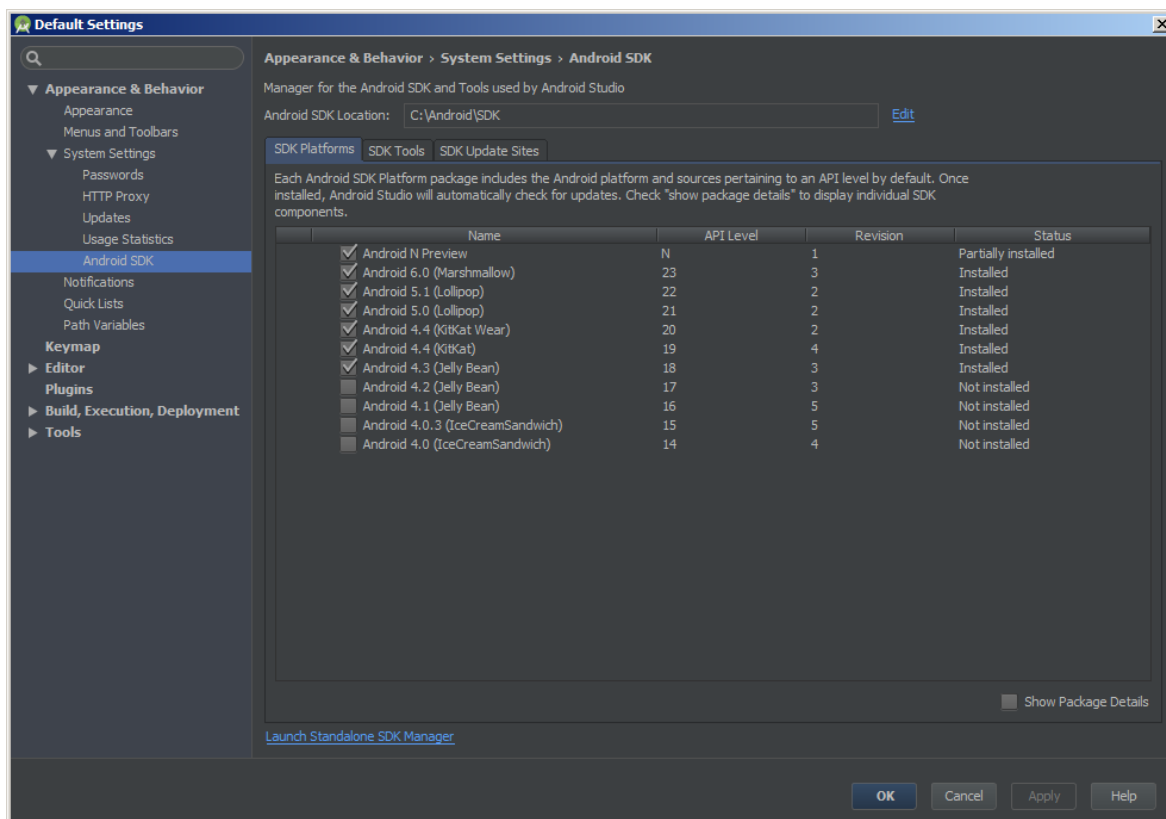
4.3 SDK MANAGER

SDK Manager je manažer Android Studia, ve kterém lze stáhnout balíčky nástrojů související s vývojem aplikací. Nachází se zde tři záložky:

- SDK Platforms
- SDK Tools
- SDK Update Sites

4.3.1 SDK PLATFORMS

Zde je možné stáhnout balíčky SDK pro různé verze systému Android, důležité k vývoji aplikací pro každou z těchto verzí.

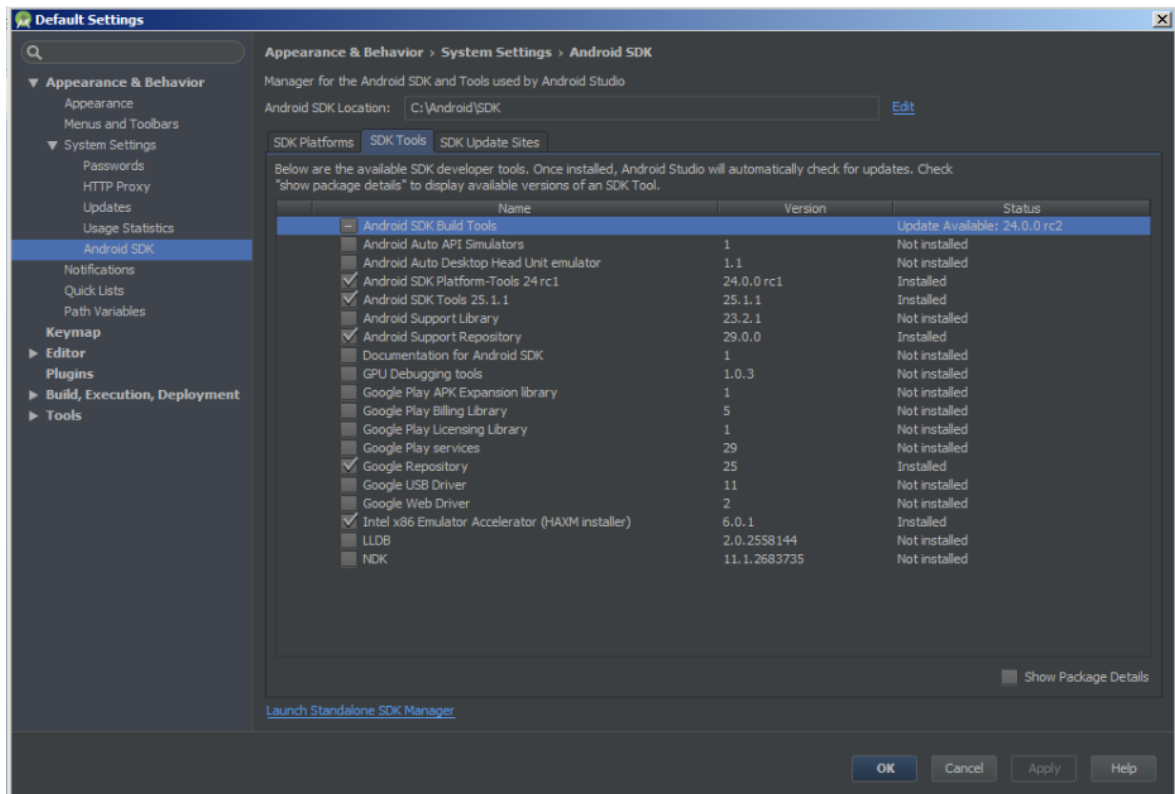


Obrázek 9: SDK Manager - SDK Platforms (zdroj: vlastní)

4.3.2 SDK TOOLS

V záložce SDK Tools umožňuje manažer přidat do Android Studia podpůrné nástroje pro vývoj aplikací. Po instalaci každého z nástrojů bude Android Studio automaticky kontrolovat jejich aktualizace.

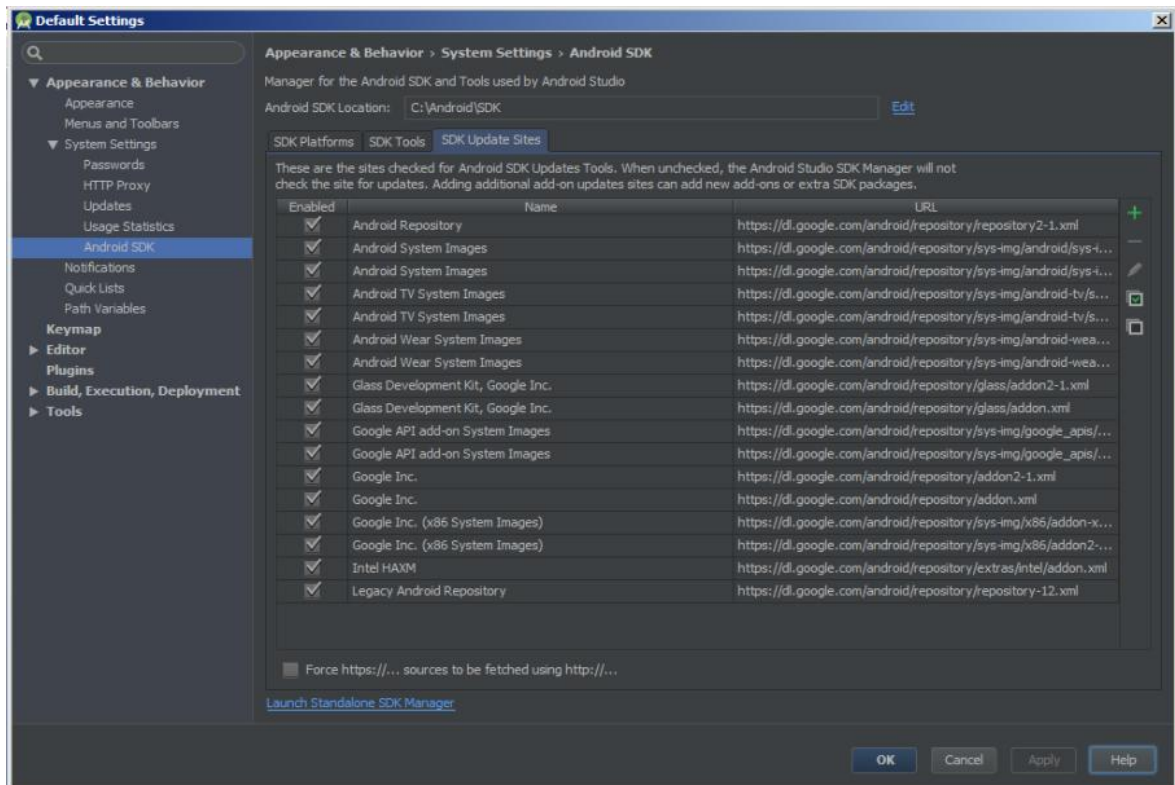
Pokud uživatel při instalaci zvolil standardní postup, nebude nainstalován nástroj Intel x86 Emulator Accelerator. Tento nástroj je důležité nainstalovat pro použití emulátoru zařízení, na kterém bude možné aplikace testovat. Google USB Driver (ovladače pro komunikaci s mobilním zařízením) je dobré nainstalovat, pokud je aplikace testována na zařízeních Google Nexus. V jiném případě je potřeba nainstalovat ovladače od konkrétního výrobce mobilního telefonu, který bude k testování použit.



Obrázek 10: SDK Manager - SDK Tools (zdroj: vlastní)

4.3.3 SDK UPDATE SITES

V této sekci se nachází seznam zdrojů, ze kterých budou kontrolovány aktualizace pro nainstalované nástroje. Pokud není zaškrťovací okénko u nějakého ze zdrojů zaškrtnuté, aktualizace se kontrolovat nebudou. S přidáním dalšího nástroje se mohou přidat i další zdroje ke kontrole aktualizací.

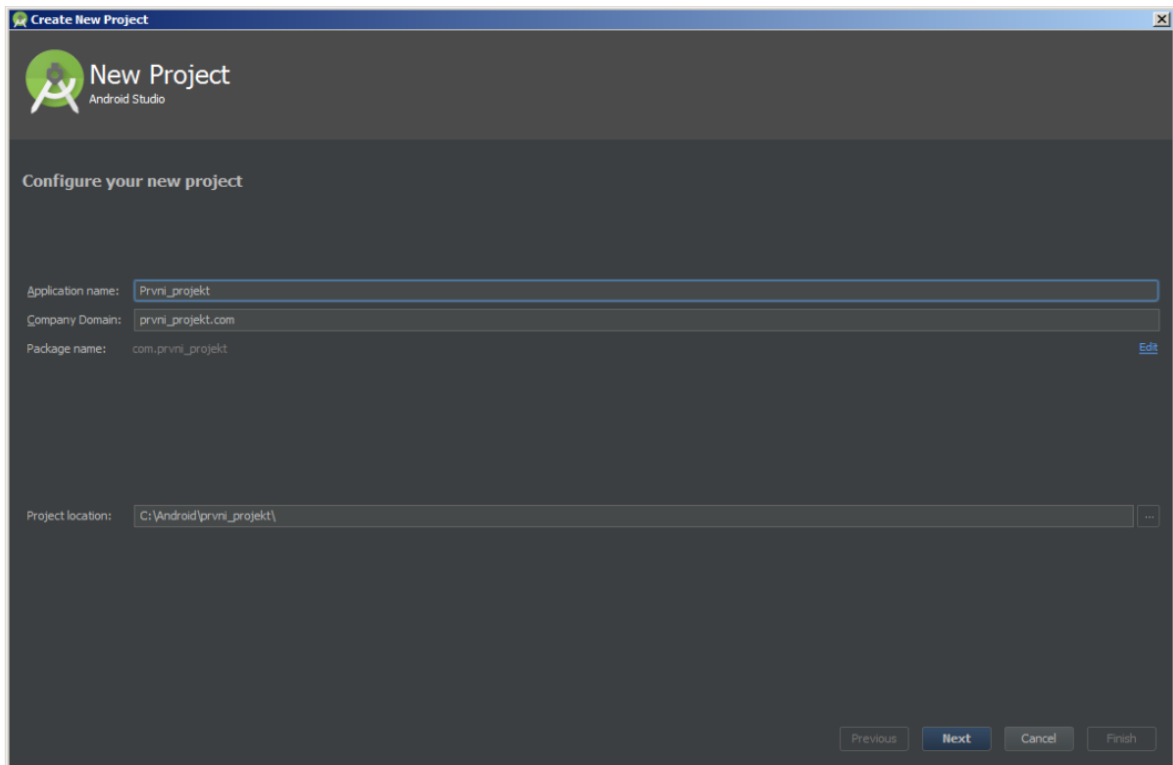


Obrázek 11: SDK Manager - SDK Update Sites (zdroj: vlastní)

4.4 VYTVOŘENÍ PRVNÍHO PROJEKTU

Jak už bylo řečeno, nový projekt se vytvoří kliknutím na záložku „Start a new Android Studio project“ v úvodním menu, viz Obrázek 8. Otevře se průvodce vytvořením nového projektu, viz Obrázek 12. Nachází se zde několik položek k nastavení:

- **Application name** – Kolonka pro vyplnění jména aplikace. Musí začínat velkým písmenem.
- **Company Domain** – Kolonka pro webovou adresu tvůrce aplikace.
- **Package name** – Název balíčku vytvořené aplikace. Mění se automaticky, podle Company Domain. Tento atribut lze ovšem upravit po kliknutí na tlačítko „Edit“.
- **Project location** – Místo na disku, kam se má projekt ukládat. Cesta k umístění nesmí obsahovat diakritiku.



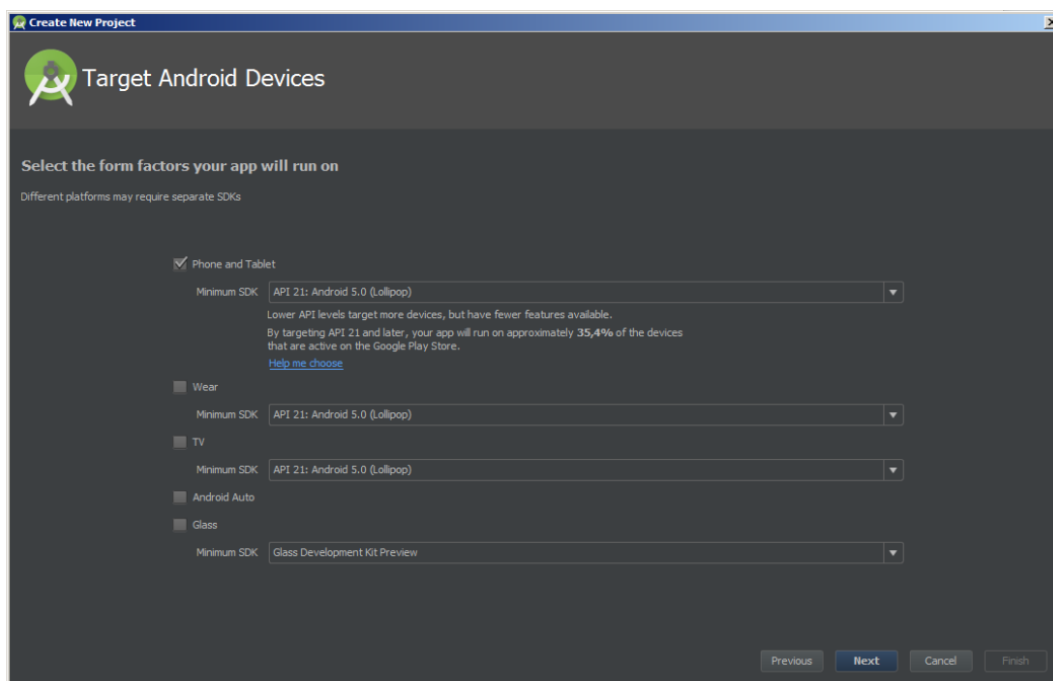
Obrázek 12: Průvodce novým projektem - Specifikace parametrů (zdroj: vlastní)

V následujícím kroku je umožněn výběr zařízení, pro které bude aplikaci vyvíjena, a na které nejnižší verzi systému Android bude aplikaci možné spustit. V jednom projektu je možné vyvíjet aplikaci, která bude pracovat na více zařízeních.

Výběr je možné provést z těchto zařízení:

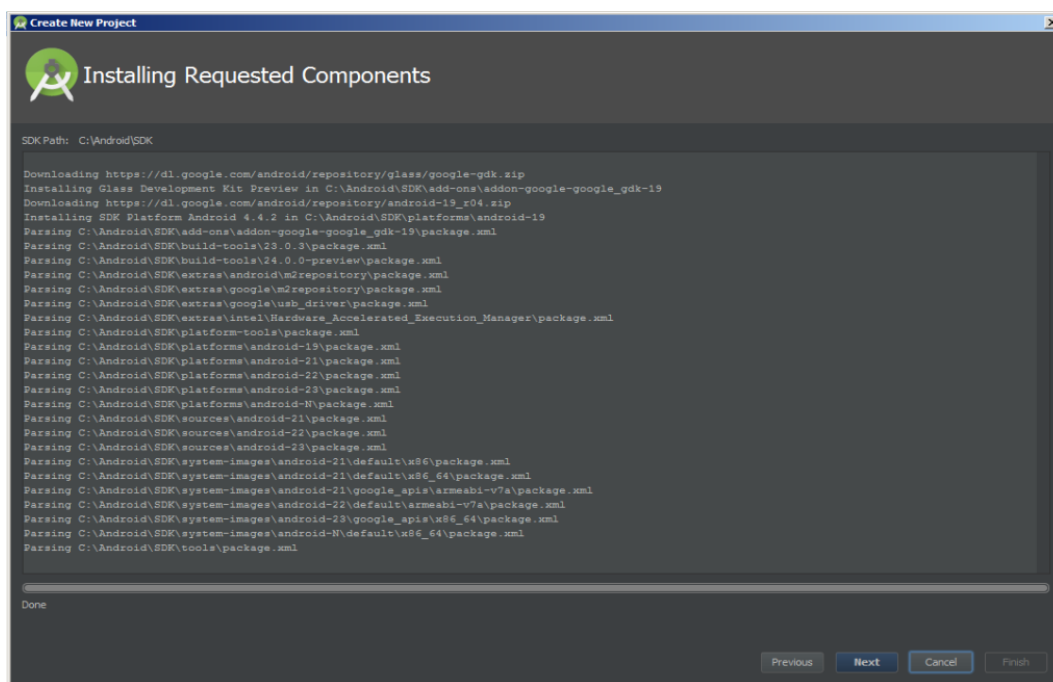
- Telefony a tablety.
- Android wear (doplňky, např. chytré hodinky).
- Televize.
- Auta se systémem Android.
- Brýle.

S výběrem verze, pro kterou bude aplikace vyvíjena, pomůže sám průvodce. Je zde řečeno, že s vybráním nižší verze, bude aplikace fungovat na více zařízeních. Bohužel s výběrem nižších verzí nebude možné používat funkce z verzí novějších. Průvodce též zobrazí přibližné procento zařízení, na kterých tato aplikace bude moci být nainstalována. Kliknutím na položku „Help me choose“ zobrazí jednotlivé verze a k nim funkce, které je možné využít. U každé verze se též zobrazuje celkové procento podporovaných zařízení.



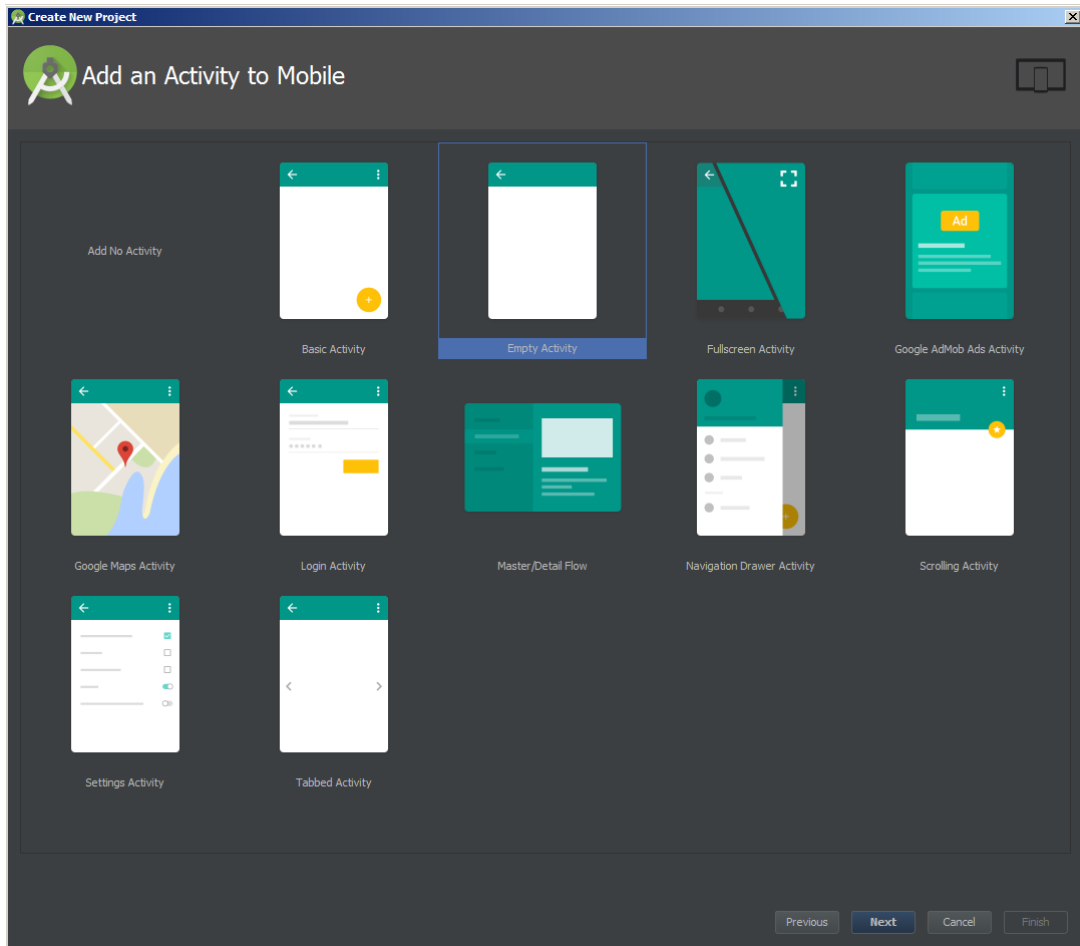
Obrázek 13: Průvodce novým projektem – Výběr API a zařízení (zdroj: vlastní)

Po kliknutí na tlačítko další se začnou stahovat potřebné nástroje pro vývoj aplikace vzhledem k nastavení. Pokud při vytvoření dalšího projektu, bude vybrána jiná verze, pro kterou bude aplikace vyvíjena, je možné, že průvodce začne stahovat další potřebné komponenty.



Obrázek 14: Průvodce novým projektem - Stahování potřebných nástrojů (zdroj: vlastní)

Další krok nabídne výběr aktivity. Android studio nabízí možnost vybrat z aktivity již předpřipravené. Například aktivitu, která bude pracovat s Google Mapami nebo aktivitu, která bude pracovat v režimu celé obrazovky. Defaultně je zvolena aktivita prázdná.

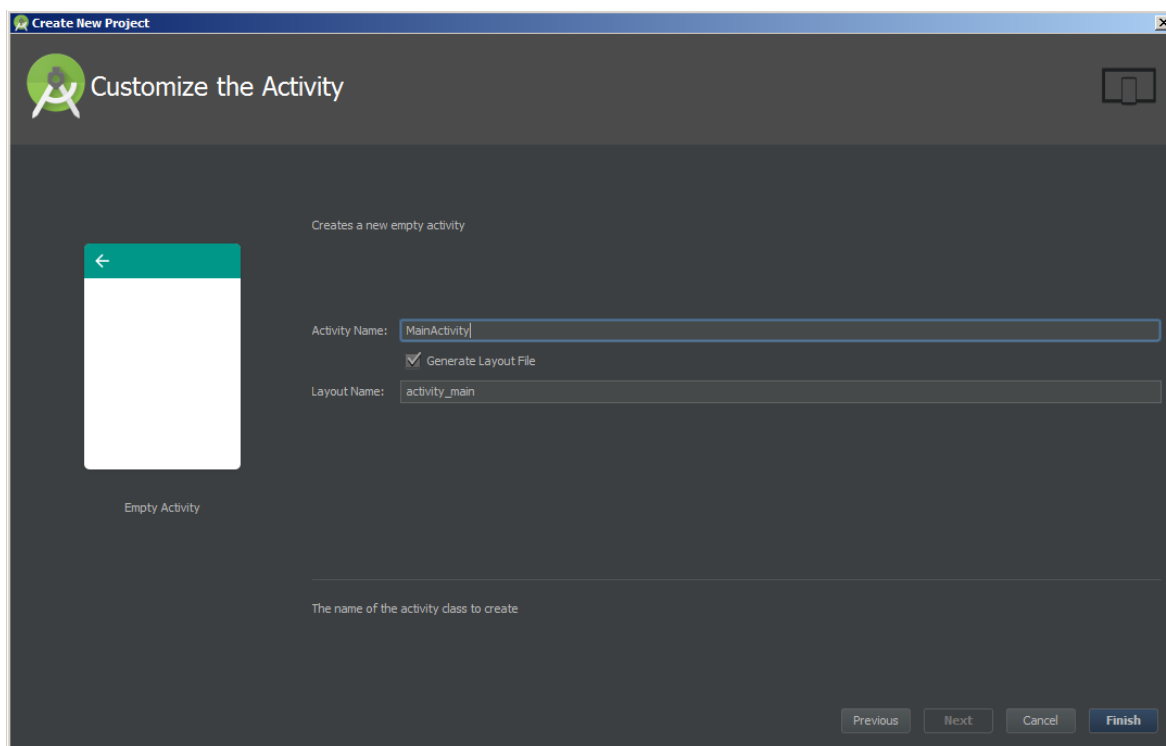


Obrázek 15: Průvodce novým projektem - Výběr aktivity (zdroj: vlastní)

Ve finálním kroku si průvodce vyžádá pojmenování hlavní aktivity aplikace a k ní připojeného XML souboru, viz Obrázek 16.

Jsou zde dvě kolonky:

- **Activity Name** – Kolonka pro pojmenování hlavní aktivity.
- **Layout Name** – Kolonka pro pojmenování XML souboru přiřazeného k této aktivitě. Rozložení pro aktivitu vygenerováno být nemusí, pokud nebude zaškrtnuto zaškrťovací políčko pojmenované „Generate Layout File“ neboli vygenerovat soubor s rozložením projektu.



Obrázek 16: Průvodce novým projektem - Pojmenování hlavní aktivity (zdroj: vlastní)

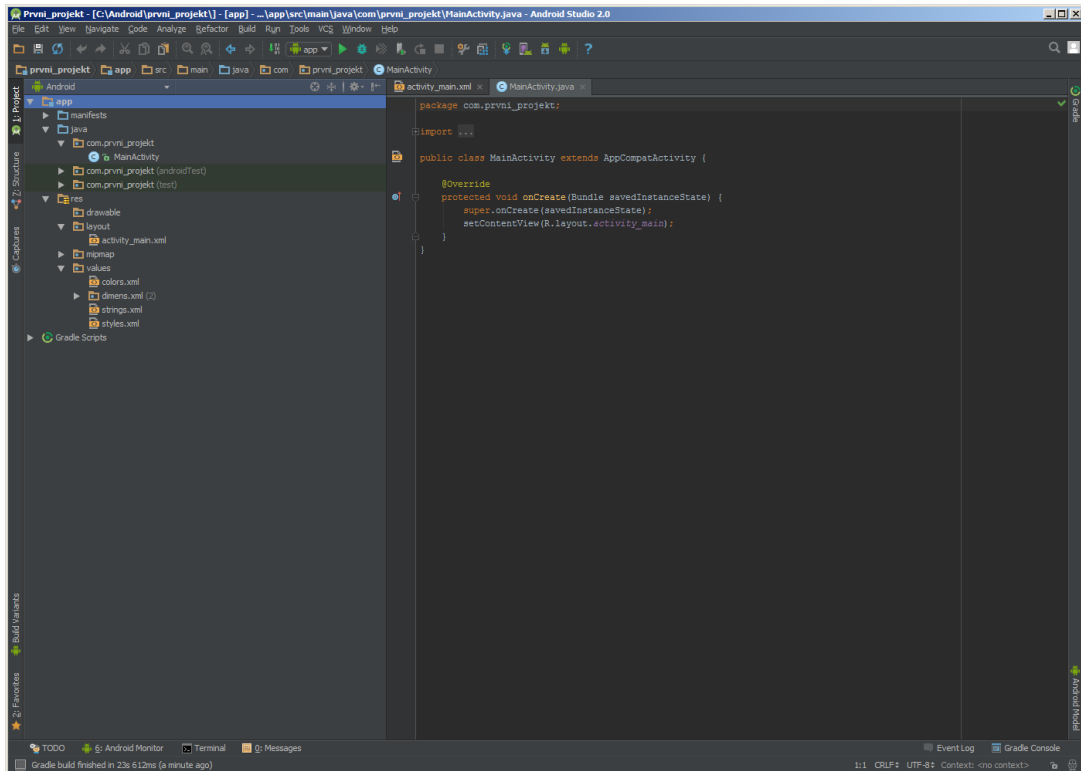
Po dokončení finálního kroku se spustí samotné Android Studio, viz Obrázek 17. V pravé části se nachází editor kódu (aktivní karta MainActivity.java), popřípadě editor uživatelského rozhraní aplikace (aktivní karta activity_main.xml, viz Obrázek 18).

Editor uživatelského rozhraní je dále rozdělen na několik částí, viz Obrázek 18. Hlavní částí je virtuální zařízení, na kterém se mohou přesouvat komponenty z jejich nabídky. Tato nabídka se nachází vlevo a obsahuje veškeré komponenty, které je možné do uživatelského rozhraní umístit. Vpravo je potom možné vidět, které komponenty byly již použity a jestli spadají pod jiné komponenty. Dále je zde možnost zadávat a upravovat atributy komponent.

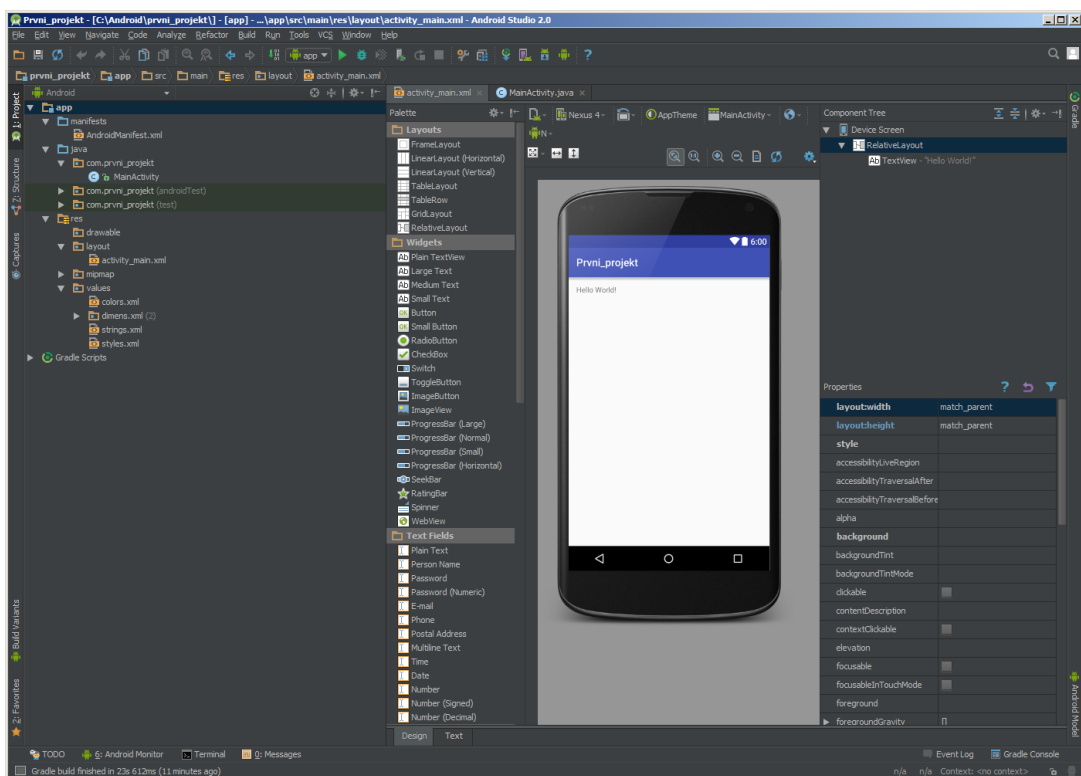
Editor uživatelského rozhraní se dá taktéž přepnout do editoru kódu. To se provede přepnutím mezi dvěma panely, které se nachází pod nabídkou komponent. Jedná se o panely Design a Text. Po přepnutí do panelu text, viz Obrázek 19, se zobrazí editor kódu a vpravo opět virtuální zařízení, na kterém můžeme okamžitě vidět změny provedené v kódu (Obrázek 18).

V levé části prostředí se nachází navigační panel, který umožňuje procházet jednotlivé soubory aplikace, které mohou být editovány. Veškeré aktivity aplikace by se měly

nacházet ve složce Java → com.prvni_projekt. Na obrázku (Obrázek 17) je možné vidět jedinou vytvořenou aktivitu – MainActivity.

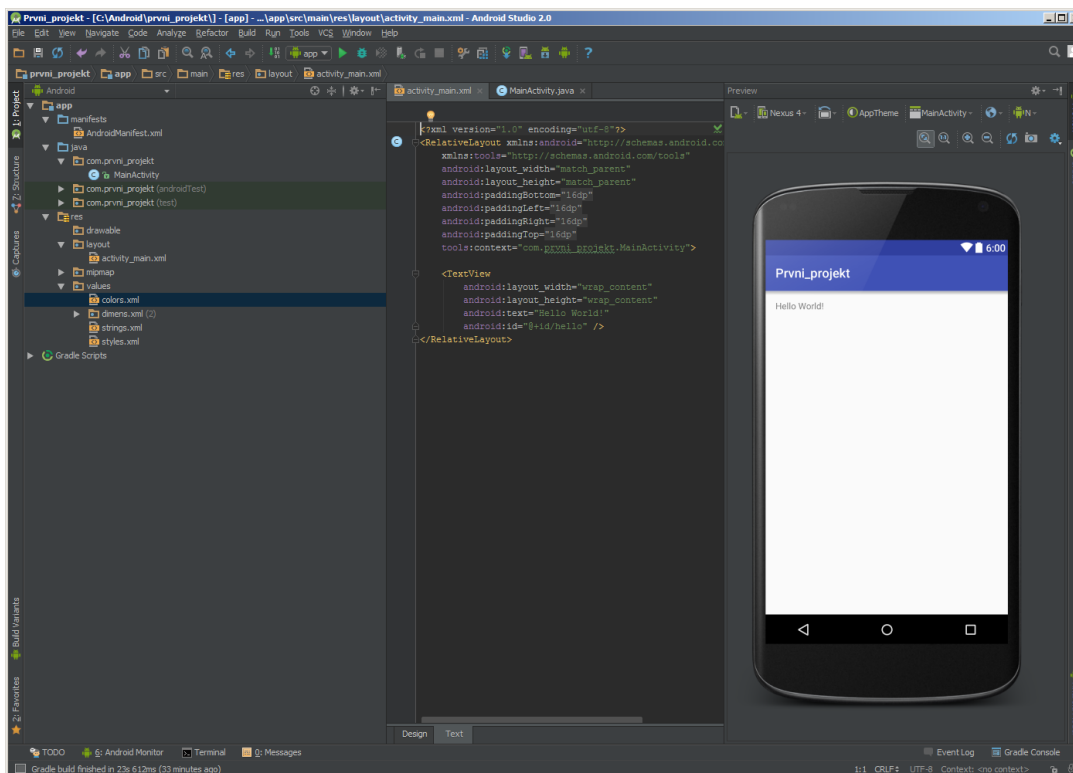


Obrázek 17: Android Studio 1 (zdroj: vlastní)



Obrázek 18: Android Studio 2 (zdroj: vlastní)

Soubory související se vzhledem aplikace se nachází ve složce `res`. V podsložce `layout` je soubor `activity_main.xml`, který určuje vzhled uživatelského rozhraní hlavní aktivity. Dále pak ve složce `values` se nachází například soubor `colors.xml`, ve kterém se definují barvy.



Obrázek 19: Android Studio 3 (zdroj: vlastní)

V panelu nástrojů jsou zobrazeny klasické nástroje jako je uložit, otevřít, kopírovat atd. Dále je zde možnost aplikaci spustit (zelená šipka). Aplikaci je možno spustit buď v emulátoru, nebo v připojeném zařízení. Emulátor i zařízení musí být správně nastaveny (viz kapitola 4.5). Dále se zde nachází debugger a odkazy na SDK Manager, AVD Manager nebo nápovědu.

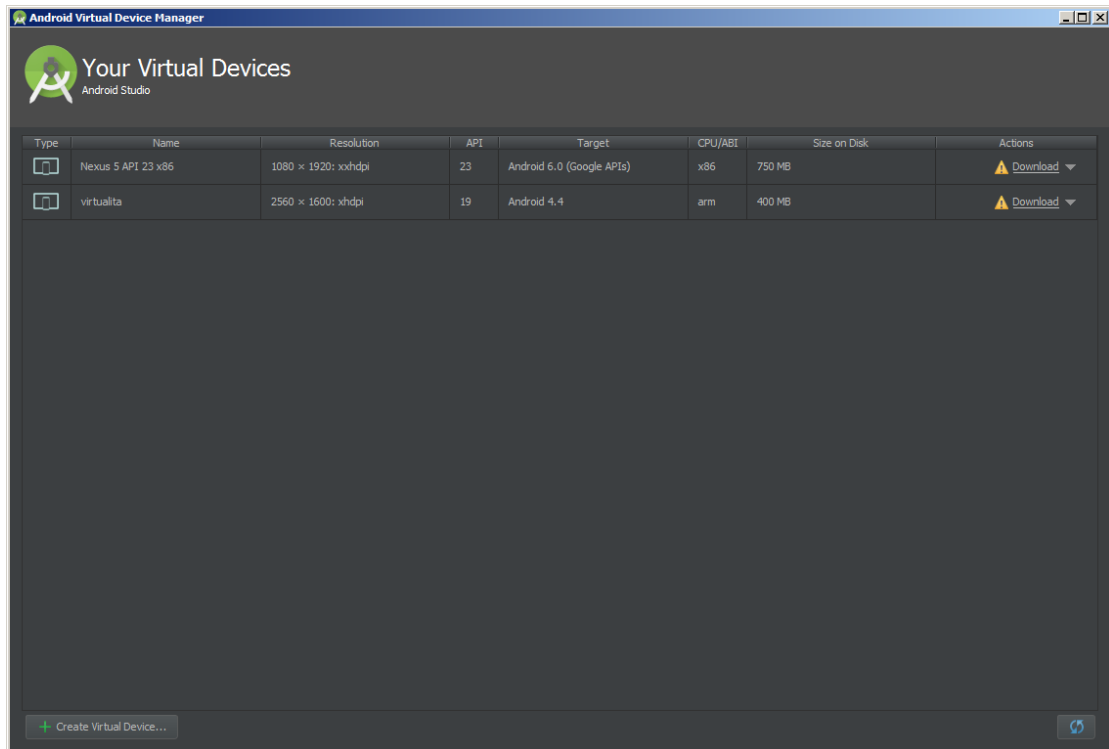
4.5 NASTAVENÍ EMULÁTORU A ZAŘÍZENÍ

Aby mohla být aplikace vyzkoušena v reálných podmínkách, je třeba mít k dispozici fyzické zařízení nebo vytvořený emulátor virtuálního zařízení. Jak bylo již řečeno, pro obě možnosti je důležité mít nainstalován nástroj `adb` (Android Debug Bridge).

Nastavení emulátoru se nachází ve správci virtuálních zařízení. Ten je možné najít v panelu nástrojů a to v záložce `Tools` → `Android` → `AVD Manager`. Po otevření správce se otevře okno s virtuálními zařízeními, viz Obrázek 20.

Zde je možnost stáhnout již předpřipravené virtuální zařízení a to Nexus 5. Uvedené je zde rozlišení obrazovky, verze systému Android, na kterém zařízení běží, architektura procesoru a kolik bude virtuální zařízení zabírat místa na disku.

V levé dolní části okna se nachází tlačítko „Create Virtual Device...“. To umožní vytvořit vlastní virtuální zařízení.



Obrázek 20: Správce virtuálních zařízení (zdroj: vlastní)

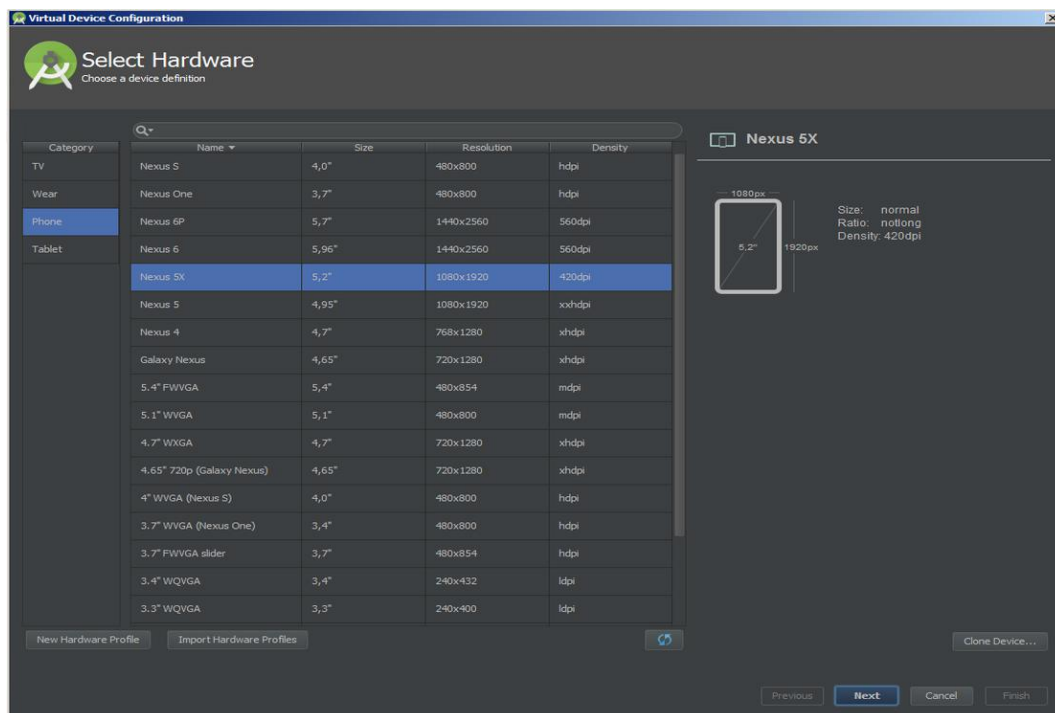
Po kliknutí na uvedené tlačítko se otevře průvodce, viz Obrázek 21, který umožňuje toto zařízení vytvořit.

Je možné vytvořit čtyři typy virtuálních zařízení:

- Simulace TV
- Simulace Android wear
- Simulace telefonu
- Simulace tabletu

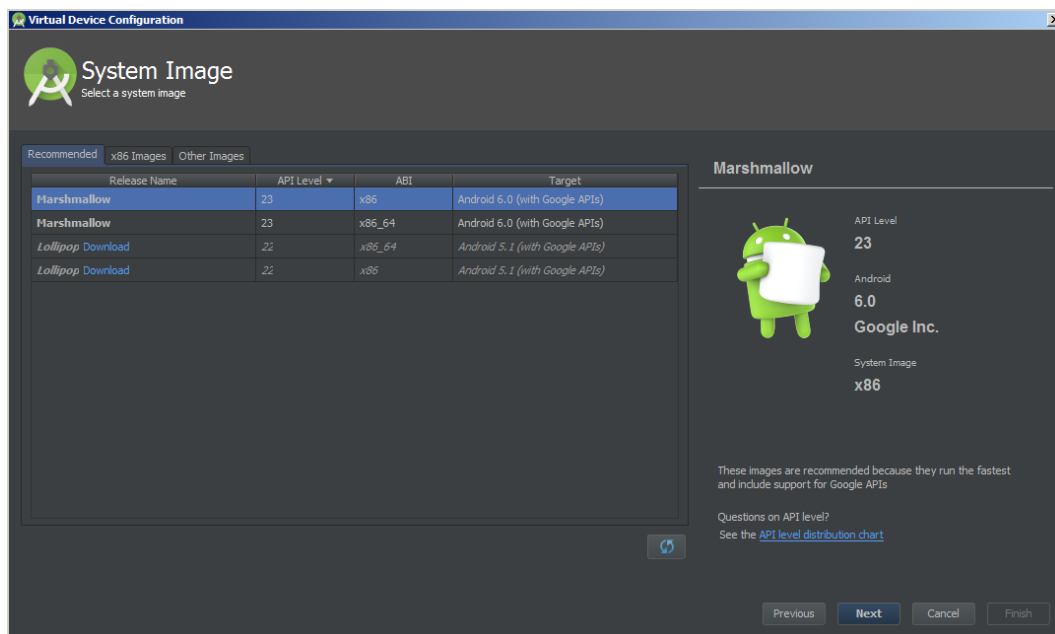
Dále má uživatel možnost vybrat konkrétní zařízení z uvedené skupiny, které má být vytvořeno. Nachází se zde také možnost vytvořit zařízení s vlastními parametry a to pod

tlačítkem „New Hardware Profile“. Stejně tak je možné načíst již vytvořené zařízení ze souboru.



Obrázek 21: Průvodce vytvořením virtuálního zařízení 1 (zdroj: vlastní)

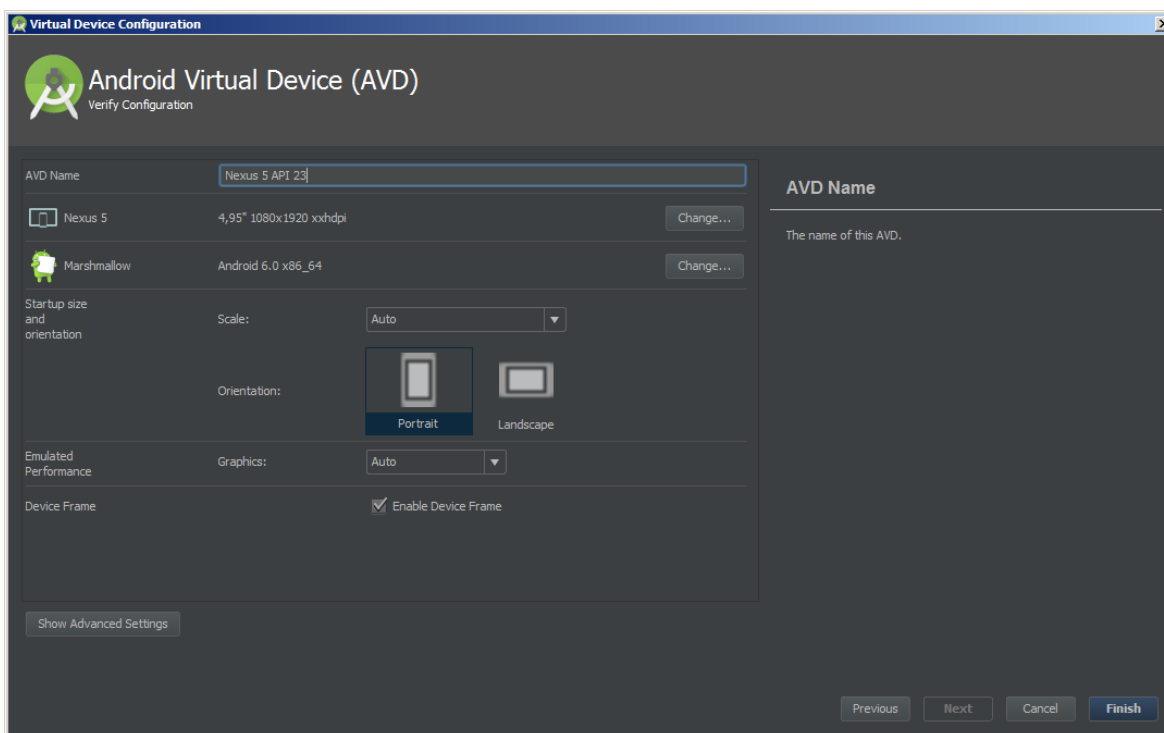
Další krok nabídne systémové obrazy Androidu. Systémový obraz určuje, na jaké verzi Androidu virtuální zařízení poběží. Ve většině případů je vhodné vybrat verzi poslední, jelikož aplikace pro nižší verze je spustitelná i na verzi vyšší, viz Obrázek 22.



Obrázek 22: Průvodce vytvořením virtuálního zařízení 2 (zdroj: vlastní)

V následujícím kroku, viz Obrázek 23, průvodce vyzývá k pojmenování virtuálního zařízení. Je zde i možnost změnit již vybrané zařízení a systémový obraz, který byl zvolen v krocích předchozích.

Dále je zde možnost nastavit měřítka v jakém se bude zobrazovat obrazovka zařízení a jak bude obrazovka orientována, jestli na výšku či na šířku. Následující položka (Emulated Performance) určuje, jak se budou emulovat grafické podklady a to buďto hardwarově za pomoci grafické karty, nebo softwarově. Pokud zvolíme možnost „Auto“, Android Studio rozhodne za uživatele na základě parametrů grafické karty. Jako poslední možnost je zde zaškrtnuté políčko, které při zaškrtnutí emulovanému zařízení přiřadí rámeček, stejný jako má reálné zařízení.



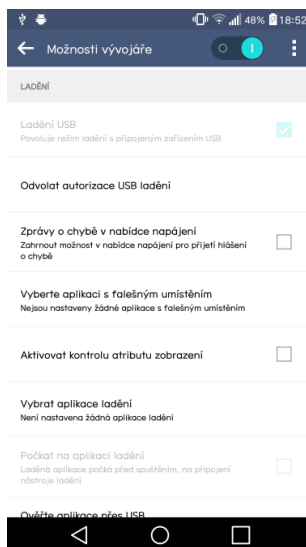
Obrázek 23: Průvodce vytvořením virtuálního zařízení 3 (zdroj: vlastní)

Po dokončení průvodce se zobrazí okno správce virtuálních zařízení, kde jsou zobrazena všechna vytvořená virtuální zařízení, viz Obrázek 20.

Pro testování aplikací přímo v mobilním zařízení je důležité mít na počítači nainstalované ovladače tohoto zařízení. Ty jsou důležité pro správnou komunikaci mezi zařízením a počítačem.

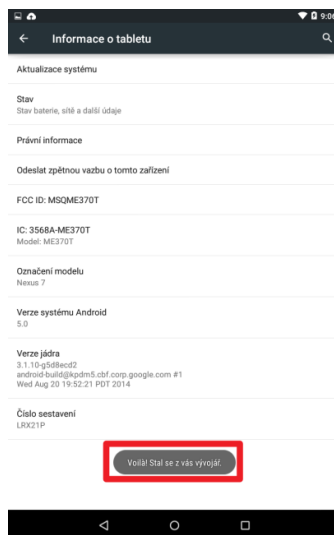
Zařízení musí mít zapnuté USB ladění, viz Obrázek 24. Tato položka se nachází v Nastavení → Možnosti vývojáře → Ladění USB. Mobilní telefony nemají možnosti vývojáře

defaultně odemčeny, proto je nutné tuto položku v nastavení odemknout. (35 stránky 46-47)



Obrázek 24: Ladění USB (zdroj: vlastní)

Odemčení možností vývojáře je možné provést v Nastavení → Info o telefonu → Softwarové informace → Číslo sestavení (toto platí konkrétně pro mobilní telefon LG G3). Pokud několikrát (přibližně sedmkrát) klikneme na tuto položku, odemknou se v nastavení možnosti vývojáře (upozorní nás na to krátké toast oznámení, viz Obrázek 25).



Obrázek 25: Odemčení možností vývojáře (zdroj: (36))

Po připojení zařízení k počítači by měl telefon automaticky spolupracovat s Android Studiem. Na novějších systémech je možné, že telefon bude vyžadovat povolení přístupu k mobilnímu telefonu ze strany počítače.

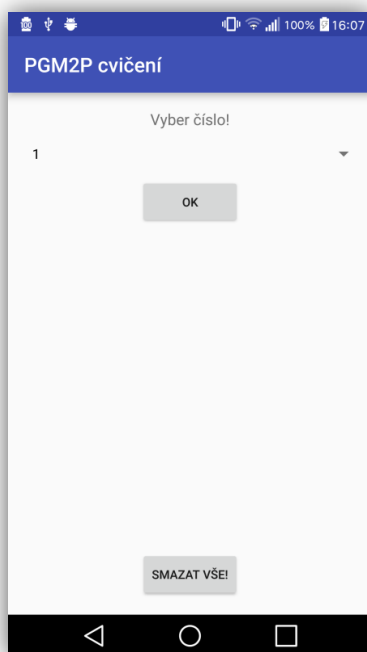
4.6 APLIKACE

4.6.1 PŘEDSTAVENÍ APLIKACE

Aplikace, která bude představena, byla původně vytvářena jako úkol na předmět Programování 2 (PGM2P) a to v programovacím jazyce Pascal. Aplikace není složitá, jejím cílem je pouze uvést uživatele do problematiky programování.

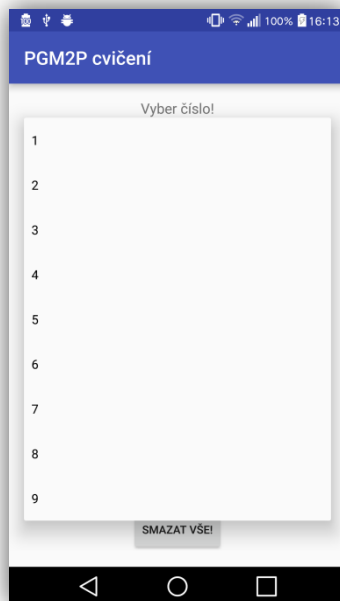
Úkolem je zobrazovat druhé mocniny jednoho z devíti čísel. Uživatel má na výběr z čísel od jedné do devíti. Číslo může vybírat prostřednictvím rozbalovací nabídky. Po výběru čísla, je nutné stisknout tlačítko OK. Po jeho stisknutí se vygenerují tlačítka s hodnotou od jedné do uživatelem zvoleného čísla. Kliknutím na některé z tlačítek se zobrazí dialogové okno, které oznámí hodnotu druhé mocniny konkrétního čísla. Vygenerovaná tlačítka mohou být smazána dalším tlačítkem – SMAZAT VŠE! Pokud si uživatel vybere číslo jiné (nižší) a opět stiskne tlačítko OK, přebytečná tlačítka zmizí.

Na obrázku (Obrázek 26) je možné vidět, jak aplikace vypadá ihned po spuštění.



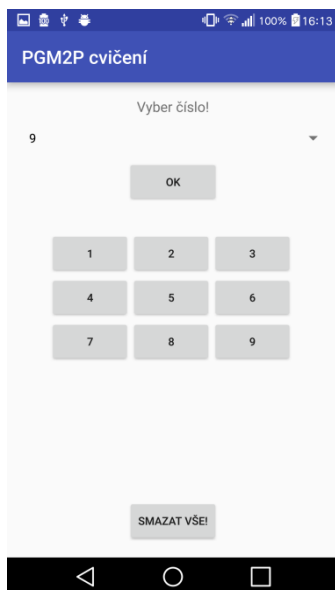
Obrázek 26: Aplikace po spuštění (zdroj: vlastní)

Nabídka pro výběr požadovaného čísla je zobrazena na obrázku (Obrázek 27).

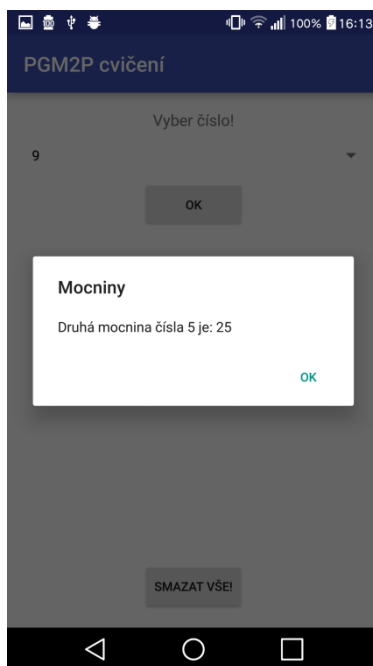


Obrázek 27: Výběr požadovaného čísla (zdroj: vlastní)

Na obrázku (Obrázek 28) je možné vidět, jak vypadá aplikace po výběru čísla a stisknutí tlačítka OK. Obrázek 29 potom ukazuje dialogové okno po stisknutí čísla pět.



Obrázek 28: Potvrzení výběru čísla (zdroj: vlastní)



Obrázek 29: Stisknutí čísla (zdroj: vlastní)

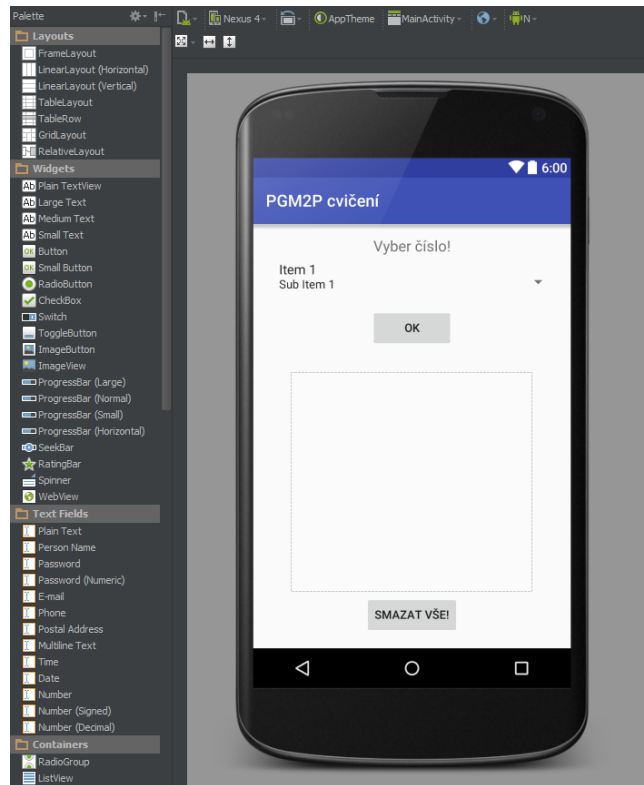
4.6.2 APLIKACE ZE VNITŘ

Prvním krokem bylo umístění komponent, které mají pevně danou pozici, na virtuální zařízení, viz Obrázek 30. Komponenty byly přidány přetažením z jejich nabídky.

Vložené komponenty:

- TextView – Slouží k zobrazení textu, v tomto případě „Vyber číslo!“.
- Spinner – Rozbalovací nabídka pro výběr čísla.
- Button – Tlačítka pro potvrzení zadaného čísla a smazání vytvořených komponent.
- GridLayout – Kontejner, do kterého se vkládají dynamicky vytvářená tlačítka.

Tyto komponenty byly přidávány na virtuální zařízení, které má defaultně nastavený kontejner RelativeLayout neboli relativní rozložení komponent v tomto kontejneru. Toto rozložení je možné změnit.



Obrázek 30: Rozložení aplikace (zdroj: vlastní)

Z následujícího zdrojového kódu je možné vidět, jak vypadá část XML kódu, který patří k hlavní aktivitě. XML soubor byl pojmenován `activity_main.xml` (kompletní kód se nachází v Příloze 1). Jak již bylo řečeno, aplikace není složitá, a proto je tento kód velice krátký.

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAlignment="center"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/nadpis"
    android:id="@+id/tv_vyber"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/se_cislo"
    android:layout_below="@+id/tv_vyber"
    android:layout_alignParentStart="true"
    android:spinnerMode="dropdown"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/ok"
    android:minWidth="100dp"
    android:onClick="onClick"
    android:layout_below="@+id/se_cislo"
    android:layout_centerHorizontal="true"
    android:id="@+id/button" />
```

```

<GridLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:layout_marginBottom="50dp"
    android:id="@+id/bt_layout"
    android:layout_below="@+id/button">
</GridLayout>

```

Nezákladnější atributy:

- **android:layout_width** – Udává šířku komponenty. Hodnoty mohou být zadávány číselně či slovně parametry `match_parent` (šířka bude stejná, jako šířka rodičovské komponenty) nebo `wrap_content` (přizpůsobení šířky obsahu).
- **android:layout_height** – Udává výšku komponenty. Hodnoty jsou zadávány obdobně jako u předchozího bodu.
- **android:text** – Text, který se zobrazí na komponentě.
- **android:id** – Slouží k identifikování komponenty.
- **android:onClick** – Udává název metody v aktivitě, která se má spustit po kliknutí na komponentu s tímto atributem.

Většinu těchto atributů doplní samo Android Studio na základě toho, jak komponenty rozložíme na virtuálním zařízení. Stejně tak doplní i ID. To je dobré změnit tak, aby se vývojář později v kódu vyznal. ID musí před konkrétním názvem obsahovat řetězec „@+id/“. Hodnoty z atributu `text`, je dobré vkládat do souboru `strings.xml` (obsah souboru `strings.xml` je možné vidět dále). Není to nutností, ale pokud bude v pozdější fázi aplikace lokalizována do více jazyků, usnadní to práci. Pokud budou hodnoty ukládány do `strings.xml`, musí před požadovaným textem obsahovat řetězec „@string/“. To odkazuje do již zmiňovaného souboru. Hodnoty veškerých atributů se musí vkládat do uvozovek. Kontejnery, do kterých se umisťují další komponenty, musí být uzavřeny (počáteční značka - `<>`, uzavírající značka `</>`), např. `<GridLayout> ... </GridLayout>`. Tyto značky jsou párové a další komponenty se vkládají mezi ně. V souboru `strings.xml` musí být takto uzavřeny veškeré komponenty.

Obsah souboru `strings.xml`:

```

<resources>
  <string name="app_name">PGM2P cvičení</string>
  <string name="nadpis">Vyber číslo!</string>
  <string name="ok">OK</string>
  <string name="delete">Smazat vše!</string>
  <string-array name="cisla">
    <item>1</item>
    <item>2</item>
    <item>3</item>
    <item>4</item>
    <item>5</item>
    <item>6</item>
    <item>7</item>
    <item>8</item>
    <item>9</item>
  </string-array>
</resources>

```

Položka `string-array` v souboru `strings.xml` slouží k naplnění Spinneru a naplnění je provedeno programově.

Kód, zajišťující funkčnost aplikace, se nachází v hlavní aktivitě `MainActivity.java` (kód hlavní aktivity aplikace se nachází v Příloze 2). Struktura kódu je stejná jako v Javě, začíná vždy údajem o tom, v jakém balíčku se aktivita nachází. Následuje výčet knihoven potřebných k chodu aplikace. Výhodou Android Studia je, že když je potřeba importovat knihovny, udělá to většinou automaticky při psaní kódu nebo napoví klávesovou zkratkou pro import (`Alt + Enter`). Pokud se tedy nejedná o knihovny externí např. námi vytvořené. Následující kód ukazuje knihovny importované v této konkrétní aktivitě.

```

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.Spinner;

```

Po importu knihoven následuje deklarace třídy:

```
public class MainActivity extends AppCompatActivity { }
```

V těle třídy je místo pro metody a proměnné potřebné k chodu aplikace. Android Studio automaticky vytvoří metodu `onCreate()`, která udává, co se stane okamžitě po spuštění aplikace. Nejprve zjistí, jestli je uložený stav z předchozího spuštění aplikace a poté načte XML soubor, přiřazený k hlavní aktivitě.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

```

V představované aplikaci jsou nejdříve vytvořeny proměnné typu GridLayout a Spinner. Definovány jsou pro celou třídu, protože jsou využity ve více metodách.

Následuje již uváděná metoda onCreate(). V této metodě se provedou úkony popisované již dříve. Komponenta Spinner, která byla vytvořena hned ze začátku projektu, se přiřadí do proměnné „s“. Dále je vytvořen adaptér „a“, který načte pole jednotlivých položek vysouvacího seznamu ze souboru strings.xml a je spojen se Spinnerem. Stejně jako Spinner se komponenta GridLayout přiřadí do proměnné „g“ a nastaví se tomuto rozložení počet řádků a sloupců.

```
private static GridLayout g;
private static Spinner s;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    s = (Spinner) findViewById(R.id.se_cislo);
    ArrayAdapter a = ArrayAdapter.createFromResource(this, R.array.cisla,
        R.layout.support_simple_spinner_dropdown_item);
    s.setAdapter(a);
    g = (GridLayout) findViewById(R.id.bt_layout);
    g.setRowCount(3);
    g.setColumnCount(3);
}
```

Projekt pokračuje metodou onClick() obsluhující tlačítko OK. Metoda nejdříve vymaže veškeré komponenty nacházející se v kontejneru GridLayout (g). Deklarována je proměnná „a“ typu Button (tlačítko) a proměnná typu String (řetězec) pro uložení hodnoty vybrané ze Spinneru, která následně musí být přetypována na celé číslo (Integer). Je to z toho důvodu, aby mohl následující cyklus pracovat s touto hodnotou. Cyklus proběhne tolikrát, kolik je zadáno ve Spinneru a tolikrát se také vytvoří tlačítko pomocí metody vytvorTlacitko(), která je popsána dále. Tlačítko je přidáno do uživatelského rozhraní.

Metoda onClick():

```
public void onClick(View view) {
    g.removeAllViews();
    Button a;
    String vybrano = s.getSelectedItem().toString();
    int cislo = Integer.parseInt(vybrano);
    for(int i = 1; i <= cislo; i++){
        a = vytvorTlacitko(i);
        g.addView(a);
    }
}
```

Metoda `vytvorTlacitko()` má za úkol vytvořit tlačítko a k tomuto tlačítku implicitně přiřadit metodu, která ho bude obsluhovat. Vrací hodnotu typu tlačítko, která se využije při další práci s ním. Tlačítku jsou nejdříve nastaveny atributy, jako je ID, text k zobrazení, minimální šířka a metoda, pro odposlouchávání akce na tomto tlačítku. Pokud bude tlačítko stisknuto, provede umocnění jeho hodnoty a vytvoří se dialog, který oznámí konečnou hodnotu. Dialog musí být potvrzen tlačítkem OK, jinak ho zrušit nelze.

Metoda `vytvorTlacitko()`:

```
public Button vytvorTlacitko(final int id){
    final Button b = new Button(this);
    b.setText(Integer.toString(id));
    b.setId(id);
    b.setMinimumWidth(150);
    b.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            int vysledek = id * id;
            AlertDialog.Builder d = new AlertDialog.Builder(MainActivity.this);
            d.setMessage("Druhá mocnina čísla " + Integer.toString(id) + " je: " +
                Integer.toString(vysledek)).setCancelable(false).setPositiveButton("
                OK", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which){
                        dialog.cancel();
                    }
                });
            AlertDialog ad = d.create();
            ad.setTitle("Mocniny");
            ad.show();
        }
    });
    return b;
}
```

Poslední metoda `smazat()` obsluhuje tlačítko SMAZAT VŠE a má za úkol pouze vymazat všechny komponenty z kontejneru `GridLayout`.

```
public void smazat(View view) {
    g.removeAllViews();
}
```

ZÁVĚR

Operační systém Android je v současné době nejrozšířenějším operačním systémem pro mobilní platformy a začíná pronikat například i do automobilového průmyslu. Jeho hlavní politikou je volná distribuce a dostupnost. Můžeme tedy předpokládat, že jeho rozšířenost bude stále více stoupat a proto je tedy možné vidět v programování aplikací pro tento systém velký potenciál.

V této bakalářské práci je popsána historie systému Android, jak vznikal, stejně jako je popsána historie všech jeho verzí od nejstarších až po ty současné. Ve vývoji je možné vidět, že vývojáři mají stále nové nápady jak tento systém vylepšit a tím pádem se otevírají nové možnosti i vývojářům aplikací pro tento systém.

Představeny byly také jazyky a volně dostupná vývojová prostředí, jež je možné využít pro vývoj aplikací. Možností je velké množství a každý si může najít to, co bude vyhovovat jeho potřebám.

V další části práce je vysvětleno, jak Android aplikace funguje. Pochopení této problematiky je důležité, protože to velice úzce souvisí s vyvíjením těchto aplikací. Aplikace jako taková, není jeden list kódu, který zajistí vše. Je to souhrn vzájemně propojených komponent, které musí spolupracovat, aby mohla aplikace fungovat jako celek.

V poslední části je představeno Android Studio, asi nejvýznamnější vývojové prostředí, které je momentálně dostupné. Vytváří ho ti samí lidé, kteří se podílejí na vytváření samotného operačního systému Android. Existuje také obrovská komunita, která se vývoji aplikací v tomto prostředí věnuje. Práce je zaměřena právě na představení Android Studia a ukázkou základních úkonů, které je možné v tomto vývojovém prostředí dělat.

RESUMÉ

Práce je zaměřena na softwarové vybavení chytrých telefonů, které fungují na operačním systému Android a představení základních možností, jak vyvinout aplikaci pro tuto platformu.

Nejdříve je představen samotný systém Android a jeho historie stejně jako jeho postavení v současnosti.

Následuje část prezentující základní teoretické znalosti týkající se vývoje aplikací pro Android. Představeny jsou jak volně dostupná vývojová prostředí, tak i jazyky, které se dají k vývoji aplikací použít.

Další kapitola seznamuje s aplikací z pohledu funkčnosti a komponent, které využívá.

Poslední část představuje vývojové prostředí Android Studio. Představeno je od úplných základů, jako je instalace, přes nastavení emulátoru až po vytvoření jednoduché aplikace.

Cílem práce je seznámit potenciálního vývojáře se základními znalostmi, které jsou potřebné k vývoji Android aplikací.

SUMMARY

This bachelor thesis is aimed on software equipment of smartphones, which are running on operating system Android and to introduce basic possibilities, how to develop application for this platform.

At first there is introduced operating system Android and his history same as it is from nowadays.

Next part presents basic theoretical knowledge about developing Android applications. There are introduced free developing tools same as are programming languages.

Following chapter is aimed on application itself from the view of functionality and components, which are being used.

Last part of this thesis presents Android Studio, one of the developing tools. It is presented from total bases like instalation, threw setting up emulátor to creating simple application.

Aim of this thesis is to introduce basic knowledge about developing Android applications to potential developer.

SEZNAM LITERATURY

1. Say Sayonara to Google Apps. *XDA-Developers Android Forums*. [Online] 3. 6 2013. [Citace: 8. 3 2016.] <http://www.xda-developers.com/say-sayonara-to-google-apps/>.
2. **Marvan, Filip**. Mobilní operační systém Android - Jak to všechno začalo. *Diit.cz - Vybráno z IT*. [Online] 27. 7 2011. [Citace: 8. 3 2016.] <http://diit.cz/clanek/mobilni-operacni-system-android>.
3. **Welch, Chris**. Before it took over smartphones, Android was originally destined for cameras. *The Verge*. [Online] 16. 4 2013. [Citace: 8. 3 2016.] <http://www.theverge.com/2013/4/16/4230468/android-originally-designed-for-cameras-before-smartphones>.
4. **Manjoo, Farhad**. A Murky Road Ahead for Android, Despite Market Dominance. *The New York Times - Breaking News, World News & Multimedia*. [Online] 27. 5 2015. [Citace: 8. 3 2016.] http://www.nytimes.com/2015/05/28/technology/personaltech/a-murky-road-ahead-for-android-despite-market-dominance.html?_r=0.
5. Industry Leaders Announce Open Platform for Mobile Devices. *Open Handset Alliance*. [Online] 5. 11 2011. [Citace: 8. 3 2016.] http://www.openhandsetalliance.com/press_110507.html.
6. **Block, Ryan**. Google is working on a mobile OS, and it's due out shortly. *Engadget / Technology News, Advice and Features*. [Online] 28. 8 2007. [Citace: 8. 3 2016.] <http://www.engadget.com/2007/08/28/google-is-working-on-a-mobile-os-and-its-due-out-shortly/>.
7. **Ionescu, Daniel**. Original Android Prototype Revealed During Google, Oracle Trial. *PCWorld - News, tips and reviews from the experts on PCs, Windows, and more*. [Online] 26. 4 2012. [Citace: 8. 3 2016.] http://www.pcworld.com/article/254539/original_android_prototype_revealed_during_google_oracle_trial.html.
8. **Wilson, Mark**. T-Mobile G1: Full Details of the HTC Dream Android Phone. *Gizmodo - We come from the future*. [Online] 23. 9 2008. [Citace: 8. 3 2016.] <http://gizmodo.com/5053264/t-mobile-g1-full-details-of-the-htc-dream-android-phone>.
9. **Morrill, Dan**. Announcing the Android 1.0 SDK, release 1. *Android Developers Blog*. [Online] 23. 9 2008. [Citace: 8. 3 2016.] <http://android-developers.blogspot.cz/2008/09/announcing-android-10-sdk-release-1.html>.
10. Infographic: The Complete History of Android - Cupcake to Android M. *Android News / Androidheadlines.com*. [Online] 15. 6 2015. [Citace: 9. 3 2016.] <http://www.androidheadlines.com/2015/06/infographic-the-complete-history-of-android-cupcake-to-android-m.html>.
11. *Market share for mobile, browsers, operating systems and search engines / NetMarketShare*. [Online] 2016. [Citace: 2. 4 2016.] <http://marketshare.hitslink.com/>.
12. Dashboards. *Android Developers*. [Online] 4. 4 2016. [Citace: 4. 4 2016.] <http://developer.android.com/about/dashboards/index.html>.
13. **Lacko, Ľuboslav**. *Vývoj aplikací pro Android*. Brno : Computer Press, 2015. ISBN 978-80-251-4347-6.

14. Java SE - Downloads | Oracle Technology Network. *Oracle | Integrated Cloud Applications and Platform Services*. [Online] 2015. [Citace: 4. 4 2016.] <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
15. What is Android SDK? *Techopedia - Where IT and Business Meet*. [Online] 2015. [Citace: 4. 4 2016.] <https://www.techopedia.com/definition/4220/android-sdk>.
16. Java Card Overview. *Oracle | Integrated Cloud Applications and Platform Services*. [Online] 2015. [Citace: 4. 4 2016.] <http://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html>.
17. Java Platform, Micro Edition (Java ME). *Oracle | Integrated Cloud Applications and Platform Services*. [Online] 2015. [Citace: 4. 4 2016.] <http://www.oracle.com/technetwork/java/embedded/javame/index.html>.
18. Java TV. *Oracle | Integrated Cloud Applications and Platform Services*. [Online] 2015. [Citace: 4. 4 2016.] <http://www.oracle.com/technetwork/java/embedded/javame/java-tv/overview/index.html>.
19. Features of Java - Javatpoint. *Javatpoint - A Solution of all Technology*. [Online] [Citace: 4. 4 2016.] <http://www.javatpoint.com/features-of-java>.
20. XML Essentials - W3C. *World Wide Web Consortium (W3C)*. [Online] 2015. [Citace: 5. 4 2016.] <https://www.w3.org/standards/xml/core>.
21. XML Introduction. *W3Schools Online Web Tutorials*. [Online] 2015. [Citace: 5. 4 2016.] http://www.w3schools.com/xml/xml_what_is.asp.
22. **Němec, Jan.** C/C++ (1) - Úvod -Linux Software. *Linux Software*. [Online] 31. 8 2004. [Citace: 5. duben 2016.] http://www.linuxsoft.cz/article.php?id_article=370.
23. **Thomas, John.** Why C++ is the perfect choice for modern app development. *Technology News and Analysis - BetaNews*. [Online] 2014. [Citace: 5. 4 2016.] <http://betanews.com/2014/07/22/why-c-is-the-perfect-choice-for-modern-app-development/>.
24. **Cunningham, Andrew.** Android Studio 1.3 supports Android's C/C++ NDK. *Ars Technica*. [Online] 28. 5 2015. [Citace: 5. duben 2016.] <http://arstechnica.com/gadgets/2015/05/good-news-for-game-devs-android-studio-1-3-supports-androids-cc-ndk/>.
25. Lua: about. *The Programming Language Lua*. [Online] 4. 2 2016. [Citace: 5. 4 2016.] <https://www.lua.org/about.html>.
26. Introduction to HTML. *W3Schools Online Web Tutorials*. [Online] 2015. [Citace: 5. 4 2016.] http://www.w3schools.com/html/html_intro.asp.
27. CSS Introduction. *W3Schools Online Web Tutorials*. [Online] 2015. [Citace: 5. 4 2016.] http://www.w3schools.com/css/css_intro.asp.
28. Android Studio Overview. *Android Developers*. [Online] 2015. [Citace: 5. 4 2016.] <http://developer.android.com/tools/studio/index.html>.
29. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2015. [Citace: 5. 4 2016.] <http://www.stencyl.com/>.
30. About Us. *MIT App Inventor | Explore MIT App Inventor*. [Online] 2012. [Citace: 5. 4 2016.] <http://appinventor.mit.edu/explore/about-us.html>.

31. WHY CHOOSE CORONA SDK? *Cross-Platform Mobile App Development For IOS, Android - Corona Labs*. [Online] 2016. [Citace: 5. 4 2016.] <https://coronalabs.com/why-choose-corona-sdk/>.
32. **Vávrů, Jiří a Ujbányai, Miroslav**. *Programujeme pro Android*. Praha : Grada Publishing, a.s., 2013. ISBN 978-80-247-4863-4.
33. **Murphy, Mark L**. *Android 2 Průvodce programováním mobilních aplikací*. Brno : Computer Press, a.s., 2011. ISBN 978-80-251-3194-7.
34. Install Android Studio. *Android Developers*. [Online] 2016. [Citace: 8. 4 2016.] <http://developer.android.com/sdk/installing/index.html>.
35. **Allen, Grant**. *Android 4 Průvodce programováním mobilních aplikací*. Brno : Computer Press, 2013. ISBN 978-80-251-3782-6.
36. 2014-12-17-08.06.46.png (800×1280). *Svět Androida - Recenze, novinky, denní zpravodajství ze světa open source operačního systému Android od společnosti Google pro chytré mobilní telefony*. [Online] 17. 12 2014. [Citace: 8. 4 2016.] <http://www.svetandroida.cz/media/2014/12/2014-12-17-08.06.46.png>.

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Graf 1: Podíl mobilních platform na trhu (zdroj: vlastní).....	14
Graf 2: Zastoupení jednotlivých verzí systému Android (zdroj: vlastní).....	15
Obrázek 3: Toast oznámení (zdroj: vlastní)	29
Obrázek 4: Oznámení ve stavovém řádku 1 (zdroj: vlastní)	29
Obrázek 5: Oznámení ve stavovém řádku 2 (zdroj: vlastní)	29
Obrázek 6: Oznámení formou dialogového okna (zdroj: vlastní)	30
Obrázek 7: Nesmyslná oprávnění aplikace (zdroj: vlastní).....	32
Obrázek 8: Úvodní obrazovka po spuštění Android Studia (zdroj: vlastní)	36
Obrázek 9: SDK Manager - SDK Platforms (zdroj: vlastní).....	38
Obrázek 10: SDK Manager - SDK Tools (zdroj: vlastní)	39
Obrázek 11: SDK Manager - SDK Update Sites (zdroj: vlastní)	40
Obrázek 12: Průvodce novým projektem - Specifikace parametrů (zdroj: vlastní)	41
Obrázek 13: Průvodce novým projektem – Výběr API a zařízení (zdroj: vlastní).....	42
Obrázek 14: Průvodce novým projektem - Stahování potřebných nástrojů (zdroj: vlastní).....	42
Obrázek 15: Průvodce novým projektem - Výběr aktivity (zdroj: vlastní).....	43
Obrázek 16: Průvodce novým projektem - Pojmenování hlavní aktivity (zdroj: vlastní)...	44
Obrázek 17: Android Studio 1 (zdroj: vlastní)	45
Obrázek 18: Android Studio 2 (zdroj: vlastní)	45
Obrázek 19: Android Studio 3 (zdroj: vlastní)	46
Obrázek 20: Správce virtuálních zařízení (zdroj: vlastní)	47
Obrázek 21: Průvodce vytvořením virtuálního zařízení 1 (zdroj: vlastní)	48
Obrázek 22: Průvodce vytvořením virtuálního zařízení 2 (zdroj: vlastní)	48
Obrázek 23: Průvodce vytvořením virtuálního zařízení 3 (zdroj: vlastní)	49
Obrázek 24: Ladění USB (zdroj: vlastní)	50
Obrázek 25: Odemčení možností vývojáře (zdroj: (36)).....	50
Obrázek 26: Aplikace po spuštění (zdroj: vlastní)	51
Obrázek 27: Výběr požadovaného čísla (zdroj: vlastní)	52
Obrázek 28: Potvrzení výběru čísla (zdroj: vlastní)	52
Obrázek 29: Stisknutí čísla (zdroj: vlastní)	53
Obrázek 30: Rozložení aplikace (zdroj: vlastní)	54

PŘÍLOHY

Příloha 1 – Zdrojový kód activity_main.xml

Příloha 2 – Zdrojový kód MainActivity.java

Příloha 3 – CD

PŘÍLOHA 1 – ZDROJOVÝ KÓD ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/rl"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.mra.zk.MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAlignment="center"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/nadpis"
        android:id="@+id/tv_vyber"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/se_cislo"
        android:layout_below="@+id/tv_vyber"
        android:layout_alignParentStart="true"
        android:spinnerMode="dropdown"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ok"
        android:minWidth="100dp"
        android:onClick="onClick"
        android:layout_below="@+id/se_cislo"
        android:layout_centerHorizontal="true"
        android:id="@+id/button" />

    <GridLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:layout_marginBottom="50dp"
        android:id="@+id/bt_layout"
        android:layout_below="@+id/button">

    </GridLayout>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/delete"
        android:id="@+id/button_delete"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:onClick="smazat"/>

</RelativeLayout>
```

PŘÍLOHA 2 – ZDROJOVÝ KÓD MainActivity.java

```
package com.example.mra.zk;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.Spinner;

public class MainActivity extends AppCompatActivity {
    private static GridLayout g;
    private static Spinner s;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        s = (Spinner) findViewById(R.id.se_cislo);
        ArrayAdapter a = ArrayAdapter.createFromResource(this, R.array.cisla,
            R.layout.support_simple_spinner_dropdown_item);
        s.setAdapter(a);
        g = (GridLayout) findViewById(R.id.bt_layout);
        g.setRowCount(3);
        g.setColumnCount(3);
    }

    public void onClick(View view) {
        g.removeAllViews();
        Button a;
        String vybrano = s.getSelectedItem().toString();
        int cislo = Integer.parseInt(vybrano);
        for(int i = 1; i <= cislo; i++){
            a = vytvorTlacitko(i);
            g.addView(a);
        }
    }

    public Button vytvorTlacitko(final int id){
        final Button b = new Button(this);
        b.setText(Integer.toString(id));
        b.setId(id);
        b.setMinimumWidth(150);
        b.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                int vysledek = id * id;
                AlertDialog.Builder d = new AlertDialog.Builder(MainActivity.this);
                d.setMessage("Druhá mocnina čísla " + Integer.toString(id) + " je: " +
                    Integer.toString(vysledek)).setCancelable(false).setPositiveButton("
                    OK", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int which){
                            dialog.cancel();
                        }
                    });
                AlertDialog ad = d.create();
                ad.setTitle("Mocniny");
                ad.show();
            }
        });
        return b;
    }

    public void smazat(View view) {
        g.removeAllViews();
    }
}
```


PŘÍLOHA 3 – CD

Na CD se nachází:

- Adresář s aplikací vytvořený Android Studiem (MyApplication2).
- Představené zdrojové kódy vytvořené aplikace (MainActivity.java, activity_main.xml, strings.xml).
- APK soubor k instalaci na zařízení (pgm2p cviceni.apk).
- Vlastní text bakalářské práce ve formátu docx a pdf.