|  |  |
| --- | --- |
| **Technická univerzita v Košiciach**  **Presný názov fakulty napr. Fakulta elektrotechniky a informatiky** | |
| **Názov práce** | |
| **2010** | **Meno** **Priezvisko1 Priezvisko2 Priezvisko3** |

|  |  |  |
| --- | --- | --- |
| **Technická univerzita v Košiciach**  **Presný názov fakulty napr. Fakulta elektrotechniky a informatiky** | | |
| NÁZOV PRÁCE | | |
| **Diplomová práca alebo Bakalárska práca** | | |
| Študijný program: | | Špecializácia |
| Študijný odbor: | | Číslo a názov odboru (pozri zadávací list) |
| Školiace pracovisko: | | Názov katedry (Skratka katedrySkratka katedry) |
| Školiteľ: | | Titul Meno Priezvisko, Titul |
| Konzultant: | | Titul Meno1 Priezvisko1, Titul  Titul Meno2 Priezvisko2, Titul |
| **Košice 2010** | **Meno** **Priezvisko1 Priezvisko2 Priezvisko3** | |

**Analytický list**

|  |  |
| --- | --- |
| Autor: | MenoMeno Priezvisko1 Priezvisko2 Priezvisko3Priezvisko1 Priezvisko2 Priezvisko3 |
| Názov práce: | Názov práce Názov práce |
| Podnázov práce: |  |
| Jazyk práce: | slovenský, anglický, nemeckýslovenský, anglický, nemecký |
| Typ práce: | Diplomová práca alebo Bakalárska prácaDiplomová práca alebo Bakalárska práca |
| Počet strán: | 3 |
| Akademický titul: | Bakalár, Magister, Magister umenia, Inžinier, Inžinier architektBakalár, Magister, Magister umenia, Inžinier, Inžinier architekt |
| Univerzita: | Technická univerzita v KošiciachTechnická univerzita v Košiciach |
| Fakulta: | Presný názov fakulty napr. Fakulta elektrotechniky a informatikyPresný názov fakulty napr. Fakulta elektrotechniky a informatiky (Skratka fakultySkratka fakulty) |
| Katedra: | Názov katedryNázov katedry (Skratka katedrySkratka katedry) |
| Študijný odbor: | Číslo a názov odboru (pozri zadávací list)Číslo a názov odboru (pozri zadávací list) |
| Študijný program: | ŠpecializáciaŠpecializácia |
| Mesto: | KošiceKošice |
| Vedúci práce: | Titul Meno Priezvisko, TitulTitul Meno Priezvisko, Titul |
| Konzultanti práce: | Titul Meno1 Priezvisko1, Titul  Titul Meno2 Priezvisko2, Titul Titul Meno1 Priezvisko1, Titul  Titul Meno2 Priezvisko2, Titul |
| Dátum odovzdania: | 1515. májmáj 20102010 |
| Dátum obhajoby: | 15. 6. 201015. 6. 2010 |
| Kľúčové slová: | Kľúčové slovo1, kľúčové slovo2, ...Kľúčové slovo1, kľúčové slovo2, ... |
| Kategória konspekt: | Predmetovú kategóriu a podkategóriu vyberte podľa http://www.lib.tuke.sk/?page=konspektPredmetovú kategóriu a podkategóriu vyberte podľa http://www.lib.tuke.sk/?page=konspekt |
| Citovanie práce: | Priezvisko1 Priezvisko2 Priezvisko3, Meno: Názov práce . Diplomová práca alebo Bakalárska práca. Košice: Technická univerzita v Košiciach, Presný názov fakulty napr. Fakulta elektrotechniky a informatiky, 2010. 3 s. |
| Názov práce v AJ: | Názov anglickyNázov anglicky |
| Podnázov práce v AJ: |  |
| Kľúčové slová v AJ: | Kľúčové slovo1, kľúčové slovo2, ...Kľúčové slovo1, kľúčové slovo2, ... |

|  |
| --- |
| **Abstrakt v SJ**  Abstrakt v slovenčine (referát) je povinnou súčasťou každej práce. Je výstižnou krátkou charakteristikou obsahu dokumentu. Abstrakt býva informatívny a zachováva tematické a štýlové vlastnosti práce. Nevyjadruje hodnotiace stanovisko autora. Obsahuje údaje o cieľoch práce, metódach, výsledkoch a záveroch. Text abstraktu sa píše ako jeden odstavec. Abstrakt neobsahuje odkazy na samotný text práce. Mal by mať rozsah asi 250 slov, nemal by presiahnuť jednu stranu. Pri štylizácii sa používajú celé vety, slovesá v činnom rode a tretej osobe. Používa sa odborná terminológia, menej zvyčajné termíny, skratky a symboly sa pri prvom výskyte v texte definujú. |

|  |
| --- |
| **Abstrakt v AJ**  Text abstraktu v svetovom jazyku je potrebný pre integráciu do medzinárodných informačných systémov (napr. The Network Digital Library of Theses and Dissertations). Ak nie je možné jazykovú verziu umiestniť na jednej strane so slovenským abstraktom, je potrebné umiestniť ju na samostatnú stranu (cudzojazyčný abstrakt nemožno deliť a uvádzať na dvoch stranách). |

**Zadanie práce**

Tu vložte naskenované zadanie úlohy.

Odporúčame skenovať na 200-300 DPI, čierno-bielo (alebo 2 farby)

! v jednej vytlačenej ZP musí byť vložený **originál** zadávacieho listu !

**Čestné vyhlásenie**

Vyhlasujem, že som celú diplomovú prácu vypracoval/a samostatne s použitím uvedenej odbornej literatúry.

Autori metodických príručiek (pozri Katuščák [17], Gonda [16]) o záverečných prácach sa nazdávajú, že takéto vyhlásenie je zbytočné, nakoľko povinnosť vypracovať záverečnú prácu samostatne, vyplýva študentovi zo zákona a na autora práce sa vzťahuje autorský zákon.

Košice, 15. máj 2010

..........................................

*vlastnoručný podpis*

**Poďakovanie**

Na tomto mieste môže byť vyjadrenie poďakovania napr. vedúcemu práce resp. konzultantom za pripomienky a odbornú pomoc pri vypracovaní práce. Nie je zvykom ďakovať za rutinnú kontrolu, menšiu spoluprácu alebo všeobecné rady. Vyjadrenie poďakovania v prípade využitia inej práce sa uskutočňuje formou citácie na konci hlavného textu práce a odkazy na citáciu sa musia uviesť aj na zodpovedajúcich miestach v texte.

**Predhovor**

Predhovor je povinnou náležitosťou diplomovej práce (pozri Gonda ([16]). V predhovore diplomant uvedie základné charakteristiky svojej diplomovej práce a okolnosti jej vzniku. Vysvetlí dôvody, ktoré ho viedli k voľbe témy, cieľ a účel práce a stručne informuje o hlavných metódach, ktoré pri spracovaní diplomovej práce použil. Napr.:

Problematika .... je predmetom početných výskumov. U nás sa týmito otázkami zaoberá XY v učebnici Názov učebnice. Vo svete ...

Hlavný dôvod, ktorý ma viedol k výberu témy bol... Zámerom mojej práce je ... Mojou ambíciou je vyriešiť ...

Predkladaná diplomová práca vznikla za spolupráce ...

**Obsah**

[Zoznam obrázkov 10](#_Toc322139659)

[Zoznam tabuliek 11](#_Toc322139660)

[Zoznam symbolov a skratiek 12](#_Toc322139661)

[Slovník termínov 13](#_Toc322139662)

[Úvod 14](#_Toc322139663)

[1 Formulácia úlohy 15](#_Toc322139664)

[2 Analýza 16](#_Toc322139665)

[3 QR kódy 17](#_Toc322139666)

[3.1 Možnosti QR kódu 17](#_Toc322139667)

[3.2 Štruktúra 18](#_Toc322139668)

[3.3 Korekcia chýb 18](#_Toc322139669)

[4 Operačný systém Google Android 20](#_Toc322139670)

[4.1 Architektúra 20](#_Toc322139671)

[4.1.1 Linuxové jadro 21](#_Toc322139672)

[4.1.2 Knižnice 21](#_Toc322139673)

[4.1.3 Android Runtime 22](#_Toc322139674)

[4.1.4 Aplikačný framework 23](#_Toc322139675)

[4.1.5 Aplikácie 23](#_Toc322139676)

[5 Analýza možností vytvárania aplikácií pre platformu Android 24](#_Toc322139677)

[5.1 Android softvérový vývojový balíček 24](#_Toc322139678)

[5.1.1 Vývojové nástroje 25](#_Toc322139679)

[5.1.2 Debugovanie 25](#_Toc322139680)

[5.1.3 Virtuálne zariadenie 25](#_Toc322139681)

[6 Jadro práce. Názov ďalšej kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly 28](#_Toc322139682)

[6.1 Názov podkapitoly Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod 28](#_Toc322139683)

[6.1.1 Názov Tretia úroveň Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia 28](#_Toc322139684)

[6.1.2 Názov Tretia úroveň Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia 29](#_Toc322139685)

[7 Ilustrácie, tabuľky, rovnice, krížové odkazy 30](#_Toc322139686)

[7.1 Ilustrácie 30](#_Toc322139687)

[7.2 Tabuľky 31](#_Toc322139688)

[7.2.1 Tabuľky prevzaté z iných zdrojov 32](#_Toc322139689)

[7.3 Rovnice, vzorce 32](#_Toc322139690)

[7.4 Krížové odkazy 33](#_Toc322139691)

[8 Záver 35](#_Toc322139692)

[Zoznam použitej literatúry 36](#_Toc322139693)

[Prílohy 39](#_Toc322139694)

[Curriculum vitae 40](#_Toc322139695)

Zoznam obrázkov

Obr. 1 Ukážka QR kódu 17

Obr. 2 Porovnanie kódov 17

Obr. 3 Štruktúra QR kódu 18

Obr. 4 Poškodený QR kód 19

Obr. 5 Umelecké stvárnenie QR kódu 19

Obr. 6 Logo operačného systému 20

Obr. 7 Architektúra operačného systému Android 21

Obr. 8 Android virtuálne zariadenie 26

Obr. 9 Životný cyklus aktivity 28

Obr. 10 33

Obr. 11 37

Zoznam tabuliek

Tab. 1 Kapacita QR kódov 17

Tab. 2 Tabuľka stupňov aplikačných rozhraní 24

Tab. 3 34

Tab. 4 35

Zoznam symbolov a skratiek

μ **micro**, 10-6

SI **S**ystème **I**nternational

V **volt**, základná jednotka napätia v sústave SI

Slovník termínov

**Dizertácia** je rozsiahla vedecká rozprava, v ktorej sa na základe vedeckého výskumu a s použitím (využitím) bohatého dokladového materiálu ako i vedeckých metód rieši zložitý odborný problém.

**Font** je súbor, obsahujúci predpisy na zobrazenie textu v danom písme, napr. na tlačiarni. To čo vidíme je písmo; font je súbor a nevidíme ho.

**Meter** (m) je vzdialenosť, ktorú svetlo vo vákuu prejde za časový interval 1/299 792 458 sekundy.

**Proces** je postupnosť či rad časovo usporiadaných udalostí tak, že každá predchádzajúca udalosť sa zúčastňuje na determinácii nasledujúcej udalosti.

Úvod

Hlavný text práce obsahuje úvod, jadro (číslované kapitoly a podkapitoly druhej a tretej úrovne s ilustráciami a tabuľkami), záver a zoznam použitej literatúry. Úvod nemá byť číslovaný.

Úvod stručne a jasne, pritom však podrobnejšie ako v predhovore (pozri Gonda [16])

* vyjadruje stav poznania alebo praxe v danej oblasti, ktorá je predmetom práce,
* zdôvodní aktuálnosť témy,
* nastolí problémy, ktoré chce vyriešiť,
* vysvetlí účel a ciele práce,
* opíše použité metódy a postup riešenia,
* uvedie vzťah práce k ďalším prácam v danej oblasti, spresní informačné zdroje a pramene, ktoré najviac využíval (a ktoré uvedie v zozname použitej literatúry),
* zdôvodní význam riešenia problematiky,
* načrtne stručný obsah kapitol.

V úvode nie je potrebné opakovať to, čo je uvedené v abstrakte. Nie je vhodné podrobne opisovať metódy, experimentálne výsledky, ani opakovať to, čo je uvedené v závere. Aj keď je úvod umiestnený na začiatku, jeho konečnú verziu píše diplomant až po dokončení celej práce.

1. Formulácia úlohy

Text diplomovej práce môže obsahovať v rámci kapitoly 1 formuláciu úlohy resp. úloh, riešených v práci. V tejto časti diplomant rozvedie spôsob, akým budú riešené úlohy a tézy, formulované v zadaní práce. Uvedie tiež prehľad podmienok riešenia. Ak formulácia úlohy nie je potrebná, uvedie sa iný názov tejto kapitoly (Názov kapitoly 1) podľa riešenej problematiky.

Prvá kapitola spravidla predstavuje teoreticko-metodologickú časť práce. Táto časť zvyčajne obsahuje:

* teoretické poznatky vzťahujúce sa na danú úlohu (diplomat prezentuje poznatky nadobudnuté štúdiom),

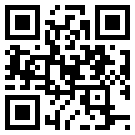
charakteristiku metód a postupov, ktoré diplomant použil pri riešení úlohy.

1. Analýza

Bolo potrebne teoreticky nastudovat toto a prakticky toto

1. QR kódy

QR (ang. Quick Response – rýchla odpoveď) kód je druh dvojrozmerného kódu vyvinutého japonskou firmou Denso Wave v roku 1994 za účelom rozšírenia použiteľnosti čiarových kódov a ľahkej interpretácie skenovacími zariadeniami.



Obr. Ukážka QR kódu

* 1. Možnosti QR kódu

Možnosti QR kóduKlasický jednorozmerný čiarový kód obsahuje narozdiel od QR kódu, dáta len v jednom smere, kdežto QR kód obsahuje dáta v smere vertikálnom aj horizontálnom a umožňuje tým uchovávať omnoho viacej informácií.



Obr. Porovnanie kódov

Kým klasické čiarové kódu umožňujú uchovávať maximálne 20 číslic, QR kód je schopný uchovávať až 7089 numerických znakov. Taktiež podporuje rôzne dátové typy ako čísela, znaky abecedy, Kanji symoboly atď.[33]

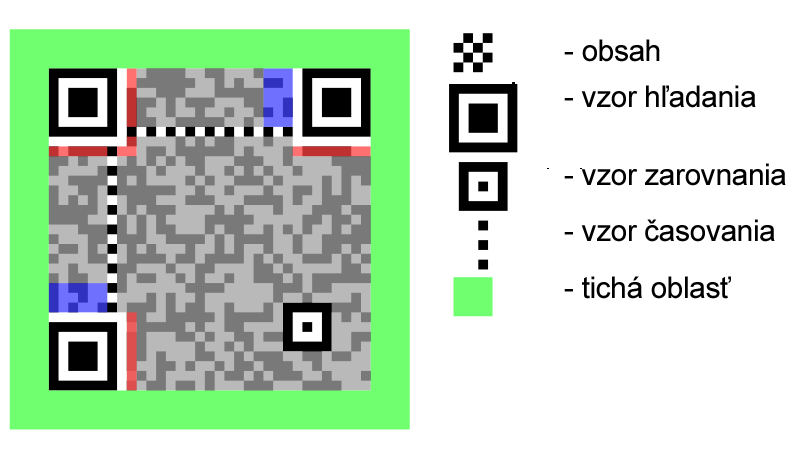
|  |  |
| --- | --- |
| Iba numerické | Max. 7,089 znakov |
| Alphanumerické | Max. 4,296 znakov |
| Kanji | Max. 1,817 znakov |

Tab. Kapacita QR kódov

* 1. Štruktúra

QR kódu je štvorcového tvaru a skladá sa z piatich hlavných častí.

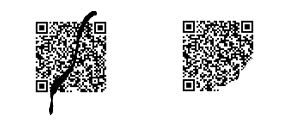
* **Vzor hľadania** – tvoria ho štvorce nachádzajúce sa v troch zo štyroch rohov. Používa sa na uľahčenie zistenia pozície, veľkosti a uhlu otočenia QR kódu.
* **Vzor zarovnania** – využíva sa pre obnovu poškodenia. Môže byť použitý na identifikáciu či kódu obsahuje nelineárne chyby. Pre tento účel je umiestnená nezávislá čierna bunka v strede vzoru.
* **Vzor časovania** – skladá sa z bielych a čiernych buniek vo vertikálnych a horizontálnych líniach, ktoré spájajú tri vzory hľadania.
* **Tichá oblasť** – prázdna oblasť okolo kódu, nevyhnutná pre čítanie.
* **Oblasť dát** – Dáta uložené v kóde sú uschované v tejto oblasti v binárnej podobe. [35]



Obr. Štruktúra QR kódu

* 1. Korekcia chýb

Kód ma schopnosť samokorekcie chýb. Poškodený kód je možné správne prečítať aj pri 30% poškodení. Korekcia je realizovaná Reed-Solomonovým algoritmom korekcie chýb. Kvôli jeho návrhu a používaniu 8-bitových slov, kód nesmie byť dlhší ako 255 slov. Samozrejme QR kódy obsahujú omnoho väčšie množstvo dát a preto je nutné obsah deliť do blokov.



Obr. Poškodený QR kód

Vďaka korekcii chýb je možné taktiež vytvárať umelecké verzie QR kódov, ktoré budu stále plne funkčné, ale omnoho príťažlivejšie pre používateľa.



Obr. Umelecké stvárnenie QR kódu

1. Operačný systém Google Android

Operačný systém Google Android je open-source operačný systém založený na Linuxovom jadre. Systém poskytuje riešenia pre mobilné telefóny, tablety, netbooky, televízory a iné zariadenia. Platforma je tvorená konzorciom firiem Open Handset Alliance, ktoré je tvorené z 84 veľkých softvérových, hardvérových a telekomunikačných firiem, vedených firmou Google.[34]

Operačný systém bol spočiatku vyvíjaný firmou Android, ktorá bola v roku 2005 odkúpená firmou Google. Následne bol zriadený Android Open Source Project, kde tento tým bol poverený vývojom a údržbou operačného systému.[31]



Obr. Logo operačného systému

Samotná platforma Android dáva k dispozícii nielen operačný systém s používateľským prostredím pre koncových používateľov, ale aj kompletné riešenie nasadenia operačného systému (špecifikácia ovládačov a pod.) pre mobilných operátorov a výrobcov zariadení a v neposlednom rade pre vývojárov aplikácií poskytuje efektívne nástroje pre ich vývoj - Software Development Kit.

* 1. Architektúra

Systém je rozdelený do piatich vrstiev. Každá vrstva má svoj účel a nemusí byť priamo oddelená od ostatných vrstiev.



Obr. Architektúra operačného systému Android [29]

* + 1. Linuxové jadro

Najnižšou vrstvou architektúry je jadro operačného systému, ktoré tvorí abstraktnú vrstvu medzi používaným hardvérom a zvyškom softvéru vo vyšších vrstvách. Jadro systému Androidu je postavené na Linuxe. Využíva mnoho jeho vlastností, ako sú podpora [správy pamäte](http://sk.wikipedia.org/w/index.php?title=Spr%C3%A1va_pam%C3%A4te&action=edit&redlink=1), správa sietí, zabudované ovládače, [správa procesov](http://sk.wikipedia.org/w/index.php?title=Spr%C3%A1va_procesov&action=edit&redlink=1) alebo súbežný beh aplikácií, ktoré bežia ako samostatné procesy s prioritou stanovenou systémom.

Táto vlastnosť prispieva k stabilite a tiež ochrane systému. Naopak systém nepodporuje grafické používateľské rozhranie X Window System a ani úplnú sadu [GNU knižníc](http://sk.wikipedia.org/w/index.php?title=GNU_C_Library&action=edit&redlink=1). Dôvodom použitia jadra Linux bola tiež vlastnosť pomerne jednoduchej kompilácie na rôznych zariadeniach a tým zaručená [prenositeľnosť](http://sk.wikipedia.org/w/index.php?title=Portovanie_softv%C3%A9ru&action=edit&redlink=1).

* + 1. Knižnice

Knižnice, písané v jazyku [C a C++](http://sk.wikipedia.org/wiki/C%2B%2B), využívajú ich rôzne časti systému. Tieto funkcie sú vývojárom poskytnuté prostredníctvom Android Application Framework. Tu sú uvedené iba niektoré príklady knižníc:

* Media Libraries - knižnica podporuje prehrávanie video a audio formátov, obrazových súborov, napr. [MPEG4](http://sk.wikipedia.org/wiki/MPEG4), [H.264](http://sk.wikipedia.org/wiki/H.264), [MP3](http://sk.wikipedia.org/wiki/MP3), [AAC](http://sk.wikipedia.org/w/index.php?title=AAC&action=edit&redlink=1), [AMR](http://sk.wikipedia.org/w/index.php?title=AMR&action=edit&redlink=1), [JPG](http://sk.wikipedia.org/wiki/JPG) a [PNG](http://sk.wikipedia.org/wiki/PNG).
* LibWebCore - knižnica webového prehliadača, ktorý podporuje aj vložené náhľady webových stránok.
* [Libc](http://sk.wikipedia.org/w/index.php?title=Libc&action=edit&redlink=1) - odvodená [BSD](http://sk.wikipedia.org/wiki/BSD) štandardná knižnica systému C vyladená pre [embedded zariadenia](http://sk.wikipedia.org/w/index.php?title=Embedded_syst%C3%A9m&action=edit&redlink=1).
* [SQLite](http://sk.wikipedia.org/w/index.php?title=SQLite&action=edit&redlink=1) - odľahčená knižnica pre prístup k [relačným databázam](http://sk.wikipedia.org/wiki/Rela%C4%8Dn%C3%A1_datab%C3%A1za)
* [OpenSSL](http://sk.wikipedia.org/w/index.php?title=OpenSSL&action=edit&redlink=1) - [secure socket layer](http://sk.wikipedia.org/w/index.php?title=Secure_socket_layer&action=edit&redlink=1)
* [FreeType](http://sk.wikipedia.org/w/index.php?title=FreeType&action=edit&redlink=1) - knižnica pre rendering bitmapových a vektorových fontov.
* [OpenGL](http://sk.wikipedia.org/wiki/OpenGL) - knižnica na vykresľovanie 3D grafiky.
  + 1. Android Runtime

Vrstva Android Runtime obsahuje aplikačný [virtuálny stroj](http://sk.wikipedia.org/w/index.php?title=Virtu%C3%A1lny_stroj&action=edit&redlink=1) zvaný Dalvik, ktorý bol vyvíjaný od roku 2005 špeciálne pre Android.

[Dalvik Virtual Machine](http://sk.wikipedia.org/w/index.php?title=Dalvik_Virtual_Machine&action=edit&redlink=1) (DVM) je [registrovo orientovaná architektúra](http://sk.wikipedia.org/w/index.php?title=Registrovo_orientovan%C3%A1_architekt%C3%BAra&action=edit&redlink=1), využíva základné vlastnosti linuxového jadra, ako je správa pamäte alebo práca s [vláknami](http://sk.wikipedia.org/w/index.php?title=Vl%C3%A1kno_(program)&action=edit&redlink=1). Vznik nového virtuálneho stroja bol iniciovaný z dvoch dôvodov. Prvým dôvodom boli [licenčné práva](http://sk.wikipedia.org/wiki/Licencia), pretože jazyk [Java](http://sk.wikipedia.org/wiki/Java) a jeho knižnice sú voľne šíriteľné, zatiaľ čo virtuálny stroj Java Virtual Machine nie je. Ďalším dôvodom bola optimalizácia virtuálneho stroja pre mobilné zariadenia a to predovšetkým v oblasti pomeru úspory energie a výkonu.

V tejto vrstve sú tiež obsiahnuté základné knižnice programovacieho jazyka Java. Knižnice sa svojím obsahom blížia platforme [Java Standard Edition](http://sk.wikipedia.org/w/index.php?title=Java_Standard_Edition&action=edit&redlink=1). Hlavný rozdiel je v neprítomnosti knižníc pre užívateľské rozhranie ([AWT](http://sk.wikipedia.org/w/index.php?title=AWT&action=edit&redlink=1) a [Swing](http://sk.wikipedia.org/wiki/Swing)), ktoré boli nahradené knižnicami užívateľského rozhrania pre Android alebo pridanie [knižnice Apache](http://sk.wikipedia.org/w/index.php?title=Kni%C5%BEnica_Apache&action=edit&redlink=1) pre prácu so sieťou. Preklad aplikácie vyvinutej pre Android prebieha skompilováním zdrojového Java kódu do Java byte kódu pomocou rovnakého [kompilátora](http://sk.wikipedia.org/wiki/Kompil%C3%A1tor), ako sa používa v prípade prekladu Java aplikácií. Potom sa prekompiluje Java [byte kód](http://sk.wikipedia.org/w/index.php?title=Byte_k%C3%B3d&action=edit&redlink=1) pomocou Dalvik kompilátora a výsledný Dalvik byte kód je spustený na DVM. Každá spustená Android aplikácie má svoj vlastný [proces](http://sk.wikipedia.org/w/index.php?title=Proces_(program)&action=edit&redlink=1) s vlastnou inštanciou DVM.

Android aplikácie sú spúšťané v sandboxe, izolovanom mieste v operačnom systéme, kde aplikácie nemajú prístup k zvyšku zdrojov systému, ak im vhodnými povoleniami udelenými používateľom nebolo toto pridelené pri inštalácií.

* + 1. Aplikačný framework

Vrstva application framework je pre vývojárov najdôležitejšia. Poskytuje prístup k veľkému počtu služieb, ktoré môžu byť použité priamo v aplikáciách. Tieto služby môžu sprístupňovať dáta v iných aplikáciách, prvky používateľského rozhrania, upozorňovací stavový riadok, aplikácie bežiace na pozadí, hardvér používaného zariadenia a mnoho ďalších služieb a funkcií. Základná sada služieb zahŕňa predovšetkým:

* **Pohľady** – (z angl. *Views*) tieto prvky sú použité pre zostavenie používateľského rozhrania ako zoznamy, textové pole, tlačidlá a iné.
* **Správca zdrojov** - poskytuje prístup k zdrojom aplikácie, ako sú [reťazce](http://sk.wikipedia.org/w/index.php?title=Textov%C3%BD_re%C5%A5azec&action=edit&redlink=1), [grafika](http://sk.wikipedia.org/wiki/Grafika), pridané súbory.
* **Správca notifikácií** - umožňuje všetkým aplikáciám zobraziť vlastné upozornenie v stavovom riadku.
* **Správca aktivít** - riadi životný cyklus aplikácií a poskytuje orientáciu v [zásobníku](http://sk.wikipedia.org/w/index.php?title=Z%C3%A1sobn%C3%ADk_(d%C3%A1tov%C3%A1_%C5%A1trukt%C3%BAra)&action=edit&redlink=1) s aplikáciami.
  + 1. Aplikácie

Najvyššiu vrstvu systému tvoria  základné aplikácie, ktoré využívajú bežný [použivatelia](http://sk.wikipedia.org/w/index.php?title=Pou%C5%BE%C3%ADvate%C4%BE_(informatika)&action=edit&redlink=1). Môže ísť o aplikácie predinštalované alebo o aplikácie tretích strán, dodatočne stiahnutých z Google Play.[30]

1. Analýza možností vytvárania aplikácií pre platformu Android

Vývoj aplikácií pre platformu Android je realizovaný pomocou Android softvérového vývojového balíčka (z angl. software development kit) v objektovo-orientovanom jazyku Java, alebo v jazykoch C a C++ pomocou Android natívneho vývojového balíčka (z angl. native development kit).

* 1. Android softvérový vývojový balíček

Softvérový vývojový balíček pre platformu Android je súhrn nástrojov, ktoré umožňujú vývoj aplikácií pre tento operačný systém. Nástroje obsahujú debugger, knižnice, emulátor, dokumentáciu, vzorové ukážky kódov atď.

Všeobecne sú nástroje delené do dvoch skupín:

* **Nástroje balíčka** – (z angl. *SDK tools*) nástroje, ktoré sú platformovo nezávislé a sú potrebné pre každý verziu platformy.
* **Nástroje platformy** – (z angl. *platform tools*) nástroje, ktoré sú špecializované pre danú verziu platformy.[18]

Tab. Tabuľka stupňov aplikačných rozhraní

|  |  |  |
| --- | --- | --- |
| Verzia platformy | API level | VERSION\_CODE |
| Android 4.0 | 14 | ICE\_CREAM\_SANDWICH |
| Android 3.2 | 13 | HONEYCOMB\_MR2 |
| Android 3.1.x | 12 | HONEYCOMB\_MR1 |
| Android 3.0.x | 11 | HONEYCOMB |
| Android 2.3.4  Android 2.3.3 | 10 | GINGERBREAD\_MR1 |
| Android 2.3.2  Android 2.3.1  Android 2.3 | 9 | GINGERBREAD |
| Android 2.2.x | 8 | FROYO |
| Android 2.1.x | 7 | ECLAIR\_MR1 |
| Android 2.0.1 | 6 | ECLAIR\_0\_1 |
| Android 2.0 | 5 | ECLAIR |
| Android 1.6 | 4 | DONUT |
| Android 1.5 | 3 | CUPCAKE |
| Android 1.1 | 2 | BASE\_1\_1 |
| Android 1.0 | 1 | BASE |

Platí pravidlo spätnej kompatibility (API level). Android 2.x boli verzie určené pre mobilné telefóny, 3.x pre tablety. S príchodom novej verzie 4.0 Ice Cream Sandwich sa toto rozdelenie zjednocuje, maže rozdiely medzi API pre tablety a telefóny a uľahčuje prácu vývojárom.

* + 1. Vývojové nástroje

Vývojové nástroje pre Android (z angl. *Android Development Tools*) je zásuvný modul pre vývojové prostredie Eclipse, predstavujúci integráciu SDK do tohto rozšíreného vývojového prostredia.Modul umožňuje jednoduchý a pohodlný vývoj Android aplikácii. Rozširuje schopnosti prostredia Eclipse a umožňuje rýchlo vytvárať Android projekty, aplikačné rozhranie, pridávať a používať komponenty z Android Framework API, debugovať aplikáciu pomocou Android SDK nástrojov.

Vývoj Android aplikácií v prostredí Eclipse je firmou Google silne doporučený, je to najrýchlejšia možnosť ako začať s vývojom, predovšetkým kvôli mnohým existujúcim návodom, vlastnému XML editoru pre tvorbu používateľských rozhraní.[19]

* + 1. Debugovanie

Debugovanie je realizované pomocou Android Debug Bridge, versatilného klient – server programu príkazového riadku umožňujúceho komunikáciu s emulátorom alebo Android zariadením. Program umožňuje napríklad inštalovať aplikáciu na zariadenie, smerovať porty alebo kopírovať súbory do alebo zo zariadenia. [20]

* + 1. Virtuálne zariadenie

Správca Android Virtual Device (ďalej správca AVD) je správca virtuálneho zariadenia, ktorý umožňuje modelovať hardvér a softvér emulovaného zariadenia.



Obr. Android virtuálne zariadenie

Android irtuálne zariadenie pozostáva z:

* **hardvérového profilu** – umožňuje definovať či zariadenie má kameru, qwerty klávesnicu, koľko pamäte používa a podobne
* **softvérového profilu** – možnosť definovať, ktorá verzia platformy Android bude spustená na emulátore
* **iné nastavenia** – umožňuje meniť rozmery, orientáciu, vzhľad, vytvoriť virtuálnu SD kartu a podobne [21]

ANDROID NDK

MOZNO ANALYZA PROSTREDI?

1. Analýza štruktúry Android aplikácie
   1. Komponenty

Android aplikácia sa skladá zo štyroch základných častí:

* + Aktivita (z angl. Activity)
  + Služba (z angl. Service)
  + Poskytovateľa obsahu (z angl. Content Provider)
  + Príjmače vysielania (z angl. Broadcast Receivers)

Nie každá aplikácia obsahuje všetky štyri komponenty, ale určite bude pozostávať z nejakej ich kombinácie. Použitie týchto prvkov musí byť zaznamenané v Android manifeste.[22]

* + 1. Aktivity

Aktivita je komponenta aplikácie, ktorá poskytuje obrazovku s ktorou používatelia komunikujú. Každá aplikácia má svoj životný cyklus a pridelené okno do ktorého vykreslí používateľské rozhranie.

Aplikácia zvyčajne pozostáva z viacerých voľne previazaných aktivít. Obvykle je jedna z aktivít označená ako hlavná, a je zobrazená používateľovi pri štarte aplikácie. Každá aktivita môže spustiť inú aktivitu, pre vykonanie nejakého špecifického úkonu. Vždy, keď je spustená nová aktivita, predchádzajúca aktivita je pozastavená, ale je stále uschovaná v zásobníku aktivít a nová aktivita preberá fokus. Zásobník aktivít podlieha klasickému LIFO algoritmu, z vplýva že, ak používateľ ukončíl prácu v aktuálnej aktvite a stlačí tlačidlo Naspäť, bude aktivita zo zásobníka odstránená a predošlá aktivita bude obnovená.

Keď je aktivita prerušená kvôli štartu inej aktivity, je jej o tejto udalosti oznámené pomozou zmeny jej stavu cez návratové volania metód jej životného cyklu.[23]



Obr. Životný cyklus aktivity [32]

* + 1. Služby

Služby sú komponety aplikácie ktoré vykonávajú dlhotrvajúce operácie na pozadí a neposkytujú používateľské rozhranie. Odlišný komponent môže spustiť službu a tá bude pokračovať vo vykonávaní činnosti aj po uzavretí aplikácie. Službou sú napríklad prehrávanie hudby, sieťové transakcie, vstupné a výstupné operácie pri práci so súbormi. Služba má dve podoby:

* **Neviazaná** – služba je neviazaná, ak aplikačný komponent, napríklad aktivita, spustí službu volaním startService(). Spustená služba môže nekonečne vykonávať svoju činnosť aj po zničení aktivity, ktorá ju spustila a nevracia výsledok alebo návratové volanie volajúcemu.
* **Viazaná** – služba je viazaná, ak ju volajúci komponent so sebou previaže volaním bindService(). Viazaná služba poskytuje klient-server rozhranie, ktoré umožňuje interakciu so službou, posielať požiadavky alebo obdržiavať výsledky. Takáto služba je spustená iba pokiaľ je spustený komponent na ktorý je previazaná.[24]
  + 1. Poskytovatelia obsahu

Poskytovateľ obsahu spravuje zdieľané dáta. Dáta môžu byť uložené v súborovom systéme, SQLite databáze alebo na internete. Pomocou poskytovateľa obsahu, aplikácie možu dopytovať alebo meniť dáta. Napríklad, Android poskytuje poskytovateľ obsahu pre kontakty používateľa. Každá aplikácia s potrebnými právami, môže čítať a zapisovať informácie o danej osobe z aplikácie kontaktov.

* + 1. Príjmače vysielania

Príjmače vysielania sú komponenty, ktoré reagujú na vysielacie oznámenia na celej šírke systému. Mnoho vysielaní pochádza zo samotného systému, ako napríklad vysielanie oznamujúce vypnutie obrazovky alebo nízkeho stavu batérie. Neposkytujú používateľské rozhranie, ale umožňujú vytvoriť notifikáciu v správcovi notifikácií. Vysielanie sa realizuje pomocou správ typu Intent.[25]

* + 1. Správy typu Intent

Špeciálne správy typu Intent predstavujú médium spúšťania komponentov aplikácie. Umožňujú previazanie vlastných komponentov aplikácie alebo komponentov aplikácie externej počas behu programu. Samotná správa je objekt typu Intent a je pasívnou dátovou štruktúrou držiacou abstraktný popis operácie ktorá sa má vykonať, alebo v prípade vysielaní (z angl. broadcast), popis niečoho čo sa udialo a je oznamované. Existujú rôzne mechanizmy pre doručovanie správ k jednotlivým druhom komponentov:

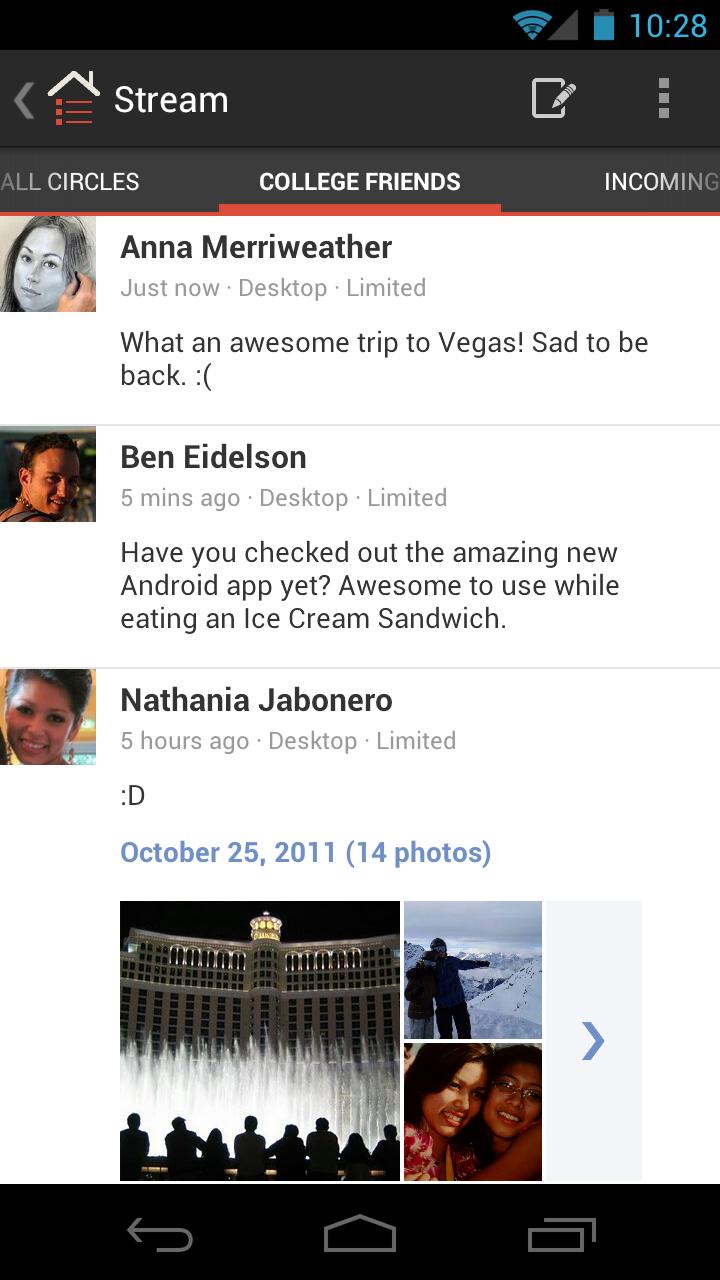
* **aktivita** – pre spustenie aktivity pre objekt typu Intent použitý ako argument volania Context.startActivity()
* **služba** – objekt správy je používaný pre spustenie novej služby alebo pre doručenie inštrukcií už spustenej službe. Pre naviazanie spojenia medzi volajúcim komponentom a volanou službou je použité volanie Context.startService()

**vysielanie** – správa je doručená pomocou Context.sendBroadcast() všetkým načúvajúcim príjmačom vysielania. [26]

* 1. Používateľské rozhranie

Používateľské rozhranie Android aplikácie je tvorené z pohľadov (z angl. Views) a skupín pohľadov (z angl. ViewGroups). Existuje mnoho druhov pohľadov a skupín pohľadov, ktoré rozširujú práve triedu View.

Rozhrania sú tvorené pomocou súboru rozmiestnenia vo formáte XML alebo z prostredia jazyka Java.



Obr. Ukážka používateľského rozhrania[28]

* + 1. Pohľady

Pohľady su základná jednotka používateľského rozhrania platformy Android. Trieda View predstavuje bázovú triedu pre prvky zvané widgety, ktoré potredstavujú plne implementované objekty používateľského rozhrania, ako napríklad tlačidlá a textové polia.

Objekty pohľadov sú dátovými štruktúrami, ktorých vlastnosti sú uchovavané v ich parametroch rozmiestnenia a obsahu, pre špecifickú obdĺžnikovú oblasť, ktorú zaberajú na obrazovke. Pohľad sám spravuje svoje rozmery, rozmiestnenie, vykresľovanie, zmeny fokusu, skrolovanie a interakciu pomocou giest na obdĺžnikovej oblasti, ktorá mu prislúcha. Pohľad je taktiež príjmateľom a prostriedkom interakcie s používateľom.

* + 1. Skupiny pohľadov

Trieda ViewGroup zase predstavuje triedu od ktorej sú odvodené podtriedy zvané rozmiestnenia (z angl. layouts), ktoré poskytujú rôzne druhy rozloženia pohľadov, ako napríklad lineárlne rozmiestnenie alebo rozmiestnenie relatívne.



Obr. Príklad definície rozmiestenania pomocou XML

Najčastejšou formou tvorby rozmiestnenia rozhrania je pomocou jeho XML formou zápisu. XML poskytuje štruktúru pre rozmiestnenie, ktorá je ľahko čitateľná pre človeka. Každý element v XML, je práve pohľad alebo skupina pohľadov, respektíve ich rozšírenia. V stromovej štruktúre predstavujú pohľady listy a skupiny pohľadov reprezentujú uzly stromu. Názov elementu v XML korešponduje s názvom Java triedy, ktorú reprezentuje. [27]



Obr. Schématický náhľad na hierarchiu

* 1. Manifest

Pred štartom aplikačného komponentu, systém musí vedieť o jeho existencií v manifestovom súbore AndroidManifest.xml. V manifeste musia byť deklarované všetky komponenty aplikácie a ten sa musí nachádzať v koreňovom adresári projektu. Okrem deklarácie komponentov, manifest obsahuje okrem iného: [again 25]

* + Práva udelené používateľom pri inštalácií aplikácie
  + Deklarácie minimálneho stupňa aplikačného rozhrania
  + Deklarácie hardvérových a softvérových prostriedkov požadovaných aplikáciou



Obr. Ukážka manifestu

UI

1. Analýza návrhu riešenia inventárneho systému

Úlohou bakalárskej práce bolo vytvoriť aplikáciu inventárneho systému pre platformu Google Android.

Aplikácia má umožňovať vykonávať inventúru v určených miestnostiach, ktorých zoznam a obsah je v preddefinovanom tvare a formáte .xml uložený na strane servera. Tento súbor je aplikáciou prevzatý cez HTTP protokol a jeho dáta sú načítané do pamäte.

Identifikácia tovaru je riešená pomocou QR kódov fyzicky umiestnených na danom predmete. Načítanie tohto kódu sa zabezpečuje pomocou funkcie kamery, kde je následne obraz rozpoznaný a samotná zakódovaná informácia je interpretovaná aplikáciou pomocou knižnice ZXing, do formy reťazca znakov.

Následne je načítaný predmet porovnaný s predmetmi v miestnosti podľa identifikačného čísla. Ak sa predmet v aktuálnej miestnosti nachádza a bol doteraz označený ako chýbajúci, bude označený ako identifikovaný.

Po ukončení procesu inventúry, je možné výsledky inventúry exportovať pomocou skriptu vo formáte .php na stranu servera, kde budú prenesené výstupy spracované do graficky prehľadného dokumentu vo formáte .html a uložené pre budúcu archiváciu. Jednotlivé výsledky budú triedené podľa dátumu inventúry a miestnosti v ktorej inventúra prebiehala. Takto archivované výstupné správy bude možné pohodlne a jednoducho prezerať v prostredí internetového prehliadača. Získané výsledky je možné uložiť taktiež aj lokálne do pamäte telefónu.

O skutočnosti vykonanej inventúry a jej výsledkov bude možné upozorniť určené osoby pomocou e-mailovej notifikácie odosielanej na preddefinované e-mailové adresy.

1. Návrh riešenia inventárneho systému
   1. Načítanie miestností

Na uschovanie obsahu inventára bol použitý zdrojový súbor vo formáte XML. Stromová povaha XML súborov je ideálna pre danú ulohu, pretože celý inventár potrebujeme členiť do miestností obsahujúce samotné položky.

Kvôli existencii už hotových a predtým používaných QR kódov na katedre, bol formát vlastností položiek prevazaný.

Presné dodržanie nasledovnej štruktúry zdrojového súboru je nevyhnutné pre správnu funkciu parsera. Zdrojový súbor musí byť v nasledujúcom tvare:

<inventory>

<room name="***<meno miestnosti>***">

<item

EVID.C.=“***<nové identifikačné číslo>***“

Stare\_C.="***<staré identifikačné číslo>***“

Opis="***<opis>***"

Kusov="***<počet kusov>***"

Miestnost="***<meno miestnosti>***" />

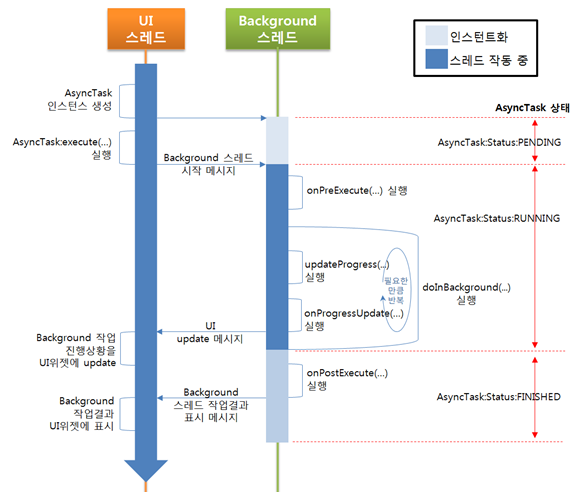
</room>

</inventory>

Takouto štruktúrou vieme definovať miestnosti, v miestnostiach položky, ktoré do nich patria a priradiť položkám ich potrebné dáta. Zdrojový súbor bude uložený na strane servera. Následne po spustení inventúry bude tento súbor vo formáte XML aplikáciou prevzatý a parserom ďalej spracovaný.

V priebehu vývoja nastal klasický problém pri začiatkoch s vývojom na platforme Android. Problem spočíva v tom, že úkon prevzatia súboru z internetu, respektíve hocijakej náročnej operácie sa vykonával na hlavnom vlákne, ktorého hlavná úloha spočíva v staraní sa o používateľské rozhranie, z toho aj plynie jej názov vlákno používateľského rozhrania (z angl. user interface thread). Z toho vyplýva, že budú všetky úkony na hlavnom vlákne pozastavené, kým daná náročná operácia neskončí. Toto má za následok, že aplikácia sa javí ako neresponzívna, pretože hlavné vlákno vykonáva spomínanú sieťovú úlohu a nespracúvava kliknutia používateľa. V tomto prípade to nebolo až tak kritické, pretože zdrojový súbor je veľmi malý, ale je dobrým zvykom do budúcnosti to riešiť korektne.

Takýto náročný úkon má byť riešený na novom vlákne, teda asynchrónne, aby vlákno používateľského rozhrania ostalo neblokované. Pre prácu s asynchrónnymi úkonmi sa používa trieda AsyncTask. Asynchrónna úloha bude bežať na pozadí a po svojom ukončení zavolá príslušné návratové volanie.



Obr. Diagram práce asynchrónnej úlohy

* 1. Parsovanie prevzatého zdrojvého súboru

Po prevzatí zdrojového XML súboru je potrebné ho preparsovať, to znamená vytiahnuť z XML potrebné údaje o položke a s týmito dátami vytvoriť v Jave objekt položky triedou Item, obdobne budú vytvorené objeky miestností pomocou inštancíí triedy Room, ktorým budú priradené zoznamy položiek ktore obsahujú.

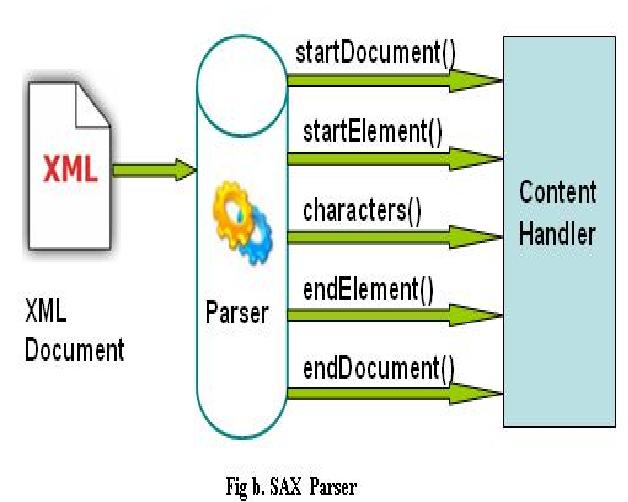
Na parsovanie bola spočiatku použitý regulárny výraz, ktorý sa javil ako postačujúci pre jednoduchú a nemeniacu sa štruktúru zdrojvého súboru ale v závere vývoja sa pri testovaní ukázal ako nedostatočný pri odchytávaní chýb v XML súbore.

Kvôli väčšej robustnosti a jednoduchosti pri odychtávaní chýb bola nakoniec použitá implementácia SAX parsera zabepečujúca trieda Parser.

* + 1. Implementácia parsera

Implementovaný SAX parser je aplikačné rozhranie parsera so sekvenčným prístupom, založeného na udalostiach. Poskytuje mechanizmus pre čítanie dát z XML dokumentu, ktorý je alternatíva ktorú poskytuje objektový model dokumentu (z angl Document Object Model). DOM pracuje ako celok, kde SAX parser pracuje s každým elementom XML dokumentu sekvenčne.

SAX parser poskytuje oproti DOM parserom určité výhody. SAX parser oznamuje len aktuálnu parsovaciu udalosť zavolaním prisúšného návratového volania a neuschováva takmer žiadne informácie predtým získané. Z tohto dôvodu je minimum pamäte potrebnej pre SAX parser priamo úmerná hĺbke stromu XML súboru a predstavuje len zlomok z pamäte potrebnej pre DOM parser, ktorý najprv v pamäti vyskladá celý strom. [36]



Obr. Fungovanie SAX parsera

* 1. Čítanie QR kódov

dasdsadsa

1. Jadro práce. Názov ďalšej kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly kapitoly

Jadrom ďalších kapitol je analýza a syntéza vedúca k riešeniu problému.

* 1. Názov podkapitoly Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod Pod

Podkapitoly diplomovej práce slúžia na členenie textu diplomovej práce s cieľom čo najväčšej prehľadnosti.

* + 1. Názov Tretia úroveň Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia

Editujte svoju prácu v kapitolách a podkapitolách. Čísla kapitol a podkapitol (druhej a tretej úrovne) sa citujú v texte práce takto:

... V kapitole 2sme už uviedli, že ...; ... pozri 4.1 ... atď. ...

Rozsah diplomovej zodpovedá účelu a obsahu. V niektorých študijných odboroch technických vied sú práce vo všeobecnosti kratšie, v spoločenských vedách sú rozsiahlejšie. Rozsah práce je spravidla 60 - 100 strán. Do tohto rozsahu sa počíta len hlavný text, t. j. úvod, kapitoly, záver a zoznam použitej literatúry. Dôležitejší ako rozsah práce je kvalita práce a úroveň jej spracovania. Pri písaní je dôležité dbať na vyváženosť (proporcionálnosť) jednotlivých častí práce (pozri Gonda [16]):

* úvod má spravidla 2 – 3 strany,
* teoreticko-metodologická časť tvorí spravidla jednu tretinu práce,
* ostatné kapitoly tvoria približne dve tretiny práce,
* záver má zvyčajne 2 – 3 strany.

#### Názov štvrtá úroveň štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá štvrtá úroveň

Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye. Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye

* + 1. Názov Tretia úroveň Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia Tretia

qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi i iuw iwu iq iqwuye Yuiweyiuyiu qwuyeiu qwieq wi

1. Ilustrácie, tabuľky, rovnice, krížové odkazy

V práci sa môžu vyskytovať okrem slovného textu aj informácie vyjadrené v obrazovej forme a symbolmi.

* 1. Ilustrácie

**Ilustrácie** sú obrázky obsahujúce **grafy**, **diagramy**, **mapy**, **schémy** a pod. Nie je potrebné rozlišovať rozličné typy ilustrácií, stačí, ak sa všetky označia ako „Obrázok”. Všetky ilustrácie musia byť očíslované súvislým radom číslic v celej práci a musia mať titulky (názov obrázku) pri každom obrázku. Text titulku musí byť pochopiteľný aj bez kontextu. Majú sa zaradiť bezprostredne za textom, kde sa spomínajú po prvýkrát (najlepšie na tej istej strane). Obrázok by mal byť podľa možnosti centrovaný. Pri odkazovaní na daný obrázok v texte treba použiť **krížové odkazy** na obrázok (napr. Obr. 1). Obrázky a tabuľky vkladáme v tejto šablóne takto:

* Vložiť *→ Obrázok*
* Pravý klik na vložený obrázok → *Vložiť popis*
* V okne Popis pri možnosti *Označenie* vyberieme *„Obr.“*
* Pri možnosti *Umiestnenie* vyberieme možnosť „*Pod vybratou položkou“*
* Do kolónky *Popis* dopíšeme názov obrázku

Sablony\Sevcovic\obrazok.emf

Obr. 16

V texte sa na vytvorené obrázky a tabuľky odkazujeme **krížovými odkazmi (**pozri **4.4).**

**Krížový odkaz na obrázky a tabuľky** vytvoríme takto:

* nastavíme sa kurzorom tam, kde sa má objaviť odkaz,
* z panela nástrojov vyberieme: *Vložiť (Insert)* → *Krížový odkaz (Cross-reference),*
* v dialógovom boxe pre Krížový odkaz vyberieme v poli *Typ odkazu (Reference Type) vyberieme Tab.* alebo *Obr.,*
* v poli *Vložiť odkaz (Insert reference) vyberieme vložiť odkaz na: Only lebel and number* (menovka a číslo),
* zo zoznamu popisov vyberieme tabuľku alebo obrázok a zaškrtneme Hypertextový odkaz (Insert as Hyperlink),
* *Insert* a *Close.*

**Aktualizácia krížových odkazov v celom dokumente: CTRL + A, potom F9.**

* 1. Tabuľky

Tabuľky prezentujú myšlienky a tvrdenia popisované v práci. Akýkoľvek tabuľkový materiál, ktorý sa skladá z viac než štyroch alebo piatich riadkov, by mal byť spracovaný do formy tabuľky Tab. 2. Popis a záhlavie tabuľky má byť zrozumiteľné samostatne bez odkazu na text. Záhlavia majú vyjadrovať druh veličiny a typy jednotiek vo forme „veličina/jednotka”, je potrebné používať rovnaké symboly a skratky ako v texte. Každá tabuľka musí mať poradové číslo a titulok, umiestnený zvyčajne nad tabuľkou. Tabuľka by mala mať rovnakú orientáciu, ako text práce.

Tab.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | PP – 01 | PP – 05 | PP – 10 | PP – 16 | PP – 22 |
| C·108 (s−2) | 10,1 | 10,0 | 11,0 | 9,2 | 8 |
| t0 ·10−14 (s) | 2,63 | 1,44 | 0,95 | 2,21 | 10,83 |
| *Ea* (kJ) | 34,26 | 8,33 | 39,76 | 37,31 | 31,86 |
| *Tmin* (K) | 354 | 367 | 367 | 369 | 367 |
| *T1min* (ms) | 141 | 160 | 157 | 175 | 181 |
| *ΔM2* (Gs2) | 5,49 | 5,66 | 5,16 | 5,09 | 5,02 |

* + 1. Tabuľky prevzaté z iných zdrojov

Ak preberáme tabuľku alebo jej časť od iného autora alebo z iného zdroja (napr. z firemných materiálov, interných materiálov inštitúcie a pod.), treba to uviesť pod tabuľkou.

Napr.:

Prameň: Interné materiály MH SR, rok 2003.

Tab.

|  |  |
| --- | --- |
| Názov | Jednotka |
|  |  |

Prameň: Názov zdroja

* 1. Rovnice, vzorce

Rovnice sa uvádzajú v strede riadka, vysvetlivky symbolov na začiatku riadku. *Vysvetlivky symbolov sa uvádzajú od začiatku riadka. Ak je v práci viac vzorcov,* uvádzame číslo vzorca do okrúhlych zátvoriek bez medzier umiestnených na pravom konci riadka. Pre písanie fyzikálnych veličín a matematických premenných sa používa kurzíva. Používame sústavu jednotiek SI (ISO 31 a ISO 1001). Pri písaní rovníc používame **editor rovníc (musíme ho mať nainštalovaný)**.

Rovnice vkladáme v tejto šablóne takto:

* Insert *→ Autotext →Rovnica*
* vyznačíme číslo rovnice (v zátvorke vpravo) a urobíme **záložku** (Bookmark) pre ďalšie odkazy v texte cez:
  + Insert *→ Bookmark →*napíšeme *Názov rovnice* do poľa *Bookmark name (napr. rovnica\_)*
* V okne Popis pri možnosti *Označenie* vyberieme *„Rovnica“*

 , . (1)

 , . (2),

Napr.:

Začnime rovnicou

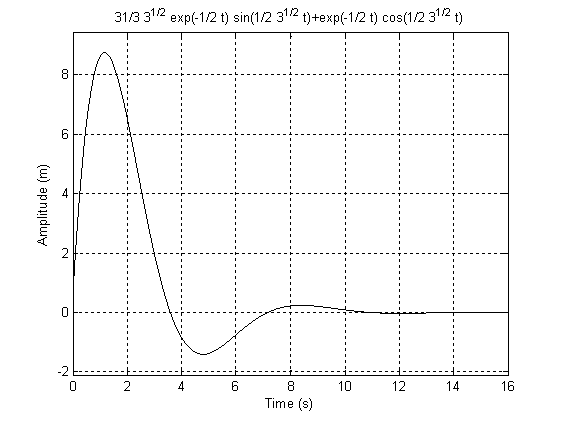
 , . (3)

Grafický priebeh riešenia rovnice (2) vidíme na obrázku (Obr. 2).

* 1. Krížové odkazy

Pomocou funkcie **krížový odkaz** je možné vytvárať odvolávky v texte na kapitoly, podkapitoly , rovnice, vzorce, obrázky, tabuľky a pod. Majú formu: rovnica (2), pozri Katuščák [17], ako sme uviedli v kapitole 2 a pod.

Krížové odkazy je možné vytvoriť pre číslované zoznamy, nadpisy, záložky, poznámky pod čiarou, popisy, číslované odseky a pod. Základom je mať vytvorený číselný zoznam referencií. Potom je možné vytvoriť krížový odkaz na tieto referencie, pričom keď sa zmení poradie referencií v zozname, Word vie automaticky aktualizovať čísla referencií. **Aktualizácia krížových odkazov v celom dokumente sa vykoná cez klávesy CTRL + A potom F9.**



Obr. 17

1. Záver

Záver by mal zachytiť jasnú a presnú prezentáciu dedukcií vychádzajúcich z jadra práce. Musí byť vecnou sumarizáciou vlastného prínosu alebo pohľadu na riešenú problematiku. Zahrnúť možno aj kvantitatívne údaje, ale podrobnosti by sa nemali uvádzať. Záver nemá obsahovať nič, čo nie je v texte práce a musí nadväzovať na úvahy a argumenty v texte práce.

V závere je vhodné poukázať na ďalšie otvorené (doteraz nevyriešené) problémy, ktorým je vhodné venovať pozornosť a ktoré presahujú odporúčaný rozsah diplomovej práce. Odporúčané sú popisy ďalších navrhovaných aktivít, ktoré priamo vyplývajú zo záverov alebo skúseností získaných v priebehu spracovania práce.

Zoznam použitej literatúry

Všetky dokumenty, ktoré v práci použijete, je potrebné zoradiť do zoznamu pozostávajúceho z  bibliografických odkazov (b. o., en bibliographic reference), ktorý označujeme napr. **Zoznam použitej literatúry**. Pre tvorbu zoznamov použitej literatúry platia štandardy. Cieľom je, aby zo zoznamu použitej literatúry bolo možné jasne identifikovať použitý zdroj a aby ho bolo možné bez ťažkostí opäť vyhľadať.

Hlavným zdrojom údajov pre tvorbu bibl. odkazov je **titulný list** (tzn. prvý list v knihe, kde sú uvedené údaje o názve autorovi atď.), príp. jeho rub. Odkazy sa môžu týkať knižných, časopiseckých a iných zdrojov informácií (zborníky z konferencií, patentové dokumenty, normy, odporúčania, kvalifikačné práce, osobná korešpondencia a rukopisy, odkazy cez sprostredkujúci zdroj, elektronické publikácie), ktoré boli v práci použité.

**Technika citovania** určuje spôsob, akým označujeme citácie v dokumente, pričom podľa normy (pozri STN ISO 690 [6]) existuje viacero spôsobov citovania:

* metóda číselných citácií (citácie umiestňujeme v práci podľa odkazových čísel (číslo zo Zoznamu použitej literatúry), ktoré zodpovedajú poradiu citácií v texte),
* citácie v poznámkach,
* metóda prvého údaja a dátumu. (citácie umiestňujeme v práci abecedne podľa prvého údaja (meno autora + rok vydania)).

Pri metóde **číselných citácií** sa v zozname bibliografických odkazov každé citované dielo uvádza v tom poradí, v akom bolo uvedené a číslované v texte. Číslované odkazy v texte sú uvedené v zátvorkách a odkazujú na dokumenty v takom poradí, v akom sa citujú po prvýkrát. Nasledujúce citácie dostávajú také isté číslo, ako má prvá citícia. Ak sa citujú osobitné časti dokumentu, môžu sa za číslom citácie uviesť čísla strán. Metóda číselných citácií je podrobne popísaná v norme STN ISO 690 (pozri [6]).

**Príklad zoznamu použitej literatúry je uvedený na nasledujúcej strane.**

1. MIHALÍK, Ján – ZAVACKÝ, Jozef – GLADIŠOVÁ, Iveta: Signály a sústavy : Návody na cvičenia. Košice : TU-FEI, 2004. 241 s. ISBN 80-8073-138-1
2. CIMBALA, Roman - BALOGH, Jozef - DŽMURA, Jaroslav: Diagnostika výkonových transformátorov s využitím prvkov umelej inteligencie 1. In: Elektrotechnický magazín ETM. roč. 14, č. 1 (2004), s. 8-9.
3. KOVAĽAKOVÁ, Mária - NOVÁK, Ladislav - STANČÁKOVÁ, Anna: Vplyv prímesi chrómu na proces hydrogenácie a dehydrogenácie FeB amorfných zliatin. In: 13. konferencia slovenských fyzikov : Zborník príspevkov. Košice : Slovenská fyzikálna spoločnosť, 2004. s. 145-146.
4. Therion Biologics Corporation, Cambridge, MA: Recombinant fowlpox virus and recombination vector. Inventors: Cohen L. K., Panicali; D. L. Int. Cl.5 C12N/701 United States Patent, 5093258. 03.03. 92.
5. ISO 690-2: 1997, Information and documentation – Bibliographic references - Part 2: Electronic documents or parts thereof.
6. STN ISO 690:1998 : Dokumentácia - Bibliografické odkazy - Obsah, forma a štruktúra.
7. Zákon č. 183/2000 Z.z. o knižniciach, o doplnení zákona Slovenskej národnej rady č. 27/1987 Zb. o štátnej pamiatkovej starostlivosti a o zmene a doplnení zákona č. 68/1997 Z.z. o Matici slovenskej.
8. Vyhláška č. 131/1997 Zb. Ministerstva školstva Slovenskej republiky zo 7. mája 1997 o doktorandskom štúdiu.
9. LAGOZE, C. a kol. The Open Archives Initiative Protocol for Metadata Harvesting [online]. Protocol Version 2.0 of 2002-06-14. Document Version 2004/10/12T15:31:00Z 2004 [cit. 2004-11-10]. Dostupné na internete: <http://www.openarchives.org/OAI/openarchivesprotocol.html>.
10. Elektronické diplomové a dizertačné práce SR: ETD SK. [online]. Košice : ETD SK, 2004. Aktualizované 14-2-2005 [cit 2005-03-10]. Dostupné na internete: <http://www.etd.sk/>.
11. UNESCO. The Guide to Electronic Theses & Dissertations [online]. Paris : UNESCO, c2001 [cit 2004-11-10]. Dostupné na internete: <http://etdguide.org/>.
12. HOGGAN, Daniele. 2002. Challenges, Strategies, and Tools for Research Scientists. In Electronic Journal of Academic and Special Librarianship [online]. 2002, vol. 3, no. 3 [cit. 2003-01-10]. Dostupné na internete: <http://southernlibrarianship.icaap.org/content/v03n03/Hoggan\_d01.htm>. ISSN 1525-321X
13. KOMOROVÁ, K. Výstava vzácnych kódexov. In Knižnica [online]. Martin : SNK, 2002 [cit. 2003-02-14], 2002, roč. 3, č. 2, s. 84. Dostupné na internete: <http://www.snk.sk/kniznica/kniznica.html>. ISSN 1212-5075
14. PARKER, Elliott. Re: Citing Electronic Journals. In : PACS-L (Public Access Computer Systems Forum) [online]. Houston (Tex.) : University of Houston Libraries, 24 November 1989; 13:29:35 CST [citované 2003-01-05]. Dostupné na internete: <telnet://brsuser@a.cni.org>.
15. BURAN, Daniel. 2003. Environmentálne informačné zdroje a služby v strednej a východnej Európe [elektronická pošta]. Správa pre: Mária MALÁ. 2002-11-15 [cit. 2003-01-05]. Osobná komunikácia.
16. GONDA, Vladimír: Ako napísať a úspešne obhájiť diplomovú prácu. Bratislava : Elita, 2003. 124 s. : il. ISBN 80-8044-076-X
17. Katuščák, Dušan : Ako písať záverečné a kvalifikačné práce. Nitra: Enigma, 2004. 162 s. il. ISBN 80-89132-10-3
18. <http://developer.android.com/guide/developing/tools/index.html>
19. <http://developer.android.com/guide/developing/tools/adt.html>
20. <http://developer.android.com/guide/developing/tools/adb.html>
21. <http://developer.android.com/guide/developing/devices/index.html>
22. Andbook
23. <http://developer.android.com/guide/topics/fundamentals/activities.html>
24. <http://developer.android.com/guide/topics/fundamentals/services.html>
25. <http://developer.android.com/guide/topics/fundamentals.html>
26. <http://developer.android.com/guide/topics/intents/intents-filters.html>
27. <http://developer.android.com/guide/topics/ui/index.html>
28. <http://images.fonearena.com/blog/wp-content/uploads/2011/11/Google-Plus.jpg>
29. <http://developer.android.com/images/system-architecture.jpg>
30. <http://developer.android.com/guide/basics/what-is-android.html>
31. <http://www.openhandsetalliance.com/press_110507.html>
32. <http://developer.android.com/images/activity_lifecycle.png>
33. <http://www.denso-wave.com/qrcode/aboutqr-e.html>
34. <http://www.openhandsetalliance.com/oha_faq.html>
35. KNIHA <http://books.google.sk/books?id=fG8JUdrScsYC&lpg=PP1&dq=Handbook+of+Augmented+Reality&pg=PA341&redir_esc=y#v=onepage&q=Handbook%20of%20Augmented%20Reality&f=false>
36. <http://www.saxproject.org/event.html>

Prílohy

1. CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe.
2. Používateľská príručka
3. Systémová príručka

Táto časť diplomovej práce je povinná a obsahuje zoznam všetkých príloh vrátané elektronických nosičov. Názvy príloh v zozname musia byt’ zhodné s názvami uvedenými na príslušných prílohách. Tlačené prílohy majú na prvej strane identifikačné údaje – informácie zhodné s titulnou stranou diplomovej práce doplnené o názov príslušnej prílohy (Systémová príručka, Používateľská príručka). Identifikačné údaje sú aj na priložených diskoch alebo disketách. Ak je médií viac, sú označené aj číselne v tvare I/N, kde I je poradové číslo a N je celkový počet daných médií.

Každá príloha začína na novej strane a je označená samostatným písmenom (Príloha A, Príloha B, ...). Číslovanie strán príloh nadväzuje na číslovanie strán v hlavnom texte.

Curriculum vitae

Táto časť je nepovinná. Autor tu môže uviesť svoje biografické údaje, údaje o záujmoch, účasti na projektoch, účasti na súťažiach, získané ocenenia, zahraničné pobyty na praxi, domácu prax, publikácie a pod.