



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Katedra informatiky

**Tvorba mobilních aplikací v Java a HTML5 pro  
Android OS**

**Creation of mobile applications in Java and HTML5  
for Android OS**

Bakalářská práce

**Vedoucí práce:** PaedDr. Petr Pexa, Ph.D.

**Vypracoval:** Ondřej Plucar

České Budějovice 2016

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH  
Fakulta pedagogická  
Akademický rok: 2012/2013

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej PLUCAR**  
Osobní číslo: **P11062**  
Studijní program: **B7507 Specializace v pedagogice**  
Studijní obor: **Informační technologie a e-learning**  
Název tématu: **Tvorba mobilních aplikací v Java a HTML5 pro Android OS**  
Zadávající katedra: **Katedra informatiky**

### Zásady pro vypracování:

Cílem bakalářské práce bude rozbor tvorby aplikací pro mobilní zařízení s Android OS a provedeno porovnání aktuální technologie Java s perspektivní variantou pomocí HTML5/CSS/JavaScript. Výstupem práce bude sada praktických příkladů a konkrétní aplikace katedry informatiky PF JU pro Android OS včetně mobilní verze jejího webu v HTML5 a otestována její funkčnost v aktuálních verzích mobilních operačních systémů Windows Phone, Windows 8, Android OS a iOS. V rámci práce bude také proveden průzkum zaměřený na rozšíření této technologie mezi odbornou veřejností v České republice.

## Příloha zadání bakalářské práce

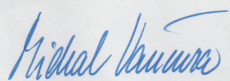
### Seznam odborné literatury:

1. UJBÁNYAI, Miroslav. Programujeme pro Android. Vyd. 1. Praha: Grada, 2012, 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.
2. MURPHY, Mark L. Android 2: průvodce programováním mobilních aplikací. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.
3. GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. HTML5 a CSS3 pro webové designéry. Vyd. 1. Brno: Zoner Press, 2011, 286 s. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.
4. Svetandroida. Svetandroida [online]. 2010 [cit. 2013-03-10]. Dostupné z: <http://www.svetandroida.cz/>
5. HTML5: Tutorial. DUNCAN AITKEN. HTML5: Tutorial [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.html-5-tutorial.com/>
6. ŠRAJER, Michal. Mobile DevCamp 2012: první všestranná mobilní vývojářská konference v ČR. Mobile DevCamp 2012[online]. 2012, č. 2012, s. 1 [cit. 2013-03-10]. Dostupné z: <http://www.svetandroida.cz/mobile-devcamp-2012-prvni-vsestranna-mobilni-vyvojarska-konference-v-cr-201204>
7. KYPTA, Tomáš. Vyvíjíme pro Android. Vyvíjíme pro Android [online]. 2011, č. 2011 [cit. 2013-03-10]. Dostupné z: <http://www.svetandroida.cz/vyvijime-pro-android-1-uvod-201103>
8. Tiggzi: Cloud-based Mobile App Platform. EXADEL INC. Tiggzi [online]. 2012 [cit. 2013-03-10]. Dostupné z: <https://appbuilder.tiggzi.com/projects>
9. Tiggzi: Tutorials. TIGGZI. Tiggzi: Tutorials [online]. 2012 [cit. 2013-03-10]. Dostupné z: <http://blog.tiggzi.com/tag/tutorials-2/>

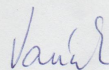
Rozsah grafických prací: **CD ROM**  
Rozsah pracovní zprávy: **40**  
Forma zpracování bakalářské práce: **tištěná**  
Seznam odborné literatury: **viz příloha**

Vedoucí bakalářské práce: **PaedDr. Petr Pexa, Ph.D.**  
Katedra informatiky

Datum zadání bakalářské práce: **16. dubna 2013**  
Termín odevzdání bakalářské práce: **30. dubna 2014**



Mgr. Michal Vančura, Ph.D.  
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 16. dubna 2013

## Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 30. prosince 2015

Ondřej Plucar

## **Abstrakt**

Cílem bakalářské práce bude rozbor tvorby aplikací pro mobilní zařízení s Android OS a provedeno porovnání aktuální technologie Java s perspektivní variantou pomocí HTML5/CSS/JavaScript. Výstupem práce bude sada praktických příkladů a konkrétní aplikace katedry informatiky PF JU pro Android OS včetně mobilní verze jejího webu v HTML5 a otestována její funkčnost v aktuálních verzích mobilních operačních systémů Windows Phone, Windows 8, Android OS a iOS. V rámci práce bude také proveden průzkum zaměřený na rozšíření této technologie mezi odbornou veřejností v České republice.

## **Klíčová slova**

Android, google, play, java, eclipse, studio, mobilní, aplikace, operační, systém, framework, struktura, emulátor

## **Abstract**

The goal of my bachelor's degree paper is an analysis of the production of applications for mobile phone appliances with Android OS and a comparison of the recent Java technology with a perspective variant which will be made with the help of HTML5/CSS/JavaScript. A series of practical examples and concrete applications for Android including a mobile version of its webpage in HTML5 will be the result of my paper for the Department of Informatics (Faculty of Education, University of South Bohemia). Its functionality will be tested in recent versions of mobile operating systems such as Windows Phone, Windows 8, Android OS and iOS. An investigation into the enlargement of this technology among the expert community in the Czech Republic will also be conducted as part of my work

## **Keywords**

Android, google, play, java, eclipse, studio, mobile, application, operating, system, framework, structure, emulator

## Poděkování

Rád bych touto formou poděkoval vedoucímu mé závěrečné práce panu PaedDr. Petru Pexovi, Ph.D. za odborné vedení při zpracování práce, cenné rady, ochotu a trpělivost při vedení mé práce.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>12</b>
1.1	Cíle práce . . . . .	12
1.2	Východiska práce . . . . .	13
1.3	Metody práce . . . . .	13
<b>2</b>	<b>HTML</b>	<b>14</b>
2.1	HTML5 . . . . .	14
<b>3</b>	<b>CSS</b>	<b>16</b>
3.1	CSS3 . . . . .	16
<b>4</b>	<b>Java</b>	<b>17</b>
4.1	Kde ji získat . . . . .	17
4.2	Verze . . . . .	17
4.3	Vizuální vývojová prostředí . . . . .	18
<b>5</b>	<b>Android</b>	<b>19</b>
5.1	Základní části . . . . .	19
5.1.1	Aktivity . . . . .	20
5.1.2	Služby . . . . .	20
5.1.3	Poskytovatelé obsahu . . . . .	20
5.1.4	Záměry . . . . .	21
5.2	Části uživatelské . . . . .	21
5.2.1	Uložiště . . . . .	21
5.2.2	Sít' . . . . .	21
5.2.3	Multimédia . . . . .	22
5.2.4	Služby určení zeměpisné polohy . . . . .	22
5.2.5	Telefonní služby . . . . .	22
5.3	Architektura . . . . .	22
5.3.1	Linux Kernel . . . . .	22
5.3.2	Libraries . . . . .	23
5.3.3	Android Runtime . . . . .	23
5.3.4	Application Framework . . . . .	23
5.3.5	Aplikace . . . . .	24

5.4	Historie verzí Androidu . . . . .	24
5.4.1	Android 1.5 Cupcake . . . . .	24
5.4.2	Android 1.6 Donut . . . . .	25
5.4.3	Android 2.0/2.1 Eclair . . . . .	25
5.4.4	Android 2.2 Froyo . . . . .	25
5.4.5	Android 2.3 Gingerbread . . . . .	25
5.4.6	Android 3.0 Honeycomb . . . . .	26
5.4.7	Android 4.0 IceCream Sandwich . . . . .	26
5.4.8	Android 4.2 Jelly Bean . . . . .	26
5.4.9	Android 4.4 KitKat . . . . .	26
5.4.10	Android 5.0 Lollipop . . . . .	26
<b>6</b>	<b>Požadavky pro vývoj aplikací</b>	<b>28</b>
6.1	Android Studio . . . . .	28
6.1.1	Instalace . . . . .	29
<b>7</b>	<b>Technologie Android Studia</b>	<b>32</b>
7.1	Responsivní IDE . . . . .	32
7.2	Gradle . . . . .	32
7.3	Design aktivit . . . . .	32
7.4	Emulátor . . . . .	33
7.5	Lokalizace aplikací . . . . .	34
<b>8</b>	<b>Praktická část</b>	<b>35</b>
8.1	Sada praktických příkladů . . . . .	35
8.1.1	GPS Lokace . . . . .	35
8.1.2	OCR . . . . .	38
8.1.3	SQLite . . . . .	41
8.2	Nativní aplikace katedry informatiky . . . . .	45
8.2.1	Metodika aplikace . . . . .	48
8.2.2	Testování . . . . .	49
8.2.3	Publikování na Google Play . . . . .	50
8.3	Responzivní web katedry informatiky . . . . .	53
8.3.1	Metodika responzivity . . . . .	54
8.3.2	Testování . . . . .	55

8.4 Porovnání HTML5 a Java . . . . .	55
<b>9 Závěr</b>	<b>57</b>

# 1 Úvod

Před příchodem chytrých mobilních zařízení byla situace webů jednodušší, protože většina lidí používala počítač nebo notebook. Rozměry obrazovek byly víceméně standartní a obsah zobrazení byl na šířku. Pro mobilní telefony a tablety s malými displeji, se musel obsah webu přizpůsobit aktuálně použitému zobrazovacímu zařízení. Pokud by tak nebylo učiněno, klasická webová stránka s fixním designem by nutila uživatele rolovat nejen nahoru a dolů, ale i doprava a doleva. Pokud chceme obsah webu přizpůsobit je možno vytvořit responzivní web.

Cílem bakalářské práce bude rozbor tvorby aplikací pro mobilní zařízení s operačním systémem Android a provedeno porovnání aktuální technologie Java s perspektivní variantou pomocí HTML5/CSS/JavaScript. Výstupem práce bude sada praktických příkladů a konkrétní aplikace katedry informatiky PF JU pro Android OS včetně mobilní verze jejího webu v HTML5 a otestována její funkčnost v aktuálních verzích mobilních operačních systémů Windows Phone, Windows 8, Android OS a iOS. V rámci práce bude také proveden průzkum zaměřený na rozšíření této technologie mezi odbornou veřejností v České republice.

Praktickou část bakalářské práce bude tvořit nativní aplikace, realizovaná pomocí responzivního webu, který je také součástí bakalářské práce. Další částí práce bude představení sady praktických příkladů v objektově orientovaném programovacím jazyce Java.

## 1.1 Cíle práce

Cílem bakalářské práce bude představit operační systém Android, vytváření aplikací pro operační systém od firmy Google Inc. a vyzkoušet tak možnosti vývoje nejen pro mobilní zařízení, ale i pro zařízení s větší úhlopříčkou např. tablety. V práci bude popsána architektura a podpora framework operačního systému Android. Nakonec bude zpracována instalace potřebného programového prostředí pro samotný vývoj nativních aplikací a porovnávání ve vývojovém prostředí s programovacím jazykem Java a HTML5. Nativní aplikace bude publikována na portálu Google Play.

Hlavním cílem bakalářské práce bude hotová nativní aplikace pro katedru

informatiky PF JU v Českých Budějovicích. Aplikace bude dostupná zdarma na aplikačním a webovém portálu Google Play a data budou automaticky aktualizována z webového portálu KIN PF JU.

## 1.2 Východiska práce

Podíl chytrých telefonů v celém světě narůstá a jejich uživatelé se tak začali poptávat stále více po rozmanitějším spektru nativních aplikací, jako jsou mobilní hry, firemní aplikace, praktické aplikace využívající GPS a také všemožné zábavné aplikace pro děti i dospělé. Mobilní aplikace pro Android jsou v současné době nejrozšířenější na světě. Podle statistik z roku 2013 měly podíl na trhu neuvěřitelných 81 procent. V současné době více vývojářů programuje v programovacím jazyce Java. Otázka pro vývojáře zní, jestli začít programovat nativní aplikace v HTML5 nebo zůstat u stávajícího jazyku Java.

## 1.3 Metody práce

Úvodní seznámení s operačním systémem Android a jeho historií. Poté představení architektury operačního systému Android, podpora framework a s ní i vrstvy v systému, které usnadní vývoj aplikací. Práce bude zahrnovat vše potřebné k programování a publikování aplikací potřebné a s ní i několik jednoduchých příkladů o programové struktuře. Nakonec programování v programovacím jazyce Java a HTML5, která bude publikována na aplikační portál Google Play.

## 2 HTML

První verze jazyka HTML<sup>1</sup> se objevila v roce 1991. Její autoři Tim Berners-Lee a Robert Caillau ji vytvořili jako součást projektu WWW<sup>2</sup>, který měl umožnit vědcům komunikaci a sdílení výsledků výzkumu. V roce 1992 přicházejí první prohlížeče WWW stránek. Tyto prohlížeče byly řádkové a zatím neměli grafiku. Návrh HTML 2.0 spolu s grafickým internetovým prohlížečem se poprvé objevují v roce 1993. První grafický internetový prohlížeč dostal název NCSA MOSAC. Nový grafický prohlížeč Netscape byl veřejnosti představen v následujícím roce. Tento WWW prohlížeč byl následníkem celosvětového prvního webového prohlížeče Mosaic.

HTML 2.0 byl oficiálně uveden v roce 1995. Netscape zavádí v tom samém roce neoficiální rozšíření verze 2.0, známé jako HTML 3.0. V květnu 1996 se objevila oficiální specifikace HTML 3.2 (který je kombinací HTML 2.0 a tabulek), Microsoft v daném období vypouští svůj první prohlížeč s názvem MsIE. Další verze 4.0 vychází v červenci 1997 (verze 3.2 vylepšená o rozšíření tabulek a formulářů, dále doplněná o rámy, kaskádové styly a skriptování).

V roce 1999 byla vydána zatím poslední oficiální verze HTML 4.1, s níž pravděpodobně končí celý vývoj HTML. Ale nekončí HTML samo o sobě, stalo se základem pro nové jazyky, např. pro XHTML<sup>3</sup>, který je velice podobný HTML, nebo pro mnohem složitější XML<sup>4</sup>.

V současnosti máme možnost velkého výběru programovacích jazyků, ale všechny mají jistou spojitost s HTML, protože ho ve větší či menší míře používají. [1]

### 2.1 HTML5

HTML5 je současnou verzí značkovacího jazyka HTML sloužícího pro tvorbu webových stránek. Oproti předchozí verze HTML4 přináší podstatné změny, přičemž nejdůležitější patří podpora přehrávání multimédií v prohlížeči a podpora pro aplikace, které fungují i bez připojení k internetu. Charakte-

<sup>1</sup>HyperText Markup Language - značkovací jazyk pro tvorbu webových stránek

<sup>2</sup>World Wide Web - světová rozsáhlá síť

<sup>3</sup>Extensible Hypertext Markup Language - rozšiřitelný hypertextový značkovací jazyk

<sup>4</sup>Extensible Markup Language - rozšiřitelný značkovací jazyk

ristika HTML5 je složena z několika víceméně nezávislých částí. Nové HTML značky sémanticky definující strukturu stránky, perzistentní úložiště formou asociativního pole, relační databáze s podporou transakcí, podpora offline aplikací. Některé z navržených funkcí HTML5 implementovalo rozšíření pro prohlížeče Google Gears. Google se rozhodl vývoj Gears postupně pozastavit ve prospěch nativní podpory HTML5 ze strany prohlížečů.

HTML5 má za cíl dodržovat jednotlivé principy pro přehlednost, jednoduchost a celkovou dokonalost moderních webových stránkách. Specifikace je nestabilní a mění se, ale vždy s ohledem na principy jako jsou: kompatibilita, užitečnost, spolupráce a přístupnost. Účelem HTML5 je založení nových funkcí HTML, CSS, DOM<sup>5</sup>, JavaScriptu, snížení potřeby externích pluginů, jednodušší zpracování chyb než v předchozích verzích, HTML5 nezávislé na řízení a proces vývoje viditelné pro veřejnost. [2]

---

<sup>5</sup>Document Object Model - objektový model dokumentu

## 3 CSS

Jazyk CSS<sup>6</sup> vstoupil do světa tvorby webových stránek počátkem 21. století. Navrhla jej společnost W3C<sup>7</sup>. Autorem prvního návrhu byl Håkon Wium Lie. Aktuálně existují verze CSS1, CSS2 a CSS3. Kaskádové styly jsou jazykem, který je určený k popisování vzhledu elementů jazyků HTML, XHTML a XML. Cílem vzniku CSS bylo oddělení vzhledu webové stránky od jejího významu. Kaskádové styly mají za úkol popisování vzhledu prvků stránky, narozdíl od značkovacích jazyků, které mají obsahu dodávat význam. [3]

### 3.1 CSS3

CSS3 je nejnovější verze od roku 2005. Tvůrci CSS zjistili chyby u verze CSS2 a tak změnili zásadně způsob vzniku třetí verze CSS. Celá specifikace CSS3 je rozdělena do modulů a každý z modulů má vlastní vývojový cyklus, který se skládá z pracovního návrhu, poslední výzvy, kandidát k doporučení, navržený k doporučení a doporučení. Specifikace CSS3 umožňuje tak rychlejší, pružnější a efektivnější změny ve specifikaci jako reakce na potřeby vývojářů webových stránek a zpětnou vazbu od výrobců prohlížečů při implementaci nových vlastností. [2]

---

<sup>6</sup>Cascading Style Sheets - kaskádové styly

<sup>7</sup>World Wide Web Consortium - mezinárodní konsorcium



## 4 Java

Java je objektově orientovaný programovací jazyk, který byl představen veřejnosti v roce 1995 firmou Sun Microsystems vedený J. Goslingem. Původně byl tento jazyk určen k programování pro vestavěné systémy a měl se jmenovat Oak podle stromu, který rostl před oknem pana Goslinga. Vzhledem k tomu, že tento název byl obsazen, rozhodli se jeho tvůrci pro název Java, podle americké varianty espressa. Logem se pochopitelně stal kouřící hrnek kávy.

Základem jejího úspěchu bylo používání v prostředí internetu, zejména vzhledem k možnostem zabezpečení. V současnosti se Java používá v mnoho oblastech. Vedle běžných aplikací se v ní píší applety tzn. krátké programy, které lze vkládat do webových stránek. Dále servlety což jsou programy, které běží na webových serverech a aplikací, které se skládají z několika programů běžících na různých počítačích a komunikujících spolu prostřednictvím sítě. [4]

### 4.1 Kde ji získat

Jazyk Java je k dispozici zdarma. Vše, co potřebujete k programování v Javě, lze stáhnout z internetové adresy <http://java.sun.com>. Při stažení J2SE5 najdete i instrukce pro instalaci. Nástroje pro programování v jazyce Java se obvykle označují zkratkou JDK<sup>8</sup>. [4]

### 4.2 Verze

Od roku 1995 bylo publikováno několik verzí jazyka Java a JDK. Z nich se můžeme nejspíš setkat s verzemi JDK 1.0.2, 1.1.8, 1.2.2, 1.3.2, 1.4.2 a 1.5.x verze JDK 1.0 a 1.1 se příliš nepoužívají. Od JDK 1.2 se hovoří o jazyce Java 2, která přinesla podstatně kvalitnější implementaci některých knihoven např. knihovny kontejnerů. Její součástí je také knihovna JFC/Swing, která obsahuje komponenty pro elegantnější vytváření grafického uživatelského rozhraní aplikací. Od JDK 1.4 se rozlišují různé edice. Vedle standardní edice se můžeme setkat s edicí pro podnikové aplikace a s edicí pro

---

<sup>8</sup>Java Development Kit - produkt Oracle Corporation

mobilní zařízení. [4]

### 4.3 Vizuální vývojová prostředí

Java umožňuje programování v různých vizuálních vývojových nástrojích často označována zkratkou RAD<sup>9</sup>. Velikou výhodou je to, že usnadňuje vytváření okenních aplikací. Když vizuálně sestavíte předem připravený komponent jako základ svého programu, vytvoří se automaticky odpovídající zdrojový kód. Většina vývojových prostředí lze získat zdarma. Jde např. o Borland JBuilder Personal Edition, kde je zapotřebí se zaregistrovat jako člen borlandské vývojářské komunity, abyste prostředí mohli používat za komerčním účelem. Dalším vývojovým prostředím je NetBeans, které lze pro nekomerční účely získat zdarma na internetových stránkách [www.netbeans.com](http://www.netbeans.com). Speciálně pro výuku Javy je určeno prostředí BlueJ, které lze získat na [www.bluej.org](http://www.bluej.org). [4]

---

<sup>9</sup>Rapid Application Development - rychlý vývoj aplikací

## 5 Android

Android je mobilní operační systém založený na jádře Linuxu, který je dostupný jako otevřený software tz. open source. Vyvíjí jej konsorcium Open Handset Alliance, jehož cílem je progresivní rozvoj mobilních technologií, které budou mít výrazně nižší náklady na vývoj a distribuci, a zároveň spotřebitelům přinese inovativní uživatelsky přívětivé prostředí. Při vývoji systému byla brána v úvahu omezení, kterými disponují klasická mobilní zařízení, jako výdrž baterie, menší výkonnost a málo dostupné paměti.

Samotná platforma Android dává k dispozici nejen operační systém s uživatelským prostředím pro koncové uživatele, ale i kompletní řešení nasazení operačního systému (specifikace ovladačů aj.) pro mobilní operátory a výrobce zařízení a v neposlední řadě pro vývojáře aplikací poskytuje efektivní nástroje pro jejich vývoj – Software Development Kit.

Společnost Android Inc. byla založena v Kalifornii v říjnu 2003 Andym Rubinem, Richem Minerem, Nickem Searsem a Chrisem Whitem. Google Inc. v srpnu roku 2005 odkoupil v té době nepříliš známou „startup“ firmu Android Inc. a udělal z ní svoji dceřinou společnost.

Po odkupu společnosti tým Googlu pod vedením Andyho Rubina vyvinul platformu založenou na Linuxovém jádře a v září roku 2007 Google získal několik patentů v oblasti mobilních technologií. Odborná veřejnost začala po akvizici spekulovat, že Google chce tímto krokem vstoupit na trh „chytrých“ mobilních telefonů a chystá vydání vlastního telefonu.

Android v našem světě najdete na každém rohu, ať už v chytrých telefonech, tabletech, televizích a set-top boxech s technologií Google TV. Brzy se však dočkáme operačního systému i v automobilech, systémech letadel a dokonce v robotech.

Hlavní oblastí uplatnění systému Android budou i nadále zařízení s menšími obrazovkami s hardwarovou klávesnicí či bez ní. [5]

### 5.1 Základní části

Při programování aplikací pro počítače má vývojář pro sebe celou vlastní doménu. Sice využívá funkce poskytované operačním systémem, ale ostatní programy, které mohou běžet v počítači současně s jeho aplikací, z větší

části v podstatě ignoruje. A pokud s jinými programy jeho aplikace interaguje, většinou se jedná o komunikaci se systémem MySQL<sup>10</sup> nebo jinými databázemi, na které využívá rozhraní API<sup>11</sup>. Systém Android využívá podobné koncepce, ale jsou jinak trukturovány, aby telefony byly odolnější vůči chybám. Operační systém Android se skládá ze čtyř částí: aktivity, služby, poskytovatelé obsahu a záměry. [5]

### 5.1.1 Aktivity

Stavebními bloky uživatelského rozhraní jsou aktivity. Aktivitu si můžeme představit jako entitu systému Android analogickou k oknu nebo dialgu klasické aplikace pro počítače. Systém Android je navržen tak, aby podporoval množství nenáročných aktivit, takže uživatel může otevírat aktivitu klepnutím a vrátit se do dříve otevřených aktivit pomocí tlačítka Zpět. [5]

### 5.1.2 Služby

Služby jsou oproti tomu navrženy k neustálému provozu, v případě potřeby nezávisle na aktivitách, a podobají se službám nebo démonům jiných operačních systémů. Službu lze použít například ke kontrole dostupných aktualizací RSS<sup>12</sup> nebo k přehrávání hudby na pozadí, dokonce i když jeho řídicí aktivita již neběží. [5]

### 5.1.3 Poskytovatelé obsahu

Poskytovatelé obsahu poskytují úroveň abstrakce pro jakákoliv data uložená v zařízení. Vývojový model aplikací pro Android vás podporuje v tom, abyste svá data kromě svých aplikací zpřístupnili i ostatním aplikacím. Dosáhnout toho můžete právě vytvořením poskytovatele obsahu, který vám umožní zachovat si úplnou kontrolu nad způsobem přístupu k vašim datům. Poskytovatelem obsahu může být cokoliv, počínaje webovými kanály přes místní databázi SQLite<sup>13</sup> až po komplikovanější varianty. [5]

---

<sup>10</sup>MySQL - databázový systém

<sup>11</sup>Application Programming Interface - rozhraní pro programování aplikací

<sup>12</sup>Rich Site Summary - informační kanál

<sup>13</sup>SQLite - relační databázový systém

### 5.1.4 Záměry

Záměry jsou systémové zprávy, které kolují v zařízení a upozorňují aplikace na různé události, počínaje změnami stavu hardwaru přes příchozí data až po události aplikací. Záměry jsou velmi podobné zprávám a událostem jiných operačních systémů. Na záměr můžete nejenom reagovat, ale můžete také vytvořit svůj vlastní a spustit pomocí něho jiné aktivity nebo se nechat informovat o určitých situacích. [5]

## 5.2 Části uživatelské

Části uživatelské jsou části operačního systému Android, se kterými uživatelé mohou pracovat. Tyto části jsou: úložiště, síť, multimédia, služby určení zeměpisné polohy a telefonní služby.

### 5.2.1 Uložiště

Neměnné komponenty aplikace, například ikony nebo soubory nápovědy, se mohou přibalit k aplikaci ve formě datových souborů. Využití najde malý prostor úložiště přímo v zařízení a je do něj ukládána databáze nebo soubory načtených dat nebo dat zadaných uživatelem, se kterými aplikace pracuje. Pokud uživatel používá velkokapacitní úložiště, například SD<sup>14</sup>, můžete zapisovat a číst soubory i v tomto úložišti. [5]

### 5.2.2 Síť

Zařízení Android jsou většinou připravena k připojení k Internetu pomocí některého z přenosových médií. Připojení k Internetu může využít na libovolné úrovni sockety jazyka Java počínaje až po vestavěný widget webového prohlížeče založeného na jádře WebKit<sup>15</sup>, který můžete vložit do své aplikace. [5]

---

<sup>14</sup>Secure Digital - paměťová karta

<sup>15</sup>WebKit - renderovací jádro prohlížeče

### 5.2.3 Multimédia

Zařízení Android má mají schopnost přehrávání a záznamu zvuku a obrazu. Ačkoliv mají jednotlivá zařízení odlišné specifikace příslušného hardwaru, můžete jejich schopnosti snadno zjistit a poté jejich multimediální schopnosti využít tak, jak uznáte za vhodné, ať už k přehrávání hudby, pořizování fotografií pomocí fotoaparátu nebo záznamu zvuku pomocí mikrofону. [5]

### 5.2.4 Služby určení zeměpisné polohy

Zařízení Android často nabízejí přístup k poskytovatelům údajů o zeměpisné poloze, například k systému GPS<sup>16</sup> nebo triangulaci na základě polohy vysílačů, které mohou vaše aplikace informovat o zeměpisné poloze zařízení. Získaná data pak můžete obratem využít k zobrazení map nebo například sledování pohybu zařízení v případě jeho odcizení. [5]

### 5.2.5 Telefonní služby

Protože jsou zařízení Android obvykle telefony, může váš software provádět volání odesílat a přijímat textové zprávy a provádět veškeré další operace, které nabízí moderní telefonní technologie. [5]

## 5.3 Architektura

K vývoji aplikací je potřeba znát základní informace o architektuře operačního systému Android, ve kterém budou aplikace spouštěny. Architekturu Androidu tvoří architektonické vrstvy. [6]

### 5.3.1 Linux Kernel

Základní a nejnižší vrstvou architektury Androidu, je upravené jádro operačního systému Linux. Úpravy se týkají redukce funkcí a přizpůsobení možnostem mobilních zařízení. Jádro slouží k přímé interakci s hardwarem mobilního zařízení, čímž zabezpečuje úplnou abstrakci od hardwaru pro vyšší softwarové vrstvy. Zabezpečuje správu paměti, správu procesorů, základní síťovou vrstvu a ovladače. Řízení procesů umožňuje, aby více procesů běželo

---

<sup>16</sup>Global Positioning System - globální polohovací systém

současně, aniž by se vzájemně ovlivňovaly. Na úrovni jádra je implementované i zabezpečení systému, správa napájení, vstupně.výstupní operace či základní grafika. [6]

### 5.3.2 Libraries

Nad jádrem je situovaná vrstva knihoven, které poskytují přímý přístup aplikací k různým komponentám systému Android. Jsou to nativní knihovny napsané v C/C++. Tvoří mezivstvu mezi různými komponentami vyšších vrstev a linuxovým jádrem. [6]

### 5.3.3 Android Runtime

Android Runtime obsahuje sadu základních knihoven. Každá aplikace pro Android je samostatný proces využívající vlastní instanci virtuálního stroje DVM<sup>17</sup>. Zabezpečuje běh spustitelných souborů s příponou DEX, kde soubory DEX vznikly kompilací z klasických souborů CLASS a JAR. Dalvik je optimalizovaný pro mobilní zařízení, to znamená, že bere v úvahu omezení možnosti napájení, menší paměť a podobně. Současně může běžet více instancí virtuálního stroje. Po spuštění aplikace Java kompilátor přeloží soubory zdrojového kódu vaší aplikace do více binárních souborů Javy. Nástroj s názvem DX transformuje tyto binární javové soubory do jednoho souboru ve formátu DEX. Virtuální stroj Dalvik potom tento soubor začne vykonávat. [6]

### 5.3.4 Application Framework

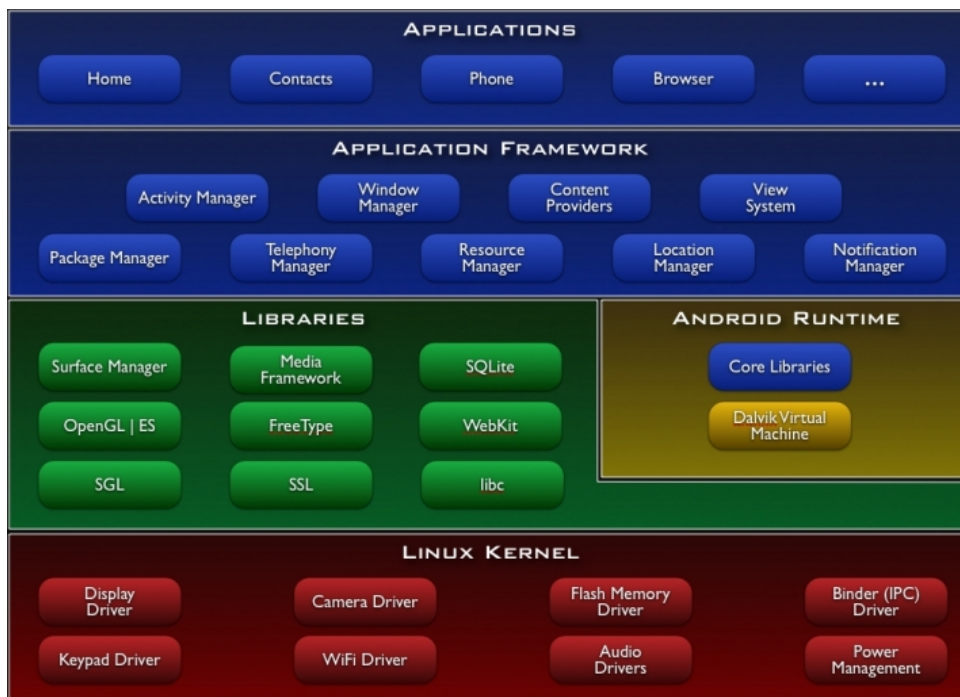
Aplikační framework obsahuje v aplikacích opakovaně použitelný software, například ovládací prvky, ikony a podobně. Framework je napsán v Javě a je to nejdůležitější vrstva pro vývojáře aplikací. Poskytuje aplikacím základní služby systému: Package Manager, Window Manager, View System a Activity Manager. [6]

---

<sup>17</sup>Dalvik Virtual Machine

### 5.3.5 Aplikace

Na nejvyšší úrovni architektury operačního systému Android, jsou aplikace, například program na posílání zpráv, navigaci, kalendář, seznam kontaktů a podobně. [6]



Obrázek 1: Architektura operačního systému Android

## 5.4 Historie verzí Androidu

Jednotlivé verze operačního systému Android mají kromě číselného označení i kódové a tím je název zákusku. [6]

### 5.4.1 Android 1.5 Cupcake

S touto verzí přišla podpora virtuálních klávesnic třetích stran s podporou vlastních slovníků, nahrávání a přehrávání videa ve formátech MPEG-4 a 3GP, možnost kopírovat a vložit obsah přes shcránku, podpora videa YouTube a obrázků Picasa, umíst'ování widgetů na domácí obrazovku a animace



přechodů mezi obrazovkami. V době nástupu verze byly v Android Marketu 3000 aplikací. [6]

#### 5.4.2 Android 1.6 Donut

Verze 1.6 přináší integrované vyhledávání Google, aplikační Market vylepšený o obrázky a hodnocení uživatelů, práce s více soubory, univerzální vyhledavač, vylepšené hlasové vyhledávání, podpora displejů s vyšším rozlišením, bezplatná navigace založená na aplikacích od Google a podpora VPN<sup>18</sup>. [6]

#### 5.4.3 Android 2.0/2.1 Eclair

Tento zákusek přinesl nový design uživatelského prostředí, optimalizaci výkonu, podpora standartů HTML5, Bluetooth 2.1, podpora Microsoft Exchange a Google Maps 3.x, živé tapety a podpora velkého množství rozlišení displejů. V době nástupu verze bylo v Android Marketu 20 000 aplikací. [6]

#### 5.4.4 Android 2.2 Froyo

Šlehačková špička přinesla nové uživatelské prostředí a webový prohlížeč s optimalizací JavaScriptu, podpora Flash 10, možnosti instalovat aplikace i na paměťovou kartu, vylepšené zálohování, USB modem, 3D galerie a sdílení kontaktů přes Bluetooth, možnost vytvořit Wi-Fi hospot, významná optimalizace používání paměti a celkového výkonu, možnost automatických aktualizací aplikací z Android Marketu. V době nástupu verze bylo v Android Marketu 100 000 aplikací. [6]

#### 5.4.5 Android 2.3 Gingerbread

Perníček přinesl vylepšenou správu napájení, podporu více fotoaparátů, komunikaci přes NFC<sup>19</sup>, podporu dalších typů senzorů, vylepšené stahování velkých souborů, podporu internetových hovorů a vylepšenou virtuální klávesnici. [6]

---

<sup>18</sup>Virtual Private Network - virtuální privátní síť

<sup>19</sup>Near Field Communication

#### 5.4.6 Android 3.0 Honeycomb

U této první verze, která je určená pro tablety, je vylepšení uživatelského rozhraní, Action Bar, zjednodušená notifikace, přizpůsobitelná domovská obrazovka, hardwarová akcelerace a podpora USB příslušenství. [6]

#### 5.4.7 Android 4.0 IceCream Sandwich

Verze zmrzlina odstraňuje rozdíly mezi verzemi pro mobilní zařízení a verzemi pro tablety. Novinkou je: aplikace na obrazovce uzamčení, nedávno spuštěné aplikace, vytváření adresářů, možnost zrušit notifikace, zastavení aplikací na pozadí, odemknutí rozpoznáním tváře, převod hlasu na text, nový internetový prohlížeč, podpora vide ve Full HD a integrace sociálních sítí do kontaktů. [6]

#### 5.4.8 Android 4.2 Jelly Bean

Verze s barevnými bonbónkami přinesla GCM<sup>20</sup>, vylepšené notifikace, vylepšenou aktualizaci aplikací, widgety na obrazovce uzamčení, vyhledávání Google Now, možnost přepínání uživatelských účtů, výrazné zrychlení vykreslování obrazu a rozpoznávání hlasu i offline. [6]

#### 5.4.9 Android 4.4 KitKat

Aktuálnější verze 4.4 přinesla vyšší výkon, vylepšenou podporu zařízení s vícejádrovými procesory, rychlejší multitasking, podporu více cloudových služeb, fullscreen režim pro několik aplikací, podporu krokoměru a infračerveného ovládání a inteligentní vypínání nepotřebných procesů na pozadí. [6]

#### 5.4.10 Android 5.0 Lollipop

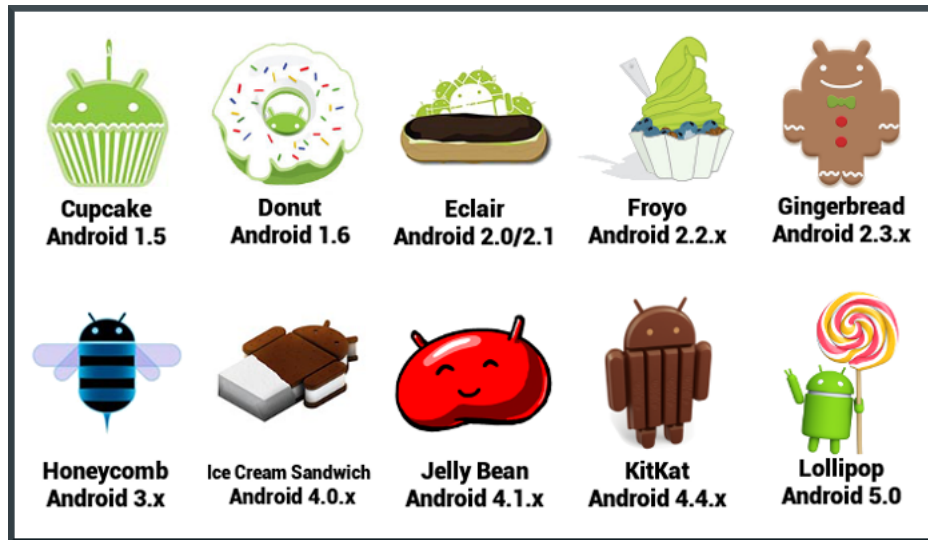
Nejnovější verzí je Android 5.0 Lollipop a jednou z velkých změn je nahrazení dosud používaného virtuálního stroje Dalvik, pod kterým běžely aplikace. Novým enginem je ART<sup>21</sup>. Hlavním přínosem ART je možnost

---

<sup>20</sup>Google Cloud Messaging - podpora uživatelských účtů

<sup>21</sup>Android Runtime

kompilovat část kódu aplikací už při jejich instalaci. To významně urychlí spouštění aplikací a sekundárně prodlouží výdrž baterie, jelikož Dalvik proces kompilace vykonával při každém spuštění. [6]



Obrázek 2: Historie verzí OS Android

## 6 Požadavky pro vývoj aplikací

Vyvíjení nativních aplikací, ať už se jedná o jakýkoliv operační systém, je odlišné a využívá předem definované nástroje. Vývojáři, kteří se rozhodnou programovat nativní aplikace pro chytré mobilní zařízení, musí počítat s náročnějšími požadavky. K vyvíjení nativních aplikací je třeba výkonný hardware počítačového zařízení s libovolným operačním systémem, nainstalované komponenty programovacího jazyka a předem vybrané vývojové prostředí.

Vývoj aplikací pro iOS je založeno na multi-dotykových gestech, tedy na objektech se kterými lze pracovat pouze dotykem prstu. Platforma je založena na operačním systému OS X s vývojovým prostředím IDE Xcode. Vývojovým jazykem pro tuto platformu je Objektiv-C, který je nastavením jazyka C.

Pro vývoj a testování na platformě Windows Phone je potřeba vývojové prostředí IDE Visual Studio s rozšířením Windows Phone Developer Tools. Toto prostředí vyžaduje operační systém Windows a vývojovým OOP jazykem je C#.

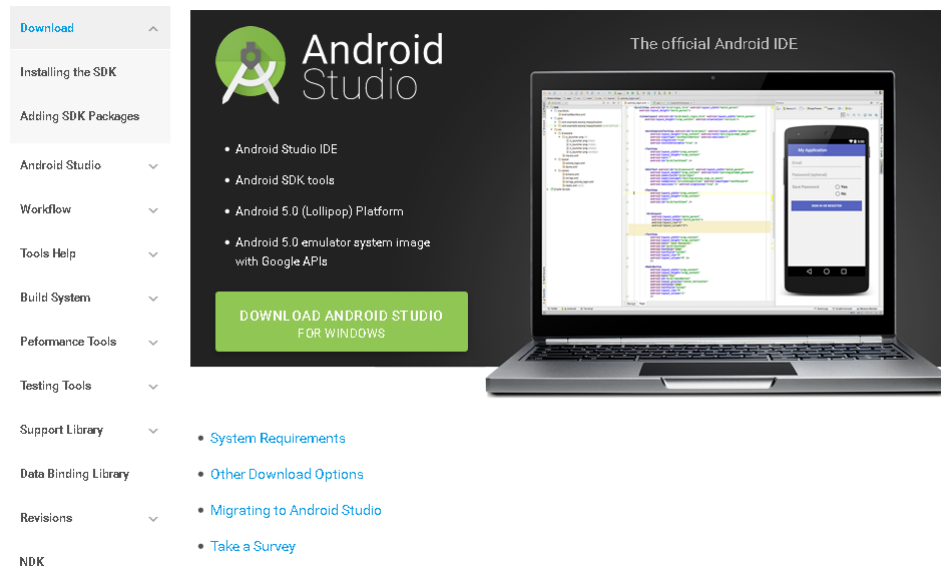
U operačního systému Android je vývoj multiplatformní, kde se může vyvíjet na Windows, OS X a Linuxu. Aplikace se programují v OOP jazyce Java a v současnosti máme nejrozšířenější vývojová prostředí Android Studio a Eclipse. Donedávna bylo základním a doporučeným vývojovým prostředím Eclipse s pluginem ADT, ale v současnosti je instalace komplikovaná, IDE pomalé a celkově nepřehledné. Společnost Google Inc. nedávno přišla s novým vývojovým prostředím Android Studio. [7]

### 6.1 Android Studio

Android Studio je společným dílem Googlu a JetBrains. Přestože většina produktů od JetBrains jsou placené komerční nástroje, Android Studio je zcela zdarma. Je postaveno verzi prostředí IntelliJ IDEA. Díky tomu získává všechny možnosti práce s kódem. [8]

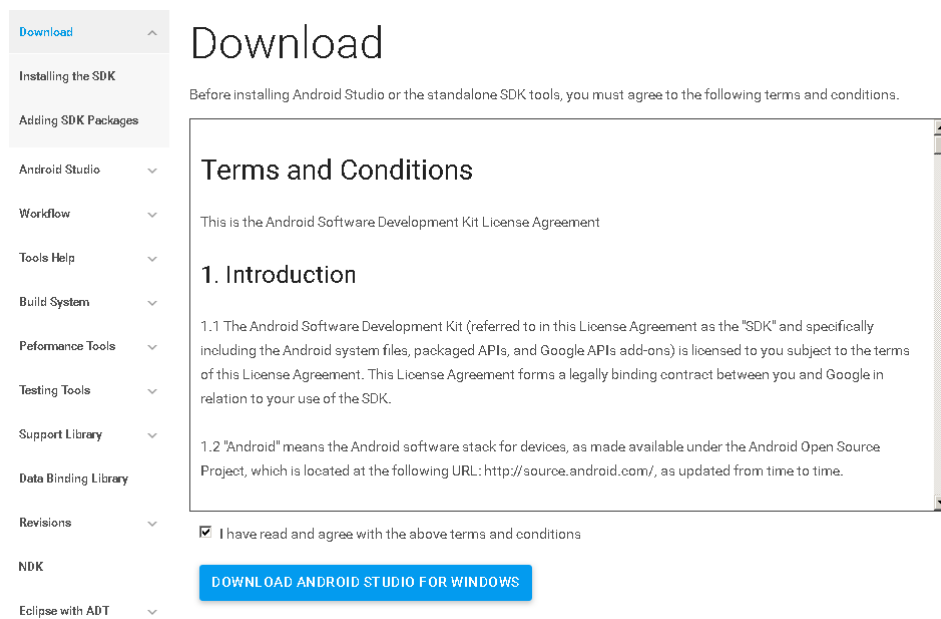
### 6.1.1 Instalace

Instalace je jednoduchá. Na stránkách <https://developer.android.com> se stáhne soubor podle vybraného operačního systému a nainstaluje se do počítačového zařízení. Android Studio se nabízí na třech platformách Windows, Mac a Linux.



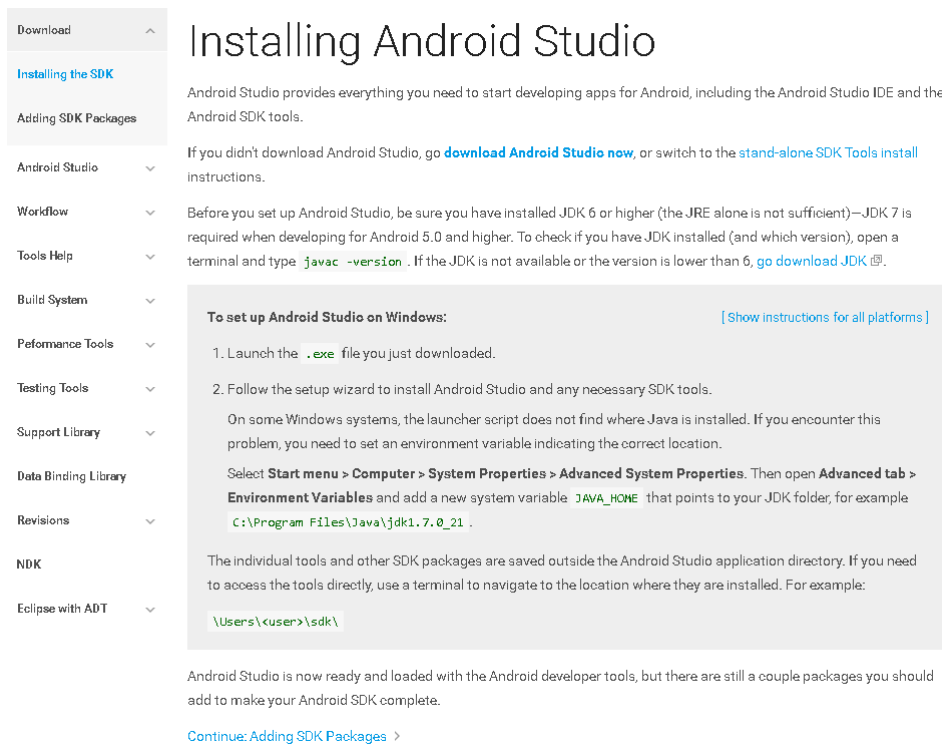
Obrázek 3: Domácí stránka Android - stažení Android Studio

Po kliknutí na stahovat se zobrazí stránka s podmínkami a pravidly. Po přečtení a zaškrtnutí o souhlasu se může soubor stáhnout. Je třeba dbát na to, že velikost instalačního souboru dosahuje k 1GB.



Obrázek 4: Domácí stránka Android - podmínky pro stažení

Dále je vidět stránka, která radí jak postupovat v instalaci souboru a instalaci JDK. Součástí instalace je samotné Android Studio IDE, Android SDK Tools, Android kompilátor a základní emulátory.



The screenshot shows the 'Installing the SDK' section of the Android Studio website. On the left is a navigation menu with items like 'Download', 'Installing the SDK', 'Adding SDK Packages', 'Android Studio', 'Workflow', 'Tools Help', 'Build System', 'Performance Tools', 'Testing Tools', 'Support Library', 'Data Binding Library', 'Revisions', 'NDK', and 'Eclipse with ADT'. The main content area is titled 'Installing the SDK' and contains the following text:

Android Studio provides everything you need to start developing apps for Android, including the Android Studio IDE and the Android SDK tools.

If you didn't download Android Studio, go [download Android Studio now](#), or switch to the [stand-alone SDK Tools install instructions](#).

Before you set up Android Studio, be sure you have installed JDK 6 or higher (the JRE alone is not sufficient)—JDK 7 is required when developing for Android 5.0 and higher. To check if you have JDK installed (and which version), open a terminal and type `javac -version`. If the JDK is not available or the version is lower than 6, go [download JDK](#).

**To set up Android Studio on Windows:** [\[ Show instructions for all platforms \]](#)

1. Launch the `.exe` file you just downloaded.
2. Follow the setup wizard to install Android Studio and any necessary SDK tools.

On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.

Select **Start menu > Computer > System Properties > Advanced System Properties**. Then open **Advanced tab > Environment Variables** and add a new system variable: `JAVA_HOME` that points to your JDK folder, for example `C:\Program Files\Java\jdk1.7.0_21`.

The individual tools and other SDK packages are saved outside the Android Studio application directory. If you need to access the tools directly, use a terminal to navigate to the location where they are installed. For example:

```
\Users\<user>\sdk\
```

Android Studio is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.

[Continue: Adding SDK Packages >](#)

Obrázek 5: Domácí stránka Android - rady při instalaci

## 7 Technologie Android Studia

### 7.1 Responsivní IDE

Celé IDE se přizpůsobuje velikosti okna. Náhledy se zmenšují nebo zvětšují, jsou vedle sebe nebo pod sebou. Paleta prvků se automaticky zobrazí ve více sloupcích podle místa v zobrazení. Má-li vývojář velký monitor, pak může tu plochu plnohodnotně využít. [8]

### 7.2 Gradle

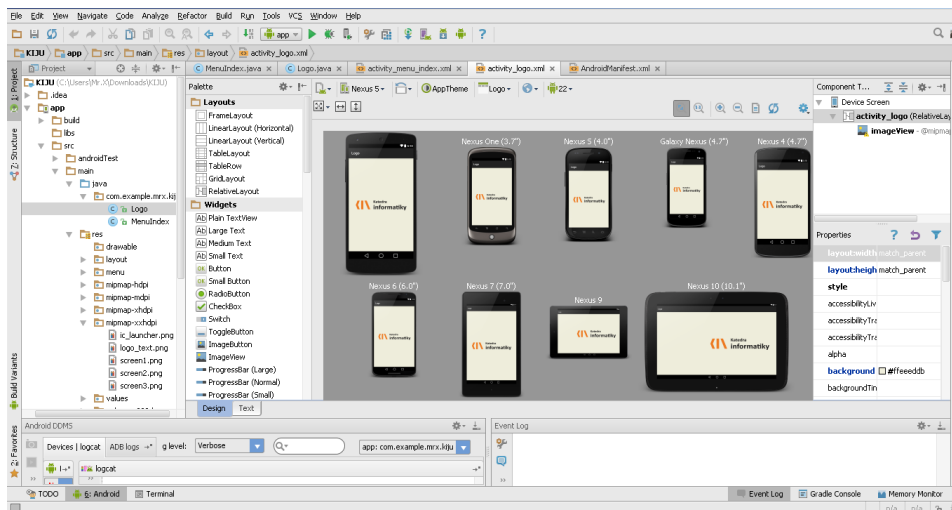
Android Studio používá stavební nástroj Gradle. Ant a Maven, což jsou starší stavební nástroje, které se používali ve vývojovém prostředí Eclipse, Gradle oba nahrazuje. [8]

### 7.3 Design aktivit

Stejně jako ve vývojovém prostředí Eclipse, lze v Android Studiu navrhovat design buď v XML nebo v Design módu. Pokud je vývojář zvyklí psát přímo v XML módu, Android Studio automaticky zobrazuje náhled ve vybraném rozlišení. Díky tomu se nemusíte pořád přepínat do Designu nebo dokonce do emulátoru.

Velkou výhodou je Preview All Screen Sizes, což místo jednoho zobrazí náhledy ve všech možných rozlišeních. [8]



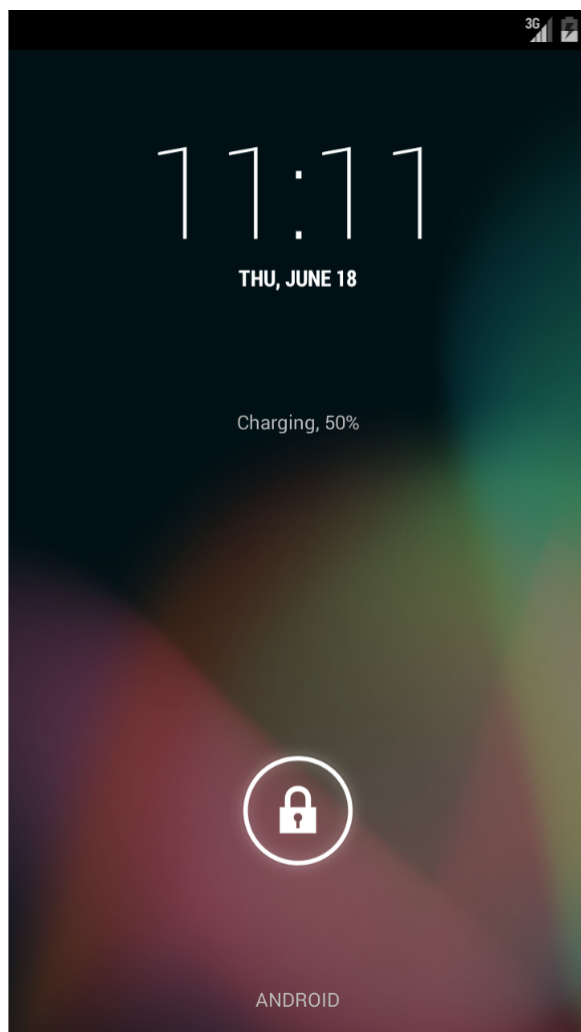


Obrázek 6: Vývojové prostředí Android Studio

## 7.4 Emulátor

Součástí Android Studia jsou emulátory pro Nexus 4, 7 a 10, Nexus S, Nexus One a Galaxy Nexus. Dále je k dispozici i několik bezejmených zařízení s různým rozlišením. Emulátor se může libovolně konfigurovat: verze Android API, velikost RAM, úložiště a SD karty nebo přední či zadní kameru. Kamery se můžou napojit na kameru vašeho počítače.

Emulátory jsou spolehlivé, ale mají jeden velký nedostatek a to, že jsou neskutečně pomalé i na rychlém počítači. Existují jiné varianty testování aplikací nejlépe připojením USB kabelu počítače s mobilním zařízením nebo stažením externích emulátorů, které jsou o něco rychlejší než ty zabudované v Android Studiu.[8]



Obrázek 7: Základní emulátor v Android Studio

## 7.5 Lokalizace aplikací

V Android aplikacích je zvykem mít všechny stringy uložené v resources, což překlad velmi usnadňuje. Android Studio to celé ještě zjednodušuje. Pouze zvolíte Add translation, vyberete jazyk, který chcete přidat, a zobrazí se tabulka, ve které přehledně editujete překlad jednotlivých stringů.[8]

## 8 Praktická část

Praktickou částí moji bakalářské práce bylo vytvořit pár praktických příkladů, které poukazují na nové funkce v objektově orientovaném programovacím jazyce Java. Tyto příklady jsou ukázány v jednotlivých nativních aplikacích pro chytré telefony s operačním systémem Android.

Hlavní částí mé práce bylo vytvořit nativní aplikaci Katedry informatiky pro operační systém Android. Další částí byla tvorba mobilní verze webové stránky Katedry informatiky a testování její funkčnosti v aktuálních verzích mobilních operačních systémů.

Poslední částí mé práce bylo porovnávání aktuální technologie Java s perspektivní variantou pomocí HTML5.

### 8.1 Sada praktických příkladů

Prvním jednodušším příkladem bylo vytvoření nativní aplikace, která ukazuje lokaci zeměpisné výšky a šířky. Tato funkce se využívá všude u aplikací, kde potřebuje znát uživatelskou gps lokaci. Nativní aplikace nese název GPS Lokace a je dostupná na distribučním serveru Google Play.

Dalším příkladem byla o něco složitější aplikace, která pomocí fotoaparátu chytrého mobilního zařízení vyfotila obrázek a pokud se na obrázku zobrazoval nějaký text, tak ho detekoval a přepsal do textového souboru. Tato aplikace se nazývá OCR a je dostupná na distribučním serveru Google Play.

Posledním příkladem byla ukázka aplikace, která se neobejde u každého vývojáře. SQLite databáze, která komunikuje se samotnou aplikací. Tato aplikace se jmenuje Sport Tracker a je dostupná na distribučním serveru Google Play.

#### 8.1.1 GPS Lokace

Velmi jednoduchá aplikace, která po spuštění zobrazí pouze jedno tlačítko uprostřed displeje. Pokud uživatel klikne na tlačítko, zobrazí se zeměpisná výška a šířka podle toho, kde se s zařízením momentálně vyskytuje. Nastavení GPS na telefonu musí být aktivní, jinak se na displeji ukáže dialog a

přesměruje uživatele do nastavení GPS. Tato nativní aplikace je dostupná na distribučním serveru Google Play pod názvem GPS Lokace.

Tento příklad funguje na základě pomocné třídy Location, která představuje údaje o geografické poloze. Vytvořená třída GPSTracker slouží ke zjišťování zapnutých a vypnutých funkcí v zařízení.

```
public Location getLocation(){
    try {
        locationManager = (LocationManager) context.getSystemService(LOCATION_SERVICE);
        isGPSEnable = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        isNetworkEnabled = locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
```

Obrázek 8: Metoda zjišťující lokaci

Pokud je GPS vypnuta, spustí se metoda showSettingsAlert(), která uživatele odkazuje do nastavení GPS.

```
public void showSettingsAlert(){
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(context);
    alertDialog.setTitle("GPS nastavení");
    alertDialog.setMessage("GPS není zapnuta. Chcete do nastavení GPS?");
    alertDialog.setPositiveButton("Nastavení", new DialogInterface.OnClickListener(){
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            context.startActivity(intent);
        }
    });

    alertDialog.setNegativeButton("Zpět", (dialog, which) -> {
        dialog.cancel();
    });
    alertDialog.show();
}
```

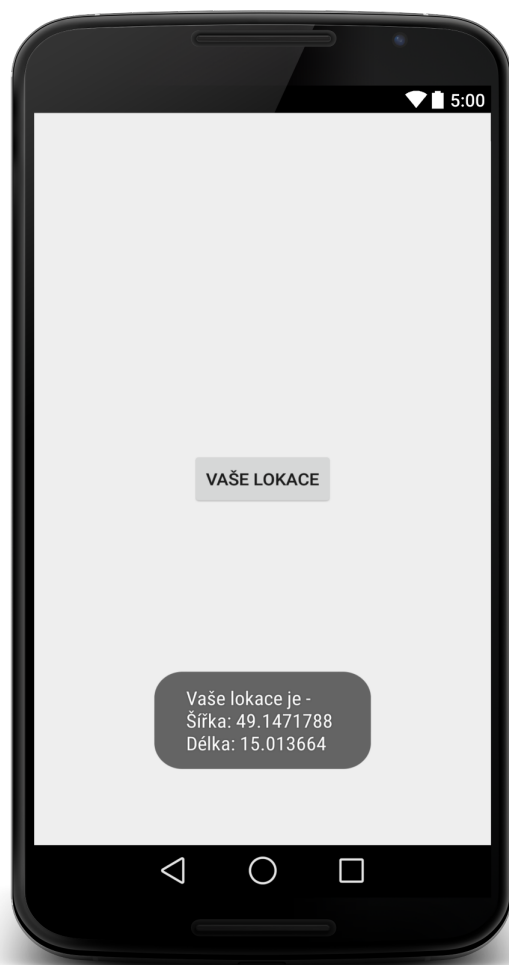
Obrázek 9: Metoda zjišťující zapnuté GPS

Pokud je GPS zapnuta a uživatel klikne na tlačítko „Vaše Lokace“, zobrazí se mu na displeji údaje geografické polohy.

```
@Override
public void onClick(View v){
    gps = new GPSTracker(MainActivity.this);
    if(gps.canGetLocation) {
        double latitude = gps.getLatitude();
        double longitude = gps.getLongitude();
        Toast.makeText(getApplicationContext(), "Vaše lokace je -\nŠířka: " + latitude
            + "\nDélka: " + longitude, Toast.LENGTH_LONG).show();
    }else{
        gps.showSettingsAlert();
    }
}
```

Obrázek 10: Metoda vypisující lokační hodnoty

Ukázka nativní aplikace GPS Lokace.



Obrázek 11: Ukázka aplikace GPS Lokace

### 8.1.2 OCR

O něco složitější praktický příklad je OCR, který umožní optické rozpoznání znaků pomocí detekace a umožňuje digitalizaci tištěných textů, s nimiž pak lze pracovat jako s normálním počítačovým textem. Po spuštění aplikace OCR se zobrazí na displeji fotoaparát, kde si můžete nastavit horizontálně a vertikálně co chcete fotit. Po vyfocení obrázku s textem se text přeformuluje do elektronické podoby.

Pokud chceme detekovat text, musíme přednastavit znaky, které chceme dostat. Existují povolené znaky, které chceme zobrazit a naopak zakázané

znaky, které nechceme zobrazit.

```

public String detectText(Bitmap bitmap) {
    Log.d(TAG, "Initialization of TessBaseApi");
    TessDataManager.initTessTrainedData(context);
    TessBaseAPI tessBaseAPI = new TessBaseAPI();
    String path = TessDataManager.getTesseractFolder();
    Log.d(TAG, "Tess folder: " + path);
    tessBaseAPI.setDebug(true);
    tessBaseAPI.init(path, "ces");
    tessBaseAPI.setVariable(TessBaseAPI.VAR_CHAR_WHITELIST, "1234567890");
    tessBaseAPI.setVariable(TessBaseAPI.VAR_CHAR_BLACKLIST,
        "!@#%&*()_+=[{}|~`.,/;<>?`ê");
    tessBaseAPI.setPageSegMode(TessBaseAPI.OEM_TESSERACT_CUBE_COMBINED);
    Log.d(TAG, "Ended initialization of TessEngine");
    Log.d(TAG, "Running inspection on bitmap");
    tessBaseAPI.setImage(bitmap);
    String inspection = tessBaseAPI.getUTF8Text();
    Log.d(TAG, "Got data: " + inspection);
    tessBaseAPI.end();
    System.gc();
    return inspection;
}

```

Obrázek 12: Kód detekce textu

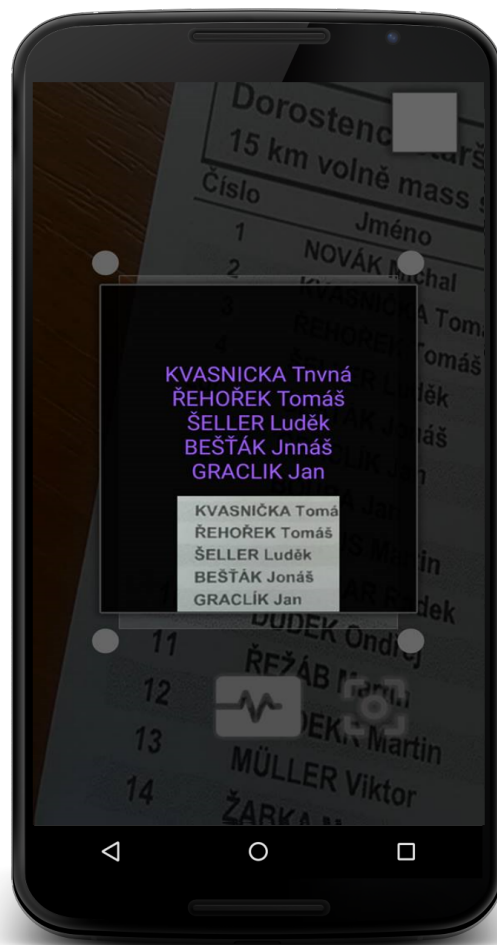
Ještě před vyfocením je třeba vytvořit okénko pro výběr textu na obrázku.



Obrázek 13: Vybraný text

Po vyfocení obrázku s vybraným textem dostaneme text v elektronické podobě.





Obrázek 14: Ukázka dekodovaného textu

### 8.1.3 SQLite

U každé nativní aplikace s větším rozsahem dat by neměla chybět databáze. Vývojář má možnost si vybrat z více typů databází. Jedním z často používaným je SQLite.

Ukázka vytvoření tabulek a řádků v databázi.

```

public static final String DB_DATABASE = "dbRunners";

public static final String DB_TABLE_RACE_CHECKPOINT = "race_checkpoint";
public static final String ATR_RACE_RUNNER_ID = "race_runner_id";
public static final String ATR_TIME = "time";
public static final String ATR_CHECKPOINT = "checkpoint";
public static final String ATR_CHECKPOINT_NAME = "checkpoint_name";

private static final String CREATE_TABLE_RACE_CHECKPOINT = "CREATE TABLE "
    + DB_TABLE_RACE_CHECKPOINT + "(" + ATR_ID + " INTEGER PRIMARY KEY,"
    + ATR_RACE_RUNNER_ID + " LONG, " + ATR_TIME + " LONG,"
    + ATR_CHECKPOINT + " INTEGER, " + ATR_CHECKPOINT_NAME + " TEXT" + ")";

```

Obrázek 15: Ukázka databáze, tabulek a řádků

Abychom mohli vytvořit databázi a její tabulky a řádky, potřebujeme se odkazovat na pomocnou třídu SQLiteOpenHelper, kde použijeme onCreate a onUpgrade. Metoda onCreate nám vytvoří databázi s tabulkami a řádkami a metoda onUpgrade nám aktualizuje tabulky a řádky podle verze databáze, kterou jí přiřadíme.

```

protected static final int DB_VERSION = 179;

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(CREATE_TABLE_RUNNER);
    db.execSQL(CREATE_TABLE_CATEGORY);
    db.execSQL(CREATE_TABLE_RACE);
    db.execSQL(CREATE_TABLE_RACE_RUNNER);
    db.execSQL(CREATE_TABLE_RACE_CHECKPOINT);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + DB_TABLE_RUNNER);
    db.execSQL("DROP TABLE IF EXISTS " + DB_TABLE_CATEGORY);
    db.execSQL("DROP TABLE IF EXISTS " + DB_TABLE_RACE);
    db.execSQL("DROP TABLE IF EXISTS " + DB_TABLE_RACE_RUNNER);
    db.execSQL("DROP TABLE IF EXISTS " + DB_TABLE_RACE_CHECKPOINT);
    onCreate(db);
}

```

Obrázek 16: Vytvoření a aktualizace databáze

Pro každou tabulku je třeba vytvořit Controller, kde se bude používat záznam pro uložení, aktualizace, čtení a mazání dat v databázi. Pokud ukládáme data do databáze, použijeme metodu `saveRaceCheckpoint`.

```
public void saveRaceCheckpoint(RaceCheckpointModel raceCheckpointModel) {
    database = DBHelper.getInstance(context).getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(DBHelper.ATR_RACE_RUNNER_ID, raceCheckpointModel.getRaceRunnerId());
    values.put(DBHelper.ATR_TIME, raceCheckpointModel.getTime());
    values.put(DBHelper.ATR_CHECKPOINT, raceCheckpointModel.getCheckpoint());
    values.put(DBHelper.ATR_CHECKPOINT_NAME, raceCheckpointModel.getCheckpointName());

    raceCheckpointModel.setId(database.insert(DBHelper.DB_TABLE_RACE_CHECKPOINT, null, values));
    listRaceCheckpoint.add(raceCheckpointModel);
}
```

Obrázek 17: Ukázka ukládání dat do databáze

Další důležitá metoda pro čtení dat z databáze je `loadRaceCheckpoint`.

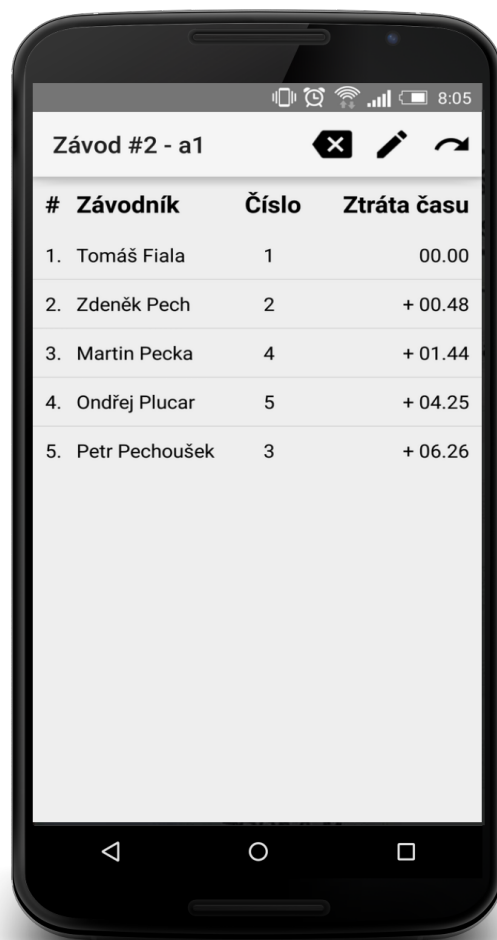
```
private List<RaceCheckpointModel> loadRaceCheckpoint() {
    listRaceCheckpoint = new ArrayList<>();
    // Select All Query
    String selectQueryRunner = "SELECT * FROM " + DBHelper.DB_TABLE_RACE_CHECKPOINT;
    SQLiteDatabase db = DBHelper.getInstance(context).getReadableDatabase();
    Cursor cursorRunner = db.rawQuery(selectQueryRunner, null);

    // looping through all rows and adding to list
    if (cursorRunner.moveToFirst()) {
        do {
            RaceCheckpointModel raceCheckpointModel = new RaceCheckpointModel();
            raceCheckpointModel.setId(Long.parseLong(cursorRunner.getString(0)));
            raceCheckpointModel.setRaceRunnerId(Long.parseLong(cursorRunner.getString(1)));
            raceCheckpointModel.setTime(Long.parseLong(cursorRunner.getString(2)));
            raceCheckpointModel.setCheckpoint(Integer.parseInt(cursorRunner.getString(3)));
            raceCheckpointModel.setCheckpointName(cursorRunner.getString(4));

            // Adding contact to list
            listRaceCheckpoint.add(raceCheckpointModel);
        } while (cursorRunner.moveToNext());
    }
    return listRaceCheckpoint;
}
```

Obrázek 18: Ukázka čtení dat z databáze

Ukázka aplikace Sport Tracker, kde načítá data z SQLite databáze.



The image shows a smartphone screen displaying a table of race results. The title of the screen is "Závod #2 - a1". The table has three columns: "# Závodník", "Číslo", and "Ztráta času". The data rows are as follows:

#	Závodník	Číslo	Ztráta času
1.	Tomáš Fiala	1	00.00
2.	Zdeněk Pech	2	+ 00.48
3.	Martin Pecka	4	+ 01.44
4.	Ondřej Plucar	5	+ 04.25
5.	Petr Pechoušek	3	+ 06.26

Obrázek 19: Ukázka aplikace Sport Tracker

## 8.2 Nativní aplikace katedry informatiky

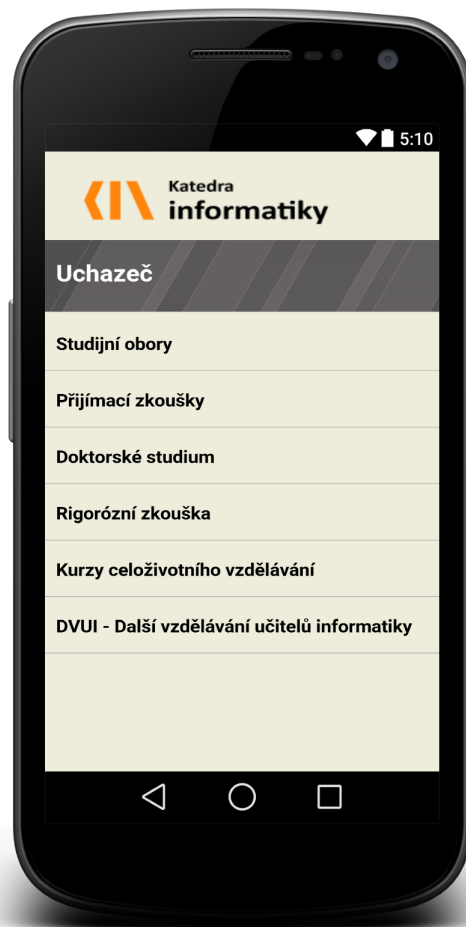
Hlavním cílem mé bakalářské práce bylo vytvořit nativní aplikaci pro katedru informatiky, která usnadní přístup k informacím katedry a poslouží těm uživatelům, kteří chtějí mít dostupné informace katedry přímo na svých mobilních zařízeních. Velkou výhodou je možnost používání aplikace bez připojení k internetu. Každým připojením k síti a spuštěním této aplikace, aktualizujete informace z katedry informatiky na současnou verzi.

Po spuštění aplikace se na displeji zařízení zobrazí úvodní stránka s logem katedry informatiky a po pár vteřinách následuje stránka s hlavním menu, kde má uživatel na výběr ze tří hlavních odkazů Uchazeč, Student a Katedra. Součástí hlavního menu jsou i Aktuality katedry informatiky. Pokaždé když uživatel klikne na logo Katedra informatiky, tak ho to vrátí na aktivitu hlavního menu.



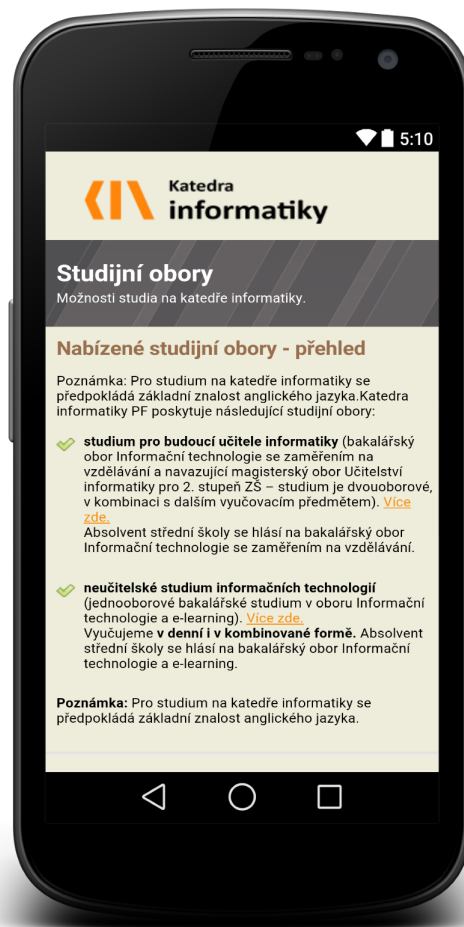
Obrázek 20: Ukázka aplikace katedry informatiky

Pokud uživatel klikne na odkaz Uchazeč, Student nebo Katedra, tak ho přesměruje na rozcestník pododkazů katedry informatiky.



Obrázek 21: Ukázka rozcestníku aplikace katedry informatiky

Cílovým obsahem je odkaz s obsahem informací katedry informatiky.



Obrázek 22: Ukázka cílového obsahu katedry informatiky

### 8.2.1 Metodika aplikace

K tvorbě nativní aplikace je třeba si najít cestu. Je řada různých možností, jak postupovat. První možností bylo propojení nativní aplikace s databází katedry informatiky. Vzhledem k tomu, že databáze katedry informatiky neobsahuje text, bylo zapotřebí najít jiné řešení.

Řešením je třída WebView, která funguje na jádře WebKit. K vykreslení obsahu v aktivitě okna postačí URL adresa. K tomu, aby fungovalo vykreslení obsahu, je třeba povolit oprávnění INTERNET v souboru Manifest.xml.



Bylo třeba obohatit aplikaci přidáním funkce, která ji převede do offline režimu. Pomocí metody `isNetworkAvailable()` bylo možné kontrolovat, jestli má zařízení zapnutý nebo vypnutý internet. Podle toho aplikace načítala přímo z internetu nebo z mezipaměti mobilního zařízení.

```
private boolean isNetworkAvailable() {
    CM = (ConnectivityManager) getSystemService(ActionBarActivity.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetworkInfo = CM.getActiveNetworkInfo();
    return activeNetworkInfo != null && activeNetworkInfo.isConnected();
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(kin.R.layout.activity_main_index);
    web = (WebView) findViewById(kin.R.id.webView);
    web.getSettings().setAppCachePath( getApplicationContext().getCacheDir().getAbsolutePath());
    web.getSettings().setAllowFileAccess( true );
    web.getSettings().setAppCacheEnabled( true );
    web.getSettings().setCacheMode( WebSettings.LOAD_DEFAULT ); // load online by default
    web.setWebViewClient(new myWebClient());
    if ( !isNetworkAvailable() ) { // loading offline
        web.getSettings().setCacheMode( WebSettings.LOAD_CACHE_ELSE_NETWORK );
    }
    web.loadUrl(url);
}
```

Obrázek 23: Kód WebView

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

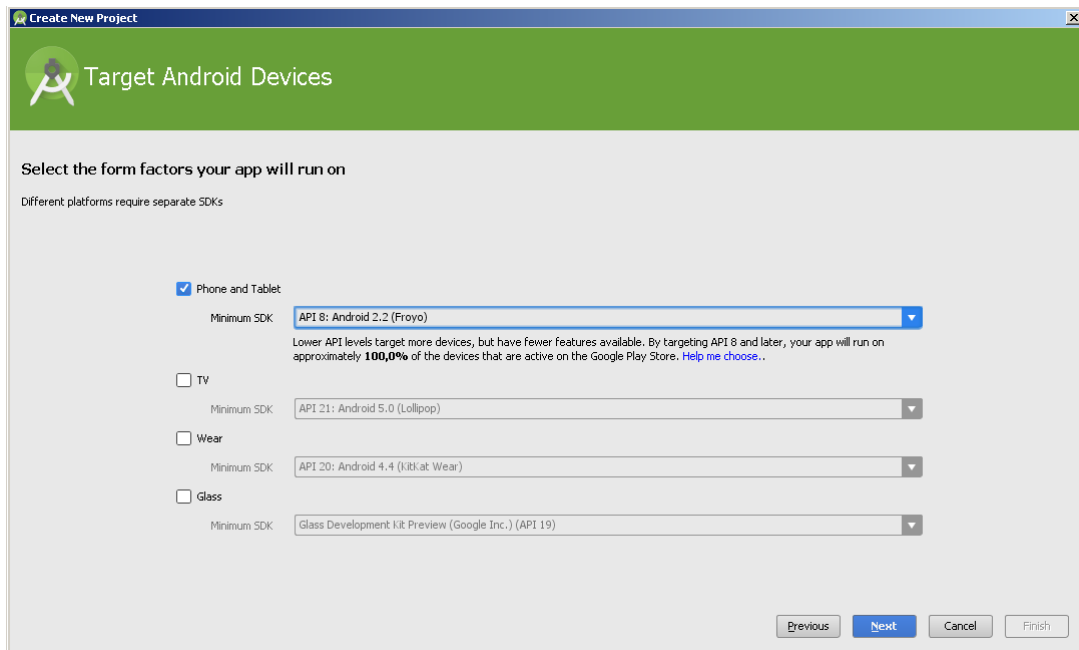
Obrázek 24: Oprávnění v manifest.xml

## 8.2.2 Testování

Testování aplikace katedry informatiky jsem prováděl pomocí USB kabelu připojením k mobilnímu zařízení. Existuje možnost testování přes různé

emulátory, ale rychlejší bylo testování aplikace přímo přes mobilní zařízení.

Testování kompatibility aplikace katedry informatiky s Android verzemi je jednoduchá záležitost. Při vytváření projektu je možnost výběru kompatibility verzí Androidu, kde je výběr od minimální po současnou verzi.



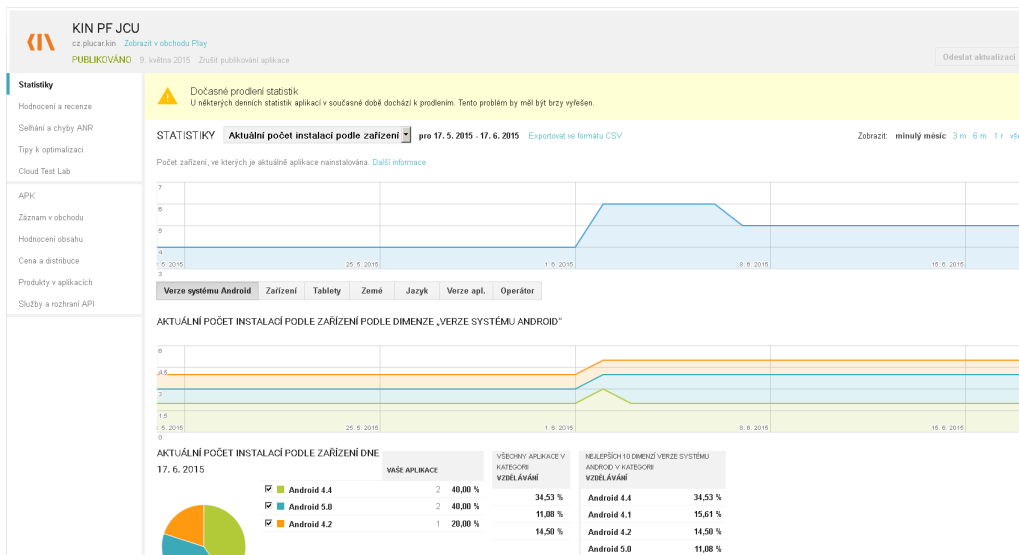
Obrázek 25: Výběr verze při vytváření aplikace

### 8.2.3 Publikování na Google Play

K tomu, aby mohl vývojář publikovat aplikace na portálu Google Play, musí mít zaplacený registrační poplatek vývojáře. Tento poplatek činí aktuálně 25 amerických dolarů, tedy něco přes 500 českých korun a platí pouze pro verze aplikací, které jsou zcela zdarma. Pokud by chtěl vývojář svoje aplikace prodávat, musel by si vytvořit účet obchodníka na Google Wallet. Prodávat aplikace na Google Play není jednoduché. Vývojář musí mít živnostenský list nebo podnikat.

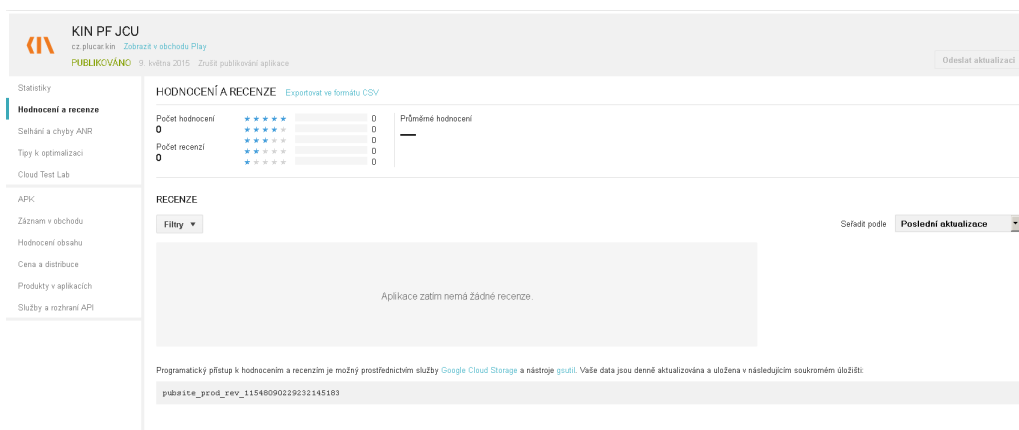
Po vytvoření a publikování na Google Play má vývojář možnost kont-

rolovat údaje o své aplikaci. Údaj o statistikách pojednává počet instalací zařízení.



Obrázek 26: Vývojářská konzole - statistiky

Uživatelé mohou hodnotit aplikaci na serveru Google Play, vývojář má tak přehled přímo ve vývojářské konzoli Google Play.



Obrázek 27: Vývojářská konzole - hodnocení

APK je pro vývojáře nejdůležitějším údajem v konzoli Google Play. Může nahrávat aplikace v alfa, beta nebo v plné verzi. Pokud vývojář bude testovat svojí aplikaci v alfa nebo v beta verzi, musí počítat s tím, že po nahrání na server Google Play, aplikaci nikdo neuvidí. Musí vytvořit skupinu, pozvat uživatele, od kterých chce, aby mu aplikaci otestovali. Dále může kontrolovat podporovaná zařízení nebo vyloučená zařízení.

KIN PF JCU  
cz.pfcarkin Zobrazit v obchodu Play  
PUBLIKOVÁNO 9. května 2015 Zrušit publikování aplikace Odeslat aktualizaci

Statistiky  
Hodnocení a recenze  
Selhání a chyby ANR  
Typy k optimalizaci  
Cloud Test Lab

**APK**  
Záznam v obchodu  
Hodnocení obsahu  
Cena a distribuce  
Produkty v aplikacích  
Služby a rozhraní API

**APK**

PRODUKCE  
Verze  
1

TESTOVÁNÍ BETA VERZE  
Nastavte testování beta verze aplikace

TESTOVÁNÍ ALFA VERZE  
Nastavte testování alfa verze aplikace

**KONFIGURACE PRODUKČNÍ VERZE** Nahraj nový soubor APK k produkci

AKTUÁLNÍ SOUBOR APK publikováno 4. 5. 2015 9:08:10

Podporovaná zařízení  
7983  
Zobrazit seznam

Vyloučená zařízení  
0  
Spravovat vyloučená zařízení

VERZE	NABÍŘENO DNE	STAV	AKCE
1 (1.0)	4. 5. 2015	ve fázi produkce	

Obrázek 28: Vývojářská konzole - APK

Dalším důležitým údajem je Záznam v obchodu, který slouží pro vyplnění nebo opravování údajů o aplikaci. V této sekci vývojář může přidávat grafické podklady.

**KIN PF JCU**  
 CZ: placemim | Zmazať v obchodu Play  
 PUBLIKOVÁNO 9. května 2015 Zrušit publikování aplikace Odeslat aktualizaci

Statistiky  
 Hodnocení a recenze  
 Selhání a chyby ANR  
 Tipy k optimalizaci  
 Cloud Test Lab  
 APK  
**Záznam v obchodu**  
 Experimenty  
 Hodnocení obsahu  
 Cena a distribuce  
 Produkty v aplikacích  
 Služby a rozhraní API

**ZÁZNAM V OBCHODU**  
 Pole označená \* musí být před zveřejněním vyplněna

**PODRBNOSTI PRODUKTU**

Čeština – cs-cz Spravovat příklady

**Název\***  
 Čeština – cs-CZ KIN PF JCU  
 10 z 30 znaků

**Krátký popis\***  
 Čeština – cs-CZ Katedra informatiky Pedagogické fakulty Jihočeské univerzity  
 60 z 80 znaků

**Úplný popis\***  
 Čeština – cs-CZ Aplikace Katedry informatiky Pedagogické fakulty Jihočeské univerzity v Českých Budějovicích  
 92 z 4000 znaků  
 Pokud chcete předjet pozastavení aplikaci, přečtěte si [tipy k vytváření popisů aplikací](#) [vyhovujících zásadám](#).

**GRAFICKÉ PODKLADY**  
 Pokud jste například lokalizovanou grafiku pro jednotlivé jazyky, bude použita grafika výchozího jazyka.  
[Další informace o grafických dílech](#)

**Snímky obrazovek\***  
 Vychází – Čeština – cs-CZ  
 JPEG nebo 24bitový PNG (bez barů alpha) Minimální délka každé strany: 320 pixelů. Maximální délka každé strany: 3840 pixelů.  
 Celkové jsou požadovány alespoň 2 snímky obrazovek. Snímky obrazovek stejného typu může být maximálně 8. Přisazením můžete upravit uspořádání snímků nebo je přesunout mezi typy

Obrázek 29: Vývojářská konzole - Záznam v obchodu

### 8.3 Responzivní web katedry informatiky

Druhým hlavním cílem bylo vytvořit optimální zobrazení webové stránky katedry informatiky pro chytré mobilní zařízení. Postup při tvorbě mobilního webu byl jednoznačný a to přes responzivní zobrazení, protože responzivní webdesign je současně nejpoužívanější technologií u vytváření webových stránek.



Obrázek 30: Responzivní webdesign

### 8.3.1 Metodika responzivity

Postup responzivního webdesignu začíná u meta tagu s názvem Viewport, který zvětšuje obsah stránky, protože mobilní prohlížeče v chytrých telefonech a tabletech běžně zobrazují web zmenšený, aby byl celý vidět.

```
<!-- Viewport zobrazuje do velikosti mobilních zařízení -->
<meta name="viewport" content="width=device-width,
initial-scale=1.0, minimum-scale=1.0">

<!-- Displeje s větším rozlišením než 1024px (notebook, pc) -->
<link rel="stylesheet" type="text/css"
media="screen and (min-device-width: 1025px)" href="css/layout.css" />

<!-- Displeje s menším rozlišením než 1025px (tablet, mobil) -->
<link rel="stylesheet" type="text/css"
media="screen and (max-device-width: 1024px)" href="css/layout_mobile.css" />
```

Obrázek 31: Funkce ViewPort

### 8.3.2 Testování

U responzivního webdesignu je důležité testovat podle aktuálních verzí mobilních prohlížečů. Testování bylo provedeno na fyzickém zařízení všech uvedených prohlížečů. Mezi nejpoužívanějšími prohlížeči jsou Chrome Mobile, Firefox Mobile, Mobile Safari a Opera Mobile. Méně používané, kde aplikace nebyla správně zobrazena, jsou Dolphin Mobile, Maxthon Mobile a Sleipnir Mobile.

Mobilní prohlížeče	Aktuální verze
Chrome Mobile	Ano
Firefox Mobile	Ano
Mobile Safari	Ano
Opera Mobile	Ano
Opera Mini	Ano
Dolphin Mobile	Ne
Maxthon Mobile	Ne
Sleipnir Mobile	Ne

Tabulka 1: Testování aktuálních verzí mobilních prohlížečů

## 8.4 Porovnání HTML5 a Java

Cílem bakalářské práce bylo také porovnat HTML5 s Javou. Porovnání bylo vytvořeno z pohledu začátečníka, který začíná programovat v OOP Java nebo vytvářet ve značkovacím jazyce HTML5. Porovnávala se:

- složitost syntaxe obou jazyků
- složitost funkcí vývojových prostředí

- složitost počátečního nastavení prostředí
- existence překladače
- kvalita vývoje obou jazyků
- hodnota aktualizací
- existence multiplatformnosti
- složitost testování

Vlastnost	Java	HTML5
Jazyk	<b>OOP</b>	<b>Značkovací</b>
Orientace v prostředí	<b>Složitá</b>	<b>Snadná</b>
Počáteční nastavení	<b>Náročné</b>	<b>Jednoduché</b>
Překladač	<b>Ano</b>	<b>Ne</b>
Kvalita vývoje	<b>Vysoká</b>	<b>Nízká</b>
Aktualizace	<b>Velké množství</b>	<b>Malé množství</b>
Multiplatformnost	<b>Ne</b>	<b>Ano</b>
Funkce	<b>Velké množství</b>	<b>Malé množství</b>
Testování	<b>Snadné</b>	<b>Složitě</b>

Tabulka 2: Porovnání HTML5 &amp; Java



## 9 Závěr

Během psaní bakalářské práce jsem prostudoval veškeré dostupné informace o vývojovém prostředí Android Studio a samotném programovacím jazyku Java, která je velmi žádaná v praxi, mezi IT firmami zabývající se tvorbou aplikací nejen pro operační systém Android, ale i pro jiné platformy. Ze začátku bylo důležité si vybrat správné vývojové prostředí. V době, kdy vývojové prostředí Eclipse ustupovalo, byla jasná volba Android Studio, které přinášelo více výhod a možností.

Má bakalářská práce se zabývá tvorbou nativních aplikací pro operační systém Android. Cílem práce bylo představit vývojové prostředí Android Studio vývojářům, kteří chtějí prozkoumat možnosti tvorby nativních aplikací. Práce je rozdělena na teoretickou a praktickou část.

V teoretické části popisují základy značkovacího jazyka HTML, CSS a programovacího jazyka Java. Dále popisují operační systém Android, její architekturu a jednotlivé verze tohoto systému. Dále popisují jaké požadavky jsou nutné splnit při vytváření aplikací pro tuto platformu. Na závěr popisují bez jakých technologií se vývojář neobejde.

V praktické části popisují sadu praktických příkladů a samotnou nativní aplikaci pro Katedru informatiky, která byla mým hlavním cílem bakalářské práce. Popisují také jak nativní aplikaci nahrát na distribuční server Google Play. V další praktické části popisují tvorbu responzivního webu pomocí technologie HTML5 a CSS.

## Použitá literatura a zdroje

- [1] Historie HTML. <i>Garth</i> [online]. 2010 [cit. 2015-06-05]. Dostupné z: <http://www.garth.cz/uvod-do-html/historie-html/>.
- [2] RUSEK, Martin. Výuka HTML5 a CSS3. Výuka HTML5 a CSS3 [online]. 2013 [cit. 2015-06-17]. Dostupné z: <http://html5css3.4fan.cz/html-uvod.php>.
- [3] DOMES, Martin. 333 tipů a triků pro CSS: [sbírka nejužitečnějších návodů pro váš web]. Vyd. 1. Brno: Computer Press, 2009, 272 s. ISBN 978-80-251-2360-7.
- [4] VIRIUS, Miroslav. Java pro zelenáče. 2. upr. vyd. Praha: Neocortex, 2005, 268 s. ISBN 80-86330-17-6.
- [5] ALLEN, Grant. Android 4: průvodce programováním mobilních aplikací. 1. vyd. Brno: Computer Press, 2013, 656 s. ISBN 978-80-251-3782-6.
- [6] LACKO, Ľuboslav. Vývoj aplikací pro Android. 1. vyd. Brno: Computer Press, 2015, 472 s. ISBN 978-80-251-4347-6.
- [7] KLOS, Roman. <i>Multiplatformní vývoj mobilních aplikací pomocí webových technologií</i> [online]. Praha, 2014 [cit. 2015-06-18]. Dostupné z: [https://www.vse.cz/vskp/show\\_file.php?soubor\\_id=1247420](https://www.vse.cz/vskp/show_file.php?soubor_id=1247420). Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Ing. Jarmila Pavlíčková.
- [8] SEMECKÝ, Vojtěch. Android Studio: Nové vývojové prostředí. <i>Android Studio: Nové vývojové prostředí</i> [online]. 2013, (1) [cit. 2015-06-18]. Dostupné z: <http://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/>.
- [9] SEMECKÝ, Vojtěch. Android Studio - nové vývojové prostředí: mobilní vývoj. In: Zdrojak [online]. 2013 [cit. 2014-05-24]. Dostupné z: <http://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/>

## Seznam obrázků

1	<sup>22</sup> . . . . .	24
2	<sup>23</sup> . . . . .	27
3	Domácí stránka Android - stažení Android Studio . . . . .	29
4	Domácí stránka Android - podmínky pro stažení . . . . .	30
5	Domácí stránka Android - rady při instalaci . . . . .	31
6	Vývojové prostředí Android Studio . . . . .	33
7	Základní emulátor v Android Studio . . . . .	34
8	Metoda zjišťující lokaci . . . . .	36
9	Metoda zjišťující zapnuté GPS . . . . .	36
10	Metoda vypisující lokační hodnoty . . . . .	37
11	Ukázka aplikace GPS Lokace . . . . .	38
12	Kód detekce textu . . . . .	39
13	Vybraný text . . . . .	40
14	Ukázka dekódovaného textu . . . . .	41
15	Ukázka databáze, tabulek a řádků . . . . .	42
16	Vytvoření a aktualizace databáze . . . . .	42
17	Ukázka ukládání dat do databáze . . . . .	43
18	Ukázka čtení dat z databáze . . . . .	43
19	Ukázka aplikace Sport Tracker . . . . .	44
20	Ukázka aplikace katedry informatiky . . . . .	46
21	Ukázka rozcestníku aplikace katedry informatiky . . . . .	47
22	Ukázka cílového obsahu katedry informatiky . . . . .	48
23	Kód WebView . . . . .	49
24	Oprávnění v manifest.xml . . . . .	49
25	Výběr verze při vytváření aplikace . . . . .	50
26	Vývojářská konzole - statistiky . . . . .	51
27	Vývojářská konzole - hodnocení . . . . .	51
28	Vývojářská konzole - APK . . . . .	52
29	Vývojářská konzole - Záznam v obchodu . . . . .	53
30	Responzivní webdesign . . . . .	54

<sup>22</sup>Dostupné z: <http://androidmarket.cz/android/jak-vypada-android-uvnitri-aneb-co-je-rom-kernel-bootloader-a-dalsi/>

<sup>23</sup>Dostupné z <http://jaredrummler.com/2014/11/09/lollipop-land/>

31 Funkce ViewPort . . . . . 54

## Seznam tabulek

1	Testování aktuálních verzí mobilních prohlížečů . . . . .	55
2	Porovnání HTML5 & Java . . . . .	56

## **Přílohy**

1. DVD - na přiloženém DVD se nachází plné znění bakalářské práce pod názvem souboru `plucar_bp.pdf`, dále se na DVD nachází zdrojové soubory k dílčím aplikacím a k aplikaci Katedry informatiky.