



Ekonomická
fakulta
Faculty
of Economics

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

Jihočeská univerzita v Českých Budějovicích
Ekonomická fakulta
Katedra aplikované matematiky a informatiky

Bakalářská práce

Porovnání katalogizačních informačních systémů Clavius a Tritius

Vypracoval: Jan Šimeček
Vedoucí práce: Mgr. Radim Remeš

České Budějovice 2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan ŠIMEČEK**
Osobní číslo: **E11413**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Ekonomická informatika**
Název tématu: **Porovnání katalogizačních informačních systémů Clavius a Tritius**
Zadávací katedra: **Katedra aplikované matematiky a informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Předmětem práce je s použitím vhodné metodiky provést porovnání knihovních katalogizačních systémů Clavius a Tritius. Bude vytvořena aplikace pro konverzi dat mezi oběma systémy.

Metodický postup:

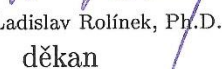
1. Studium odborné literatury.
2. Obecný popis používaných knihovních systémů, principy ukládání dat a jejich formáty.
3. Porovnání a analýza vybraných produktů, zhodnocení jejich použitelnosti v reálném prostředí.
4. Popis implementace vlastního řešení.

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: 40 - 50 stran
Forma zpracování bakalářské práce: tištěná
Seznam odborné literatury:

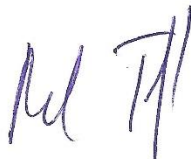
1. **BALÍKOVÁ, Marie.** *Katalogizace ve formátu MARC 21: stručná instrukce a příklady pro knihy a některé typy pokračujících zdrojů. 2., aktualiz. vyd.* Praha: Národní knihovna České republiky, 2008, 163 s. ISBN 978-80-7050-539-7.
2. **MINTY, Wilma.** *AACR2R a UNIMARC: příručka pro praktickou katalogizaci s příklady. 1. české vyd.* Praha: Národní knihovna ČR, 1998, [500] s. ISBN 80-705-0323-8.
3. **PECINOVSKÝ, Rudolf.** *OOP: naučte se myslet a programovat objektivě.* Vyd. 1. Brno: Computer Press, 2010, 576 s. ISBN 978-80-251-2126-9.
4. **UNIMARC to MARC 21 Conversion Specifications. LIBRARY OF CONGRESS.** *Library of Congress [online]. Version 3.0. Washington, D.C.: Library of Congress, August 2001, 09/08/2011 [cit. 2013-03-09]. Dostupné z: <http://www.loc.gov/marc/unimarc21.html>*
5. *Záznam pro souborný katalog UNIMARC: tištěné monografie. 1. vyd.* Praha: Národní knihovna ČR, 1996, 31 s. ISBN 80-705-0237-1.

Vedoucí bakalářské práce: **Mgr. Radim Remeš**
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: **2. ledna 2013**
Termín odevzdání bakalářské práce: **15. dubna 2014**


doc. Ing. Ladislav Rolínek, Ph.D.
děkan

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
EKONOMICKÁ FAKULTA
Studentská 13 (26)
370 05 České Budějovice


prof. RNDr. Pavel Tlustý, CSc.
vedoucí katedry

V Českých Budějovicích dne 29. března 2013

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

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to – v nezkrácené podobě/v úpravě vzniklé vypuštěním vyznačených částí archivovaných Ekonomickou fakultou - 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 a 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.

Datum

Podpis studenta

Děkuji vedoucímu bakalářské práce Mgr. Radimu Remešovi
za cenné rady, připomínky a metodické vedení práce.

Anotace

Tato práce se snaží popsat informační systémy Clavius a Tritius. Dále se zde rozebírá problematika bibliografických formátů MARC21, UNIMARC a Bibframe. Pomocí těchto poznatků bude vytvořena aplikace pro převod záznamů mezi Claviem a Tritiem. Konkrétně mezi formáty MARC21 a UNIMARC.

OBSAH

1.	ÚVOD.....	9
2.	DEFINICE POJMŮ.....	10
3.	METODIKA.....	11
4.	KATALOGIZAČNÍ INFORMAČNÍ SYSTÉMY	12
4.1.	Význam a současná situace na trhu	12
4.2.	Clavius	12
4.3.	Tritius	16
4.4.	Aleph.....	21
4.5.	Evergreen	21
4.6.	Srovnání systémů Clavius a Tritius	22
5.	BIBLIOGRAFICKÉ FORMÁTY	24
5.1.	Historie	24
5.2.	Struktura záznamu v MARC21/UNIMARC.....	24
5.3.	MARC21.....	25
5.4.	UNIMARC	26
5.5.	Bibframe a budoucnost MARC21	26
6.	STANDARDY, METODICKÉ POSTUPY	32
6.1.	Norma ISO 2709	32
6.2.	International Standart Bibliographic Description (ISBD).....	32
6.3.	Angloamerická katalogizační pravidla (AACR2R)	33

7. PŘEVODNÍK	34
7.1. Úvod.....	34
7.2. Analýza.....	34
7.2.1. Jazyk Java	34
7.3. Návrh.....	35
7.3.1. UML class diagram.....	35
7.3.2. Vývojový diagram	36
7.4. Implementace.....	37
7.4.1. Metoda kodToM21().....	37
7.4.2. Metoda sort().....	38
7.4.3. Uživatelské rozhraní převodníku	39
7.4.4. Algoritmus převodníku	40
7.5. Test.....	45
8. ZÁVĚR	46
SUMMARY AND KEYWORDS	47
SEZNAM LITERATURY	48
SEZNAM OBRÁZKŮ A ZDROJOVÝCH KÓDŮ	
SEZNAM PŘÍLOH	
PŘÍLOHY	

1. ÚVOD

Katalogizační systémy jsou v dnešní době stále více potřebné než dřív. V minulosti se k uchovávání informací o katalogizovaných předmětech (ať už se jedná o knížky, součástky nebo cokoli jiného) používaly papírové kartičky. Samozřejmě to mělo mnoho nevýhod, jako třeba poškození kartičky nebo její ztráta. To vedlo k znehodnocení informací na ní. Tento problém právě vyřešily katalogizační systémy. Všechna data lze lehce spravovat skrze uživatelské rozhraní, které nabízejí. Pokud dotyčná instituce bude dodržovat i správná pravidla zálohování dat, nemůžou se už nikdy data znehodnotit a nenastane chaos v katalogizaci. Není to jen ulehčení pro zaměstnance, kteří se systémem pracují, ale i pro zákazníky. Tato práce se bude zabývat knihovnamí, takže to budou návštěvníci a zaměstnanci knihovny. Je pro ně mnohem snazší si vyhledat, popřípadě zamluvit knihy v klidu z domova přes internet než přímo v knihovně. Proto se práce zaměří především na katalogy.

Tato práce se zabývá konkrétně knihovními informačními systémy Clavius a Tritius. Budou zde zmíněny jejich rozdíly, výhody a nevýhody. Budou zde i okrajově zmíněny zahraniční softwary pro lepší srovnání.

V další části zde bude rozebrána problematika bibliografických formátů. To z důvodu pochopení uchovávání informací a jejich následného převodu mezi formáty. Podrobně zde budou zmíněny nejpoužívanější bibliografické formáty MARC21 a UNIMARC. Pro lepší porozumění dané problematice zde budou zmíněny i standardy, které ukládání bibliografických formátů upravují.

Praktická část bakalářské práce obsahuje aplikaci pro převod knižních záznamů mezi formáty MARC21 a UNIMARC. Tato aplikace vznikla na požadavek firmy Lanius s. r. o., protože systém Tritius je primárně nastavený na MARC21.

Před teoretickou částí je umístěn slovník použitých pojmů a metodika. Tyto základní pojmy jsou nezbytné pro pochopení problematiky základů programování v praktické části. Metodika popisuje postup při zpracování literatury k dané problematice, návrh a následnou implementaci převodníku.

V závěru práce se nacházejí výsledky a zhodnocení předem stanovených cílů.

2. DEFINICE POJMŮ

UML – inženýrský, grafický jazyk používaný pro tvorbu návrhů a dokumentaci softwaru.

Metody – jsou nositeli dovedností objektů.

Třída (class) – skupina objektů, které mají stejné vlastnosti.

Objekt – instance třídy.

Vlastnosti objektů – jsou to různé atributy, charakteristiky objektů.

Pole – objekt, který poukazuje na množinu jiných objektů, které jsou v něm uloženy.

String – řetězec znaků.

Int – celá čísla.

Cyklus – smyčka s různým opakováním.

GUI – grafické uživatelské rozhraní.

If – podmínka ve zdrojovém kódu.

Open Source – volně šiřitelný software, kterému lze libovolně upravovat zdrojový kód.

3. METODIKA

Při tvorbě této práce se bude postupovat podle metodického postupu rozepsaného v bodech uvedených níže.

1. Studium teorie

Na začátku práce bude v teoretické části zpracována literární rešerše k danému tématu, která je nezbytná k pochopení řešeného problému.

2. Analýza požadavků na aplikaci a stanovení cílů

Praktická část práce začne analýzou požadavků na výslednou aplikaci. Důraz bude kladen na požadované vstupy a výstupy.

3. Návrh aplikace

Po analýze bude následovat návrh optimální struktury a algoritmu aplikace. Pro lepší přehlednost budou využity diagramy.

4. Implementace aplikace

Až po hotovém návrhu bude následovat samotná implementace aplikace převodník. Pro naprogramování aplikace bude použit jazyk Java.

5. Test aplikace

Dokončená aplikace bude podrobena testu, který prokáže provozuschopnost převodníku.

6. Zhodnocení výsledků v závěru práce

Nakonec praktické části bude zhodnocen výsledek testů a splnění předem stanovených cílů.

4. KATALOGIZAČNÍ INFORMAČNÍ SYSTÉMY

4. 1. Význam a současná situace na trhu

Katalogizační systémy mají za úkol vést evidenci a uživatelům poskytovat lepší orientaci a tím také usnadnit jejich práce. Práce se zaměří konkrétně na knihovny, tedy na knihovní systémy. Tyto systémy spravují nejen knihy, ale i jiné materiály jako jsou např.: plány, časopisy atd.

V současné době je na trhu k dispozici velký výběr produktů. Nejdůležitějším odlišením je jejich cena a samozřejmě i množství poskytovaných funkcí. Ceny se pohybují ve velikém rozmezí. Některé jsou dokonce open source. Ty jsou obzvláště výhodné pro nenáročného uživatele (knihovny).

4. 2. Clavius

Obecná charakteristika a historie:

Knihovní systém Clavius začala firma Lanius s. r. o. vyvíjet již v roce 1998. Na první pohled by se mohlo zdát, že je tento systém následovníkem systému Lanius od stejné firmy, který se pouze převedl na Windows. Pravda je opakem. Clavius je samostatně vyvíjený knihovní systém, který se vyvíjel od nuly za pomoci tehdy moderních technologií. Před začátkem vývoje byl kladen důraz na to, aby nový software hned od úplného začátku nahradil nedostatky Lania. Myslelo se především na budoucnost, aby software nezastaral hned v následujících letech. Z toho důvodu Clavius zajišťuje: neomezenou délku vstupních polí, ukládání dokumentů do společné databáze, je připraven na změnu knihovních pravidel, je připravena měna Euro, spolupracuje s aplikacemi z balíčku Microsoft Office, nabízí tvorbu formulářů pro vstup a výstup, umožňuje využít různých možností vyhledávání v dokumentech, čtení čárových kódů, nově je zde přidáno i čtení RFID čipů nebo možnost rozesílání různých emailů čtenářům. Data se ukládají buď ve formátu UNIMARC nebo MARC21. Clavius podporuje pravidla AACR2 a normu ISBD. (LANius, n. d.)

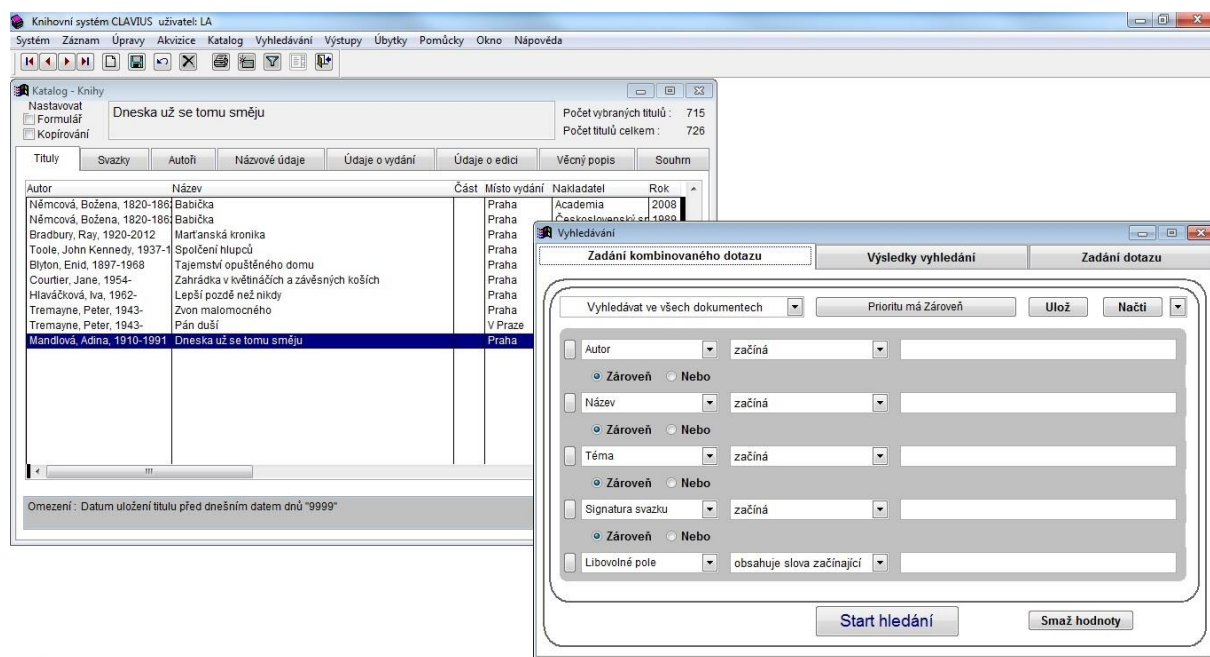
Moduly:

Clavius je rozdělen do 17 částí (modulů):

Nejdůležitější částí celého systému je katalogizace. Do systému lze ukládat monografii, hudebniny, periodika atd. Pro každý typ dokumentu jsou předdefinované formuláře. Pole i podpole lze upravovat nebo vytvářet svá vlastní. Při ukládání systém kontroluje vyplněná pole. Vše je nastaveno na formáty UNIMARC/MARC21 s pravidly AACR2. (LANius, n. d.)

Před spuštěním konkrétního katalogu se vybírá, pro který druh dokument se má katalogizace otevřít. Data jsou rozdělena do záložek podle druhu dokumentu. Vpravo nahoře lze vidět počet titulů.

Obrázek 1: Clavius (katalog a vyhledávání):



Na obrázku 1 je vidět katalogizace Clavia. Nahoře se nachází hlavní menu. Vlevo je vidět otevřená katalogizace a napravo vyhledávání. Zvětšený obrázek je uložen jako příloha 4.

Akvizice řeší problematiku nákupu fondu. Zde se nadefinují odběratelé a dodavatelé. Požadované dokumenty se ukládají do hlavní databáze a jsou dostupné i z OPAC. Objednávky modul tiskne nebo posílá elektronickou poštou. Všechna data se průběžně ukládají, proto je vše i zpětně k dispozici např.: faktury a statistiky. Modul sám sleduje, které objednávky jsou urgentní a včas uživatele upozorní. (LANius, n. d.)

Evidence periodik je integrována do modulu katalogizace. Díky tomu lze efektivněji evidovat periodika. (LANius, n. d.)

Audiovizuální média a elektronické zdroje je modul, který rozšiřuje katalogizaci, výpůjční protokol a vyhledávání o gramofonové desky, magnetofonové pásky, CD, DVD, elektronické zdroje a mnoho dalších. (LANius, n. d.)

Modul vyhledávání (OPAC) je určený pro pracovníky knihovny k okamžitému dotazu na celou bázi různých druhů dokumentů. Modul obsahuje kombinované vyhledávání, do kterého lze napsat všechny ukládané údaje nebo třeba jen jeden. Výsledky lze řadit anebo označovat pro lepší orientaci. Tyto výsledky lze exportovat např.: jako tiskové výstupy nebo uložit do souboru HTML, ISO, XML. (LANius, n. d.)

Www katalog pracuje na aplikačním serveru. Jeho účelem je on-line přístup k databázi dokumentů knihovny. Přes www katalog lze provádět rezervace, spravovat čtenářské konto, prodlužovat výpůjčky atd. Toto řešení je velice výhodné hlavně kvůli minimálním nárokům na hardware a zatížení sítě. Od roku 2010 je k dispozici katalog Carmen, který je modernější verzí www katalogu. (LANius, n. d.)

Obrázek 2: www katalog - Clavius:

Městská knihovna Tábor - katalog Clavius

ON-LINE katalog - Zadání dotazu :

Často kladené dotazy

Vyhledávat : ve všech dokumentech 2

Autor :

Název :

Téma (PH+kl slova) :

Signatura svazku :

Libovolné pole :

Hledej Smaz Kombinovaný dotaz

Instrukce : Do připravených polí formuláře zadejte hledaný termín a stiskněte tlačítko HLEDEJ. Pokud vyplníte více řádek, vyhledané výsledky budou splňovat všechna kritéria zároveň. Je možné použít pravostranného kráčení slov (nedoplníte slova znakem " nebo ?). Kliknutím na nadpis pole si zobrazíte rejstřík termínů použitých v tomto poli. Můžete též použít formulář pro [Kombinovaný dotaz](#).

Další možnosti : [Seznamy a novinky](#), [Vaše čtenářské konto](#), [Předregistrace čtenáře](#), [Návrat na hlavní stránku](#)

Další možnosti vyhledávání : [Souborný katalog naučné literatury](#), [Souborný katalog článků](#), [Souborný katalog regionu](#)

Katalogy knihoven REgionálního Knihovního Systému Tábor :

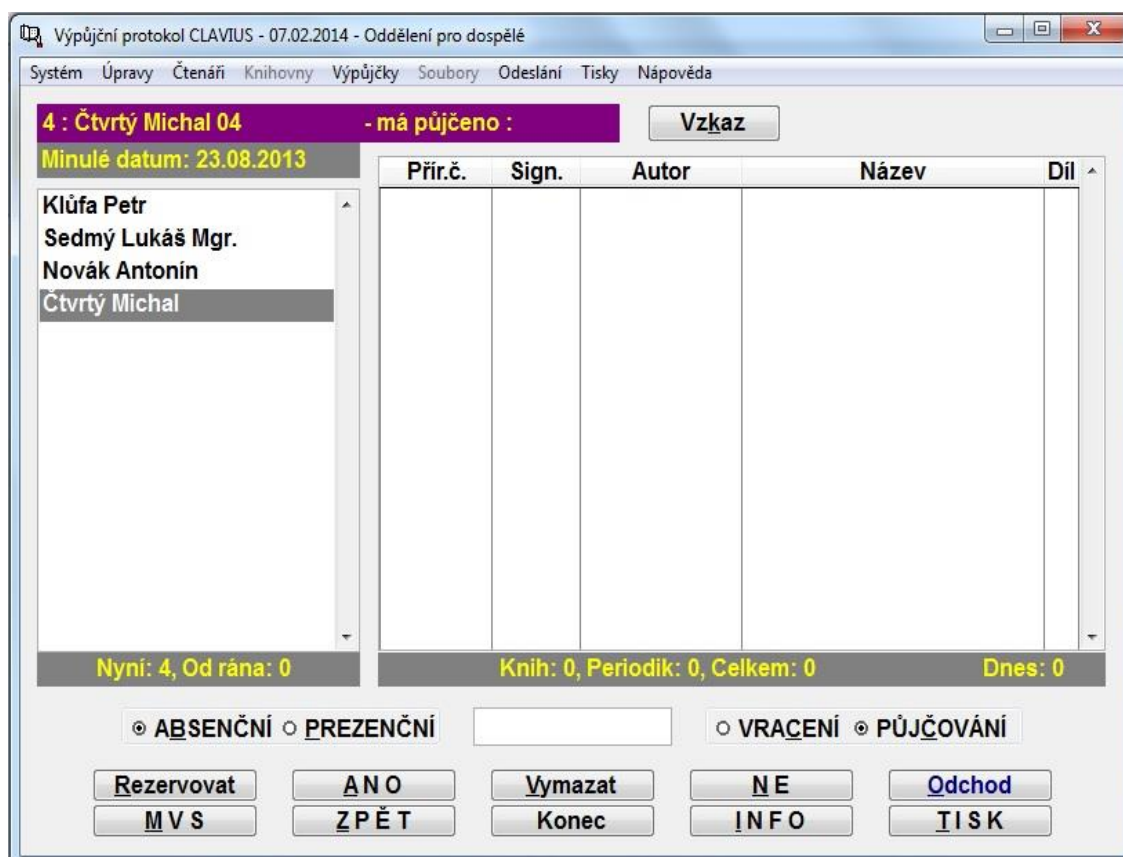
[Borkovice](#), [Borotín](#), [Brandlín](#), [Březnice](#), [Budislav](#), [Dírná](#), [Dobronice u Chýnova](#), [Dráčov](#), [Dražice](#), [Choustník](#), [Klenovice](#), [Komařov](#), [Košice](#), [Mlýny](#), [Myslkovice](#), [Nadějkov](#), [Nemysl](#), [Pořín](#), [Přebořov](#), [Radení](#), [Radčice](#), [Radimovice u Tábora](#), [Radimovice u Želče](#), [Rataje](#), [Ratibořská Hory](#), [Roudná](#), [Řepeč](#), [Řípec](#), [Skalice](#), [Slápy](#), [Smilov Hory](#), [Stádlec](#), [Sudoměřice u Bechyně](#), [Sudoměřice u Tábora](#), [Sviny](#), [Vesec](#), [Vilice](#), [Vlastibůf](#), [Vodice](#), [Žalší](#), [Zvěrotice](#), [Želč](#)

Hleďte ve Vaší knihovně

Na obrázku je vidět www katalog Clavia. Tento katalog využívají převážně návštěvníci knihoven k hledání dokumentů. V centru se nachází formulář, kde uživatel vyplní pole podle, kterých chce nalézt dokument. Ve spodní části se nachází seznam knihoven, který jsou propojeny pomocí REKS (regionální knihovní systém). Zvětšený je k vidění v příloze 5.

Výpůjční protokol je primárně určen pro evidenci čtenářů, jejich výpůjček a poplatků. Uchovává si údaje o tom, kdo měl konkrétní titul půjčený. Je zde i funkce, která umožňuje upozornění čtenářů na připravenou rezervaci pomocí emailu. Výpůjční protokol také uchovává statistiky o výpůjčkách a o tom, které tituly jsou nejžádanější. (LANius, n. d.)

Obrázek 3: výpůjční protokol:



Na obrázku 3 je zobrazen výpůjční protokol. Nahoře se nachází hlavní menu, kde uživatel může změnit nastavení nebo nalézt méně využívané funkce jako např.: anonymní vrácení. V centru obrazovky jsou vidět 2 menší okna. V okně vlevo se zobrazují návštěvníci, kteří právě přišli do knihovny. Větší, pravé okno, zobrazuje dokumenty, které má návštěvník půjčené. Zaškrtnutím vrácení/půjčení se dokumenty buď přidají, nebo odeberou. Ve spodní části se nachází menu, které nabízí nejdůležitější funkce pro práci s výpůjčním protokolem např.: rezervovat, tisk nebo odchod.

Dětský online katalog ulehčuje využití systému dětem, které ještě nechodí do školy, protože ovládání katalogu zde probíhá pomocí obrázků. Ty nabízejí nejžádanější žánry. Pro vyspělejší je zde plnohodnotné vyhledávání. (LANius, n. d.)

Dispečink budovy usnadňuje evidenci přístupu uživatelů k síti. Obsluha knihovny díky tomu ví, jak dlouho je čtenář připojený na internetu. Po vypršení času je obsluha upozorněna a program vypočítá poplatek a vytiskne účtenku. Dispečink si uchovává statistiky využití. (LANius, n. d.)

Revize modulu slouží k revizi fondu daného oddělení. Za pomoci revizních scannerů a čárových kódů, tak lze snadno a rychle zjistit, které knihy chybí, následně vytisknout seznam a knihy odepsat. (LANius, n. d.)

Sdílená katalogizace během 4 hodin odesílá nový záznam do ostatních připojených knihoven. Tím je omezena nutnost často vyplňovat údaje u nové knihy. (LANius, n. d.)

Výměnné soubory slouží pro veřejné knihovny, které nechávají obíhat část svého fondu po okolních přidružených knihovnách. Modul eviduje všechny druhy dokumentů. Také umožňuje rezervaci konkrétního dokumentu pro určitou knihu. (LANius, n. d.)

EMVS posílá upomínky, oznámení o rezervaci e-mailem. (LANius, n. d.)

Z39.50 client umožňuje vyhledávání ve vzdálených Z39.50 serverech. Lze stahovat bibliografické a autoritní záznamy, které lze přidat do vlastní knihovny. (LANius, n. d.)

Z39.50 server umožňuje vystavovat fond knihovny pro ostatní. Server běží na Win32 serverech nebo Linux. (LANius, n. d.)

OAI provider se liší od Z39.50 serveru tím, že sám čeká až ho OAI harvester požádá o záznamy. Nejčastěji odpovídá na dotazy typu: co je od kdy nového. Záznamy jsou odesílány ve formátu XML a to buď jako MARC21 nebo UNIMARC. (LANius, n. d.)

OAI harvester stahuje data od OAI providera a stažená data ukládá do zvláštní licence Clavius. (LANius, n. d.)

4. 3. Tritius

Obecná charakteristika:

„Tritius je katalogizační informační systém nové generace. Je navržen pro evidenci jakýchkoli dokumentů v jakékoli sbírce. Tritius je také úplný knihovní systém. Je

nástupcem systému Clavius - nejrozšířenějšího knihovního systému v ČR a SR.“
(Effectiva Solutions, © 2012)

Tritius je webová aplikace, která funguje na všech možných zařízeních bez ohledu na hardware či operační systém. Z toho vyplývá, že Tritius funguje i na moderních přenosných zařízeních jako jsou tablety, chytré telefony atd. a to bez jakékoliv složité instalace. Jedinými požadavky jsou připojení k internetu a webový prohlížeč. (Effectiva Solutions, © 2012)

Hlavní výhody tedy jsou:

1. nenáročnost na hardware
2. stále aktuální software
3. žádná instalace

Tritius je přizpůsobivý, protože v něm může evidovat jakékoliv dokumenty. Není pevně dáno zaměření na knihovní dokumenty. Lze tedy evidovat i technickou dokumentaci, plány atd. (Effectiva Solutions, © 2012)

Nový systém Tritius díky Unicode podporuje všechny národní abecedy. Lze si vybrat od latinky až po čínštinu. (Effectiva Solutions, © 2012)

„Při každodenní práci je komfort uživatelského rozhraní klíčovou vlastností každého systému. Proto jsme do Tritia implementovali i pokročilé funkce, které se standardně ve webových aplikacích neobjevují - zejména z důvodu náročnosti jejich implementace. Systém podporuje klávesové zkratky, pohyb pomocí klávesnice, rychlé dohledávání psaním hledaného výrazu v tabulkách a další.“ (Effectiva Solutions, © 2012)

“Systém Tritius je od počátku navržen pro práci více samostatných organizačních jednotek v jedné databázi. Organizační jednotkou může být knihovna, nebo ve firemní sféře pobočka či pracoviště. Umožňuje sdílení dat všech organizačních jednotek bez ohledu na jejich sídlo.“ (Effectiva Solutions, © 2012)

Tritius podporuje komunikaci s jinými systémy: SOAP, WSDL a XML. Systém podporuje knihovní standarty: MARC21, FRBR, RDA a OAI-PMH.

Systém sice podporuje MARC21, ale lze do budoucna nadefinovat i jakékoliv jiné MARC formáty. (Effectiva Solutions, © 2012)

Tritius by nebyl moderní, kdyby nepodporoval i moderní technologie pro práci s dokumenty. Proto se od začátku systém vyvíjel tak, aby byl schopen pracovat s RFID čipy. (Effectiva Solutions, © 2012)

Základní znaky Tritia tedy jsou:

1. přizpůsobivost
2. unicode
3. uživatelské rozhraní
4. strukturovanost
5. podpora standardů
6. RFID technologie (Effectiva Solutions, © 2012)

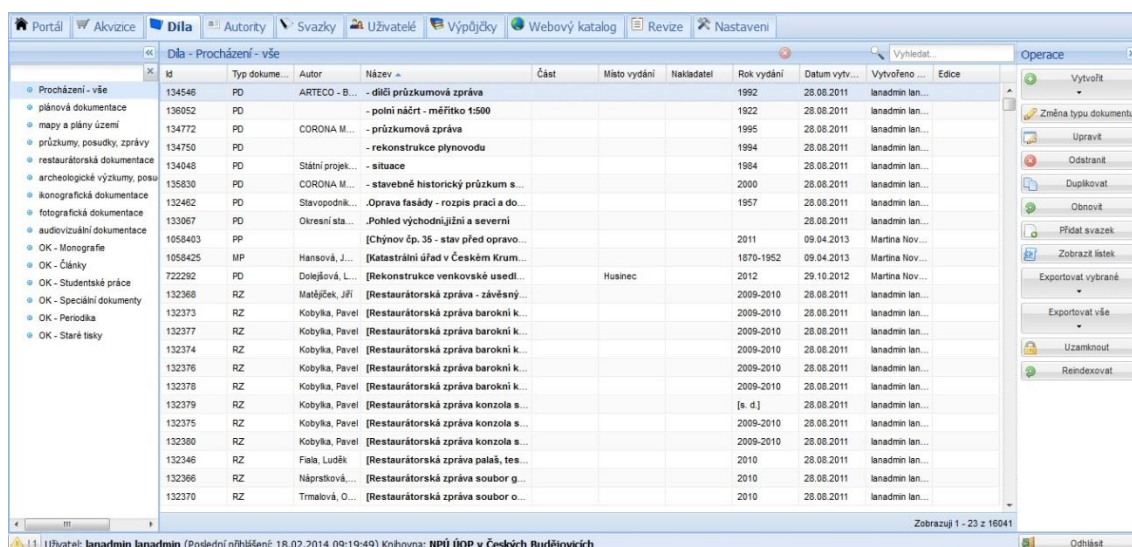
Moduly:

Momentálně má Tritius 8 částí:

Pro zahájení práce s Tritiem se musí uživatel nejdříve přihlásit.

Základním a nejdůležitějším modulem stejně jako u Clavia je katalogizace. Při vytváření nového dokumentu se vyplňují pole ve formátu MARC21 a pravidly AACR2. Tato pole lze libovolně měnit. Nacházejí se tady pole jako autor, název díla, místo vydání, rok vydání a mnoho dalších. Formuláře jsou předem předdefinované na tyto typy děl: monografie, AV média, hudebniny, periodika a analytické popisy článků periodik. Formuláře katalogu mají řadu funkcí, které ulehčují práci při tvorbě záznamu. Každé pole nabízí nápovědu s příklady. Tritius neustále kontroluje, jestli uživatel vyplnil pole správně. Při ukládání záznamu Tritius zkontroluje, jestli jsou vyplněna všechna povinná pole. U každého záznamu se ukládají svazkové údaje pro pozdější kopírování. Exempláři lze přiřadit čárový kód pro výpůjční protokol. (Effectiva Solutions, © 2012)

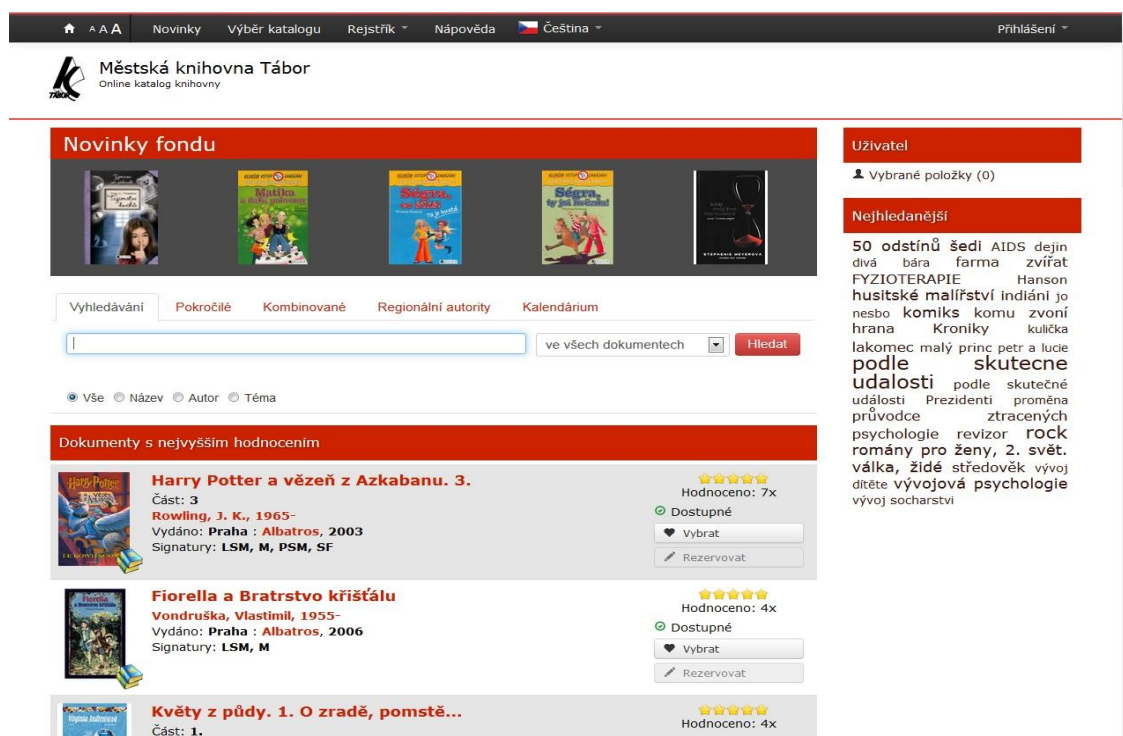
Obrázek 4: Tritius - katalogizace:



Na obrázku 4 je vidět systém Tritius, konkrétně modul katalog. Nahoře se nachází menu, ve kterém se uživatel pohybuje v rámci jednotlivých modulů a nastavení. Nalevo jsou vidět kategorie děl, jako jsou monografie, články, plánová dokumentace atd. Po rozklepnutí kategorie se zobrazí díla, která sem patří. Tato díla se zobrazují uprostřed viz obrázek. Napravo od děl je menu operace, kterým se pracuje s vypsáními položkami. Jsou zde funkce jako vytvoření nového díla nebo jeho odstranění. Vpravo nahoře se nachází vyhledávání. Nastavení vyhledávání se vyvolá kliknutím na lupu. Pro ukončení přihlášení se používá tlačítko odhlásit vpravo dole. Pro lepší zobrazení je zvětšený obrázek Tritia zobrazen v příloze číslo 7.

Pro Tritius byl vyvinut katalog Carmen. Tento katalog je kompatibilní i se systémem Clavius. Carmen je webový katalog založený na principech OPAC 2.0. Katalog umožňuje veřejnosti snadno nalézt hledaný exemplář nebo spravovat vlastní uživatelský účet v knihovně. Katalog má moderní a přehledný vzhled, který napomáhá snadné orientaci. Hledání lze provádět rychlým hledáním nebo kombinovaným hledáním. Rychlé hledání vyhledává pouze podle jednoho kritéria, naopak kombinované hledání vyhledává podle různých kritérií najednou, mezi kterými lze volit možnost „A“ a „Nebo“. Výsledky hledání lze řadit podle oblíbenosti, relevance atd. Výsledky lze omezit pomocí řezů. (Effectiva Solutions, © 2012)

Obrázek 5: Katalog - Carmen:



Na obrázku je vidět www katalog Carmen. Napravo se po přihlášení objeví uživatelské menu, kde lze sledovat historii výpůjček, rezervace, poplatky. Lze zde nastavit individuální nastavení jako je např.: barva katalogu, primární způsob vyhledávání a jazyk. Uprostřed katalogu se nachází vyhledávání. To je buď rychlé, kombinované nebo pokročilé. Pod vyhledáváním se pak zobrazují výsledky. Vpravo dole jsou po vyhledání dokumentů vidět omezující řezy. Zvětšený katalog je uložen v Příloze 6.

Výpůjční protokol slouží pro evidenci čtenářů, jejich výpůjček a poplatků. Mezi funkce modulu patří: kategorizace čtenářů, podpora čárových kódů a RFID čipů, rezervace titulů a čtenářů, tisk upomínek a rezervací, statistiky výpůjček, vytváření přehledných výstupů. (Effectiva Solutions, © 2012)

Akvizice umí sama vyhledávat požadované dokumenty v databázích včetně cen. Hlavní funkce akvizice jsou: výtisk objednávky nebo elektronické odeslání, evidence dokladů (faktury), dotažení informací o titulu do katalogizace, kontrola rozpočtu. Čísla objednávek se automaticky generují. (Effectiva Solutions, © 2012)

Revizní modul slouží ke kontrole exemplářů na oddělení. Modul může kontrolovat nejen přítomnost exempláře, ale i jeho správné umístění. Kromě čárových kódů podporuje i RFID čipy. (Effectiva Solutions, © 2012)

Poslední tři moduly jsou Z39.50 client, Z39.50 Server a OAI Provider. Tyto moduly mají stejnou funkci jako u Clavia.(Effectiva Solutions, © 2012)

4. 4. Aleph

Aleph je vyvíjen předním dodavatelem řešení pro knihovny – izraelská firma Ex Libris. Ex Libris má zákazníky ve více než 70 zemích na 6 kontinentech. Tím má Aleph dominantní postavení na trhu. Mezi uživatele Aleph totiž patří i Library of Congress. Jeho cena tomu samozřejmě odpovídá. (Ex Libris, n. d.)

„Hlavním problémem pro získání systému Aleph 500 byla jeho cena, která se v případě velkých knihoven pohybuje až v nemalých jednotkách miliónů Kč (cena systému je odvozena jednak od velikosti jeho bází, jednak podle počtu souběžně pracujících knihovníků a čtenářů).“ (Bartošek, © 2011, 6-10)

Obrázek 6: Aleph - katalog:

The screenshot shows the Aleph library catalog interface. At the top, there is a navigation bar with the title 'Katalog JU' and several menu items: 'Přihlášení', 'Konec', 'Čtenář', 'Nastavení', 'Datažce', 'Dotazy', 'MVS', and 'Nápověda'. Below this is a secondary navigation bar with 'Prohlížení', 'Vyhledávání', 'Výsledky dotazu', 'Předchozí dotazy', and 'Má schránka'. A horizontal line separates these from the main content area. Below the line, there are several search-related links: 'Základní vyhledávání', 'Vyhledávání z více polí', 'Vyhledávání z více bází', 'Pokročilé vyhledávání', and 'CCL'. The main content area is titled 'Základní vyhledávání' and contains a search form. The form has a text input field for 'Zadejte slovo nebo slovní spojení', a dropdown menu for 'Pole pro vyhledávání' (set to 'Všechna pole'), radio buttons for 'Blížkost slov?' (set to 'Ne'), and a dropdown for 'Báze pro vyhledávání' (set to 'Celý katalog JU'). There are 'OK' and 'Vyčistit formulář' buttons. Below the search form, there is a section 'Omezit vyhledávání na:' with dropdowns for 'Jazyk dokumentu:' (set to 'všechny'), 'Rok od:' (empty), 'Rok do:' (empty), and 'Druh dokumentu:' (set to 'všechny'). There is also a 'Uložení dokumentu:' dropdown (set to 'všechny'). At the bottom of the form, there is a link: 'Nenašli jste knihu v našem katalogu? Zkuste ji vyhledat v Jihočeské vědecké knihovně.' Below the form, there is a section 'Tipy pro vyhledávání:' with three bullet points providing search tips.

Na obrázku 6 je vidět www katalog Aleph. Nahoře se nachází hlavní nabídka, kde lze spravovat např.: čtenáře nebo nastavení. Pod menu je základní vyhledávání, které lze rozšířit o funkce, které se nacházejí pod vyhledáváním.

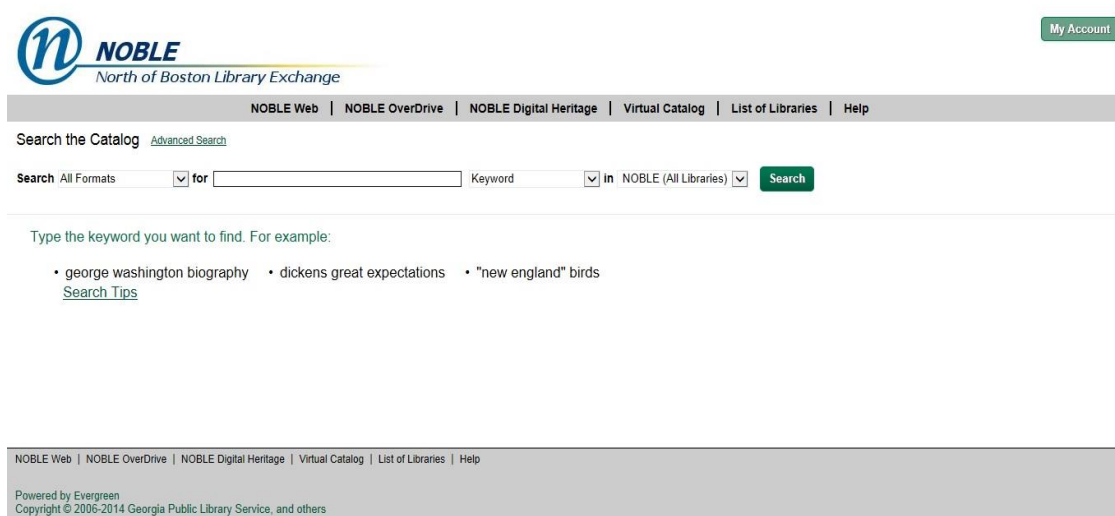
4. 5. Evergreen

Z open source systému stojí za zmínku především systém Evergreen. Evergreen je vyvíjen The Evergreen project. Používá ho přes 1000 knihoven po celém světě. Projekt byl zahájen Georgia Public Library System v roce 2006, který měl sloužit pro

zhruba 245 veřejných knihoven ve státě Georgia. Po jeho vydání ho začaly používat i knihovny v US, Kanadě a severní Americe. Ačkoli je systém docela mladý, tak se vývojářská komunita stále rozrůstá a proto lze do budoucna očekávat od tohoto systému stále více. (Georgia Public Library Service, © 2008-2014)

V České Republice začaly některé knihovny uvažovat nad zavedením Evergreen. Především kvůli pořizovací ceně. Komerční systémy, ale mají výhodu, protože Ministerstvo kultury nevidí v open source budoucnost. Proto byla implementace systému Evergreen zastavena. (Karpíšek, 2013)

Obrázek 7: Evergreen - katalog:



Na obrázku 7 je vidět www Evergreen katalog. Pod logem je hlavní nabídka. Vpravo nahoře se pod tlačítkem „My Account“ skrývá správa účtu. Ve středu obrazovky je samotné vyhledávání.

4. 6. Srovnání systémů Clavius a Tritius

Největšími rozdíly mezi těmito systémy je několik. Hlavní rozdíl je v náročnosti na hardware. Clavius je postaven na architektuře tlustého klienta. Pro využívání Clavia je tedy nutné, aby byl nainstalován na každém počítači tak, aby s ním mohl uživatel pracovat. Tritius je oproti tomu tenký klient. Tritius je konkrétně webová aplikace, takže se k němu připojíte z jakéhokoliv počítače, který má internetový prohlížeč.

Výhoda Tritia je i v tom, že všechny moduly a funkce Tritia se nacházejí v jednom okně (v prohlížeči). Clavius je rozdělen na několik částí (oken), které se spouští rozdílnými exe soubory. Další výhodou Tritia je podpora RFID čipů. Clavius

tuto funkci sice taky nabízí, ale pouze využitím externí aplikace. To znamená další okno na obrazovce navíc.

5. BIBLIOGRAFICKÉ FORMÁTY

5. 1. Historie

V první polovině 90. let byl UNIMARC na vrcholu sil. V porovnání se starým USMARC byl modernější, a to především díky větší granularitě. V této době měl UNIMARC podporu federace IFLA (Mezinárodní federace knihovnických asociací a institucí), tak i Evropské komise (pro získání finančních prostředků od Evropské komise byl podmínkou právě UNIMARC). Library of Congress proto vytvořila konverzní tabulky mezi USMARC a UNIMARC. Ani jedna instituce se nechtěla vzdát svého formátu, proto pokusy o jejich harmonizace a spojení do jednoho nebyla možná. Následovaly větší nároky, kterým se UNIMARC nebyl schopen přizpůsobit (např.: popis elektronických zdrojů). USMARC byl minimálně harmonizován a dostal název MARC21. Knihovny nebyly jednotné, protože některé nebyly ochotné vzdát se granularity a vazebních polí. Někde se využíval UNIMARC, jinde se přecházelo na MARC21. Tato rozepře nakonec vyústila v referendum. Referendum skončilo jasným vítězstvím pro MARC21. Velké země začaly přecházet na MARC21. Tím získal MARC21 větší podporu, jak personální tak finanční. To vedlo k nedostatku financí na podporu UNIMARCU. IFLA proto počítala do budoucna s finanční podporou od uživatelů. Tato vize se ale nenaplnila, a proto přestala UNIMARC podporovat. Dnes už i dodavatelé knihovních systémů více podporují MARC21. (Stoklasová, 2001; Národní knihovna České Republiky, © 2012)

Současnost v ČR:

„Od dubna 2015 budou platit nová katalogizační pravidla RDA, pro jejich aplikaci je nezbytné používat formát MARC21. Důvodem je, že pravidla RDA používají pole, která nejsou ve formátu UNIMARC k dispozici.“ (LANius, n. d.)

5. 2. Struktura záznamu v MARC21/UNIMARC

Formáty určují strukturu zápisu (kam se má údaj zapsat). Bibliografické formáty MARC21/UNIMARC mají následující strukturu. V monografii je jeden záznam, jedna kniha. První řádek záznamu je tzv. label neboli popisek. Následující řádky až do dalšího labelu jsou data k této knize. Každý řádek má tzv. tag. Label má tag ‚lbl‘ nebo ‚lab‘. Ostatní řádky jsou pole. Tato pole mají tag, který má vždy trojmístné číslo. Např.: 245,

100, 020. Následuje mezera, po které je indikátor. Indikátor se skládá se dvou čísel. Pokud indikátor v záznamu z nějakého důvodu chybí, nahradí se #. Tedy, pokud je záznam pouze 1, doplní se 1#. Po indikátoru následuje další mezera. Po mezeře začínají podpole. Každé podpole začíná oddělovačem \$. Po tomto oddělovači je označení pole např.: a, 4, b. Teprve za označením podpole se nacházejí údaje pro konkrétní podpole. Tedy, jedno pole se skládá z tagu, indikátoru a podpolí. Např.: 210 ## \$aPraha\$cAlbatros\$d2003. Zde je vidět kde, kdy a kým byla kniha vydána. Takže takovéto záznamy obsahuje katalog knihovny v závislosti na její velikosti. (Minty, 1998; Balíková, 2008)

5. 3. MARC21

Marc znamená Machine-Readable Cataloging. V češtině tedy strojem čitelná katalogizace. Je formátem pro bibliografickou komunikaci. Pomocí MARC21 byla umožněna elektronická výměna bibliografických formátů a sdílená katalogizace. Přístup k bibliografickým datům probíhal pomocí OPAC.

Kongresová knihovna v 50. letech zažádala o grant na vyvinutí automatizovaných technik. Následně na to vznikla skupina, která se tím měla zabývat. V čele této skupina byla Henriette Avramová. Byla to právě ona, kdo se z velké části podílel na vývoji Marc. Pilotního projektu se zúčastnilo 16 knihoven (Kabinet informačních studií a knihovnictví, 2011).

Nezkoumalo se pouze, jak by formát měl vypadat pro nejlepší uchování informací, ale zároveň, aby výstup z počítače byl dobře čitelný. Projekt započal v roce 1966 a pokračoval do roku 1971. Během něj se zkoumalo rozpoznávání formátu za účelem přiřazování indikátorů atd. (Ehrlichová, 2007)

Projekt přinesl výsledky, které ovlivnili MARCI. To vedlo k zdokonalení na MarcII (USMARC). (Kabinet informačních studií a knihovnictví, 2011).

Kongresová knihovna začala MARC používat v roce 1970. (Ehrlichová, 2007)

Marc II byl stále zdokonalován, až vznikla verze MARC21. MARC21 je kombinací formátů USMARC a CANMARC (Kanadský Marc). (Library of Congress, 2001)

Struktura MARC se stala základem pro mezinárodní normu ISO 2709. MARC21 je závislý na AACR2R a vyžaduje interpunkci. (Kabinet informačních studií a knihovnictví, 2011).

Vznik Marc měl velký dopad na rozvoj internetu. Lidé seděli doma a pouhým kliknutím mohli zjistit podrobnosti o požadované knize, pro kterou si následně mohli pouze dojít.

5. 4. UNIMARC

UNIMARC vytvořila IFLA (The International Federation of Library Associations and Institutions) v roce 1977. UNIMARC znamená univerzální MARC formát. Druhá verze UNIMARC byla vydána v roce 1980 a následně roku 1983 byl vydán UNIMARC Handbook. Až v roce 1987 byl teprve vydán manuál k UNIMARCU. Druhá verze manuálu byla vydána pod názvem UNIMARC Manual: bibliographic format v roce 1994. Následná vylepšení vycházela v letech: 1996, 1998, 2000, 2002 a 2005. Poslední a stávající verze vyšla roku 2008. (IFLA, 2014)

UNIMARC oproti MARC21 má větší granularitu. Také rozlišuje další autory od původců. (LANius, n. d.)

5. 5. Bibframe a budoucnost MARC21

V předchozích kapitolách byly popsány nejrozšířenější formáty UNIMARC a MARC21. V této kapitole se bude řešit nově vznikající formát Bibframe a jeho dopad na budoucnost dnes nejrozšířenějšího formátu MARC21.

Tady se objevují pojmy: linked data, RDF a sémantický web. Pro lepší pochopení jsou vysvětleny níže.

V počítačové terminologii se linked data vysvětlují jako metody zveřejňování popisující strukturu dat. Využívají se standartní webové technologie, jako jsou http, RDF a URI. Spíše než pro čtenáře a jejich čtení na webu jsou určeny pro počítače a jejich automatické zpracování. To umožňuje spojovat a zpracovávat data z různých zdrojů. Linked data je tedy popis doporučeného postupu sdílení informací na sémantickém webu prostřednictvím URIs a RDF. (Bizer, Heath a Berners-Lee, 2011, s.1-20)

Výhody linked data:

1. Umožňují lepší využití dat vytvářených knihovnami.
2. Tvoří nové propojení s webem.
3. Pokračují a zlepšují to, o co knihovnám jde – to je zpracování a zpřístupnění informací. (Janíček, 2013)

Problémy spojené s linked data:

1. menší kontrola dat
2. větší závislost na práci jiných
3. MARC21 bude muset být nahrazen. (Janíček, 2013)

Nejdůležitější „hráči“ se novému trendu již začali přizpůsobovat: Library of Congress, Deutsche Nationalbibliothek, LIBRIS, Harvard University, Medline, British Library atd. (Janíček, 2013)

Resource Description Framework (RDF) je standardní model pro výměnu dat na webu. RDF vypracovali W3C. Specifikace RDF se neustále rozšiřuje. (RDF Working Group, 2014)

Sémantický web je novým stupněm dosavadního webu, který staví na RDF. Informace jsou zde uchovávány podle standardizovaných pravidel. To vede k snadnějšímu vyhledávání a zpracování. Specifikace pro sémantický web vytváří W3C. Podnět k vytvoření sémantického webu dal roku 2001 Tim Berners-Lee. Tehdy řekl, že současný web je jen halda různých webových stránek, ve kterých je stále těžší se orientovat. Základní myšlenkou sémantického webu je standardizování popisu zdrojů. To znamená, že každý www zdroj bude popsán stejnými údaji (autor, typ zdroje atd.). Tím by byl umožněn přístup k síti www jako k relační databázi. Dotazování by pak mohlo probíhat podobnými příkazy jako je v jazyku SQL. (Matulík & Pitner, 2011)

Použití linked data má ale několik nedostatků. Prvním je přehodnocení dosavadních technologií. Zde se dostáváme k problému s MARC21. MARC21 v sobě obvykle ukládá věci, ne řetězce. Například v poli 100 je uložený autor jako Božena Němcová, ale ne odkaz na něj jako <http://nemcova.cz/>. Tím se stává MARC21 pro potřeby linked data špatně použitelným. Proto vznikl Bibframe. (Library of Congress, 2012)

Bibframe vyvíjí Library of Congress. Bibframe je linked data model pro knihovny. Bibframe je základem pro budoucnost bibliografického popisu, který se děje na webu v propojeném světě. Tím se z něj stává základní model pro definování webového přístupu k co největší efektivnosti při sdílení, orientaci a spolupráci.

Bibframe by měl přinést nové směry k:

1. rozlišení mezi koncepčním obsahem a fyzickými údaji
2. jednoznačnému identifikování informační entity (např.: autority)
3. odhalení a následnému využití vztahů mezi více entitami. (Library of Congress, 2012)

Je totiž důležité, aby ve webovém světě bylo možné uvést data knihovny způsobem, který odlišuje koncepční část od fyzických detailů. Takže oddělit např.: název a autora od počtu stránek a jestli má dílo ilustrace. Stejně důležité je, aby u dat knihovny byl jasně identifikován vznik (autor, vydavatel) a předmět ve spojení se zdrojem. (Library of Congress, 2012)

Další kroky při vývoji Bibframe jsou:

1. Rozšíření experimentace ve více institucích.
2. Vytvoření více detailní dokumentace.
3. Kompletní slovník mapování adres všech MARC a RDA polí.
4. Definování protokolů.
5. Spolupráce s vývojáři a členy komunity pro vývoj nástrojů a poskytování služeb. (Library of Congress, 2012)

Bibframe model je koncepční model, který vyvažuje požadavky těch, kteří chtějí podrobné informace a těch, kteří nevyžadují tolik detailní informace. (Library of Congress, 2012)

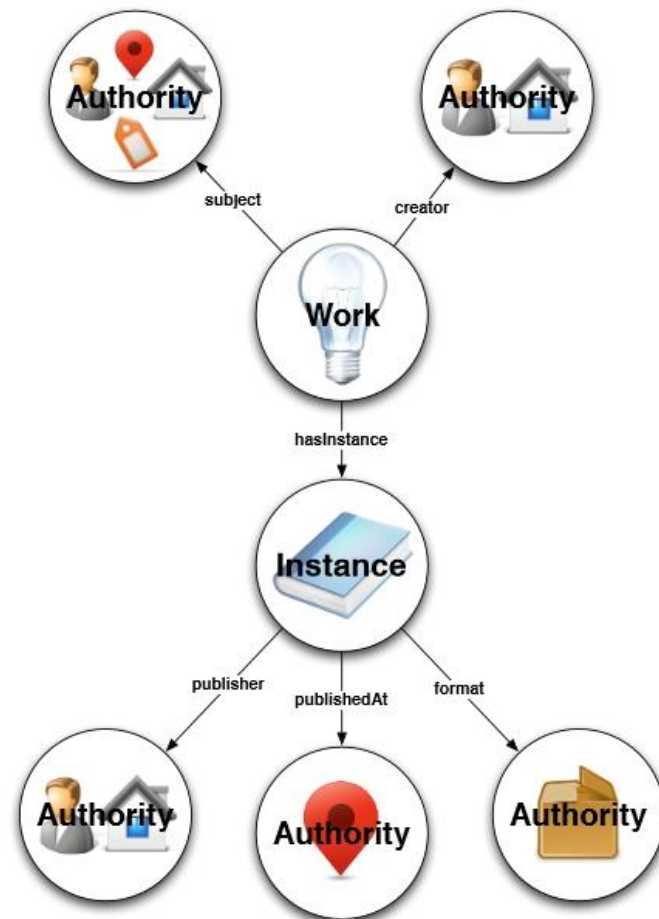
Bibframe model má čtyři základní entity:

1. Bibframe Work
2. Bibframe Instance
3. Bibframe Authority
4. Bibframe Annotation

Bibframe Work identifikuje podstatu něčeho (spojuje instance do celku), Bibframe Instance odráží materiální ztělesnění (konkrétní kus), Bibframe Authority

označuje věc spojenou s Bibframe Work a Bibframe Instance. Bibframe Annotation poskytuje nový způsob pro rozšíření Bibframe entit. (Library of Congress, 2012)

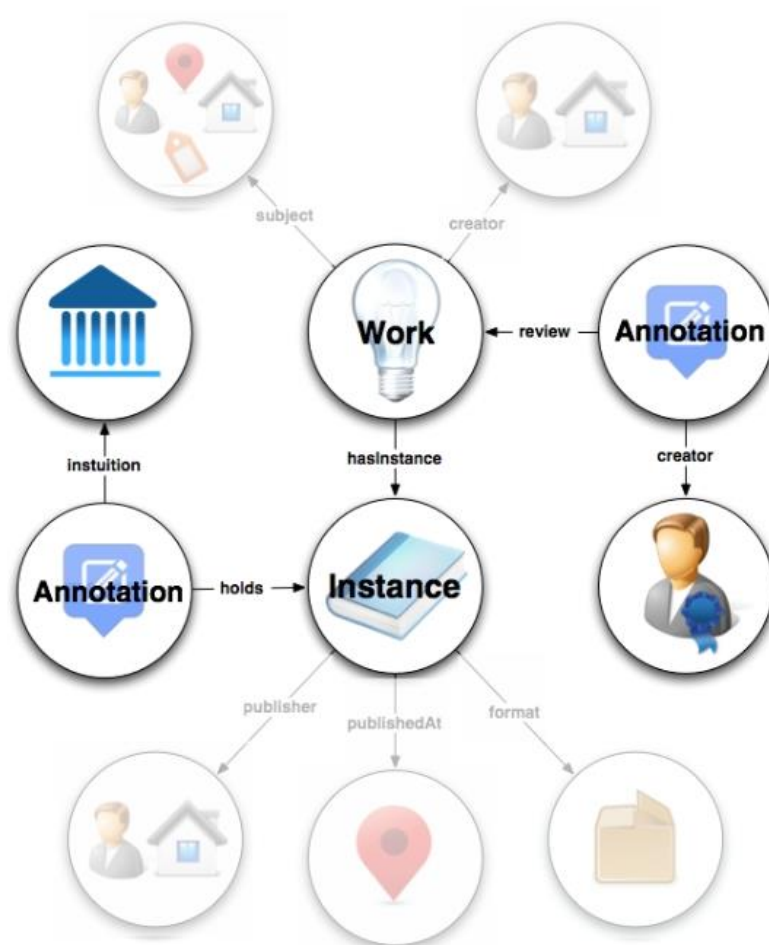
Obrázek 8: Bibframe model:



Zdroj: (Library of Congress, 2012)

Na tomto obrázku 8 jsou zobrazeny základní entity. Entita Work tedy ukrývá předmět dokumentu a autora dokumentu. Instance pak obsahuje vydavatele, místo vydání a formát konkrétního výtisku.

Obrázek 9: Bibframe model (Annotation): (Library of Congress, 2012)



Zdroj: (Library of Congress, 2012)

Na obrázku 9 lze vyčíst navíc anotace. Anotace k Work je například review (přehled) a anotace k instanci je hold (držitel).

Bibframe Vocabulary je klíč k popisu zdrojů. Stejně jako MARC má nadefinované atributy a elementy, tak Bibframe Vocabulary má třídy a vlastnosti. Třída určuje typ zdroje a vlastnost slouží jako popis zdroje. (Library of Congress, 2012)

Základní rozdíly mezi Bibframe a MARC21 jsou následující. Bibliografický formát MARC se zaměřuje na katalogové záznamy. MARC shromažďuje informace o díle, fyzickém nosiči a používá řetězce pro identifikátory, jako jsou osobní jména atd. Místo toho, aby se vše svazovalo do záznamu a tím duplikovaly informace, Bibframe spoléhá na vztahy mezi zdroji. To se daří pomocí identifikátorů pro věci (lidi, místa atd.). MARC některé tyto identifikátory už používá (geografické a jazykové kódy), ale

Bibframe se snaží, aby toto bylo standardem než jen výjimkou. (Library of Congress, 2012)

Bibframe je tedy následovníkem MARC21, který od samého začátku vývoje počítá s propojením pomocí zdrojů na rozdíl od svého předchůdce, který byl až následně upravován pro moderní potřeby. Bibframe se ještě nepoužívá. Je jen otázkou času než Bibframe dozraje, aby byl schopný nasazení. Už dnes jsou hotové nástroje pro převod z MARC21 do Bibframe. Opačný převod dnes ještě neexistuje. Jeho vývoj se plánuje až po dokončení Bibframe modelu a Bibframe vocabulary. (Library of Congress, 2012)

6. STANDARDY, METODICKÉ POSTUPY

6. 1. Norma ISO 2709

ISO 2709 je ISO standart pro bibliografické popisy. Byl vyvinut ve spojení s MARC týmem vedeným Henriette Avram v Library of Congress. Nejdříve se jmenoval Z39.2. Byl to první standart pro informační technologii. Poslední verze standartu byla vydána v prosinci roku 2008 pod označením ISO 2709:2008. (ISO, 2011)

ISO 2709 má tři části:

1. hlavička záznamu
2. adresář
3. data polí (ISO, 2011)

Hlavička záznamu je prvních 24 znaků. Je to jediná část záznamu, která má pevně danou délku. Obsahuje délku záznamu a základní adresu dat obsažených v záznamu. Adresář společně s tagy v polích poskytuje vstupní pozice do polí záznamu. Data polí jsou stringy obsahující data polí, tedy i podpolí. Z této normy vychází struktura záznamu v kapitole 5. 2. Struktura záznamu ve formátu MARC21/UNIMARC. (ISO, 2011)

6. 2. International Standart Bibliographic Description (ISBD)

Mezinárodní standardní bibliografický popis (ISBD) je sada pravidel, kterou vytvořila IFLA. Cílem těchto pravidel je katalogizace knihovnických materiálů. ISBD je spravováno federací IFLA. Původním účelem bylo poskytnout standardizaci pro mezinárodní výměnu záznamů. (Gabriela Obstová, 2010)

ISBD dělí bibliografický záznam na osm oblastí s údaji. Tedy určuje, jaké údaje budou v určité oblasti při určité interpunkci obsaženy. Díky tomu lze porozumět jakémukoliv bibliografickému záznamu kdekoliv ve světě. Z ISBD vychází formát UNIMARC nebo pravidla AACR2. Jeho vývoj započal v roce 1969. Konečné revize se

ISBD dočkalo v roce 2009. Ustálená verze dnes slučuje dohromady sedm specializovaných ISBD (jako jsou mapy, monografie atd.). (Gabriela Obstová, 2010)

Popisné údaje se dělí do 8 oblastí:

1. oblast údajů o názvu a odpovědnosti
2. oblast údajů o vydání
3. oblast specifických údajů (druh publikace)
4. oblast nakladatelských údajů
5. oblast údajů fyzického popisu
6. oblast údajů o edici
7. oblast údajů poznámky
8. oblast údajů o standardním (nebo alternativním) čísle a dostupnosti (Obstová, 2010)

6. 3. Angloamerická katalogizační pravidla (AACR2R)

Anglo-American Cataloguing Rules (AACR) jsou angloamerická katalogizační pravidla, která byla poprvé zveřejněna v roce 1967 (American Library Association, Canadian Library Association, the Chartered Institute of Library a Information Professionals, 2006a).

AACR2 je druhé vydání těchto pravidel, která vznikla roku 1978 (Huthwaite, Bibliographic Services Manager, Queensland University of Technology Library, Grove Campus, Park Rd. a Grove Qld, 2001), které již bylo přizpůsobeno ISBD. Tím byla sjednocena dvě pravidla do jednoho.

AACR společně vydali American Library Association, Canadian Library Association a Chartered Institute of Library and Information Professionals. Editorem je knihovník Michael Gorman. Online verze je dostupná z Library of Congress. Poslední verze vyšla v roce 2005. (American Library Association, Canadian Library Association, the Chartered Institute of Library a Information Professionals, 2006b)

7. PŘEVODNÍK

7. 1. Úvod

Praktická část je zaměřena na vstupní formáty pro katalogizační systémy Clavius a Tritius. Clavius je sice starším softwarem od firmy Lanius s.r.o., ale přesto má výhodu podpory formátu UNIMARC. U nového Tritia se už počítalo s rozšířenějším formátem MARC21. Tritius není na UNIMARC momentálně nastavený, ale to neznamená, že by to v budoucnu nebylo možné.

Právě převod bude řešit tento převodník, který byl pro převod vytvořen. Pozdější převod zpět bude také možný, protože převodník bude převádět i z MARC21 na UNIMARC.

7. 2. Analýza

Před zahájením tvorby převodníku se muselo zodpovědět několik hlavním otázek. Bylo třeba zvolit jazyk, v kterém se bude převodník programovat. Tato volba byla v celku jednoduchá a to z důvodu toho, že Tritius je vyvíjen v Javě. Kvůli budoucí kompatibilitě je Java nejlepším řešením. Více o jazyce Java je napsáno v kapitole 7. 2. 1. jazyk JAVA.

Vstup bude načítán z textového souboru formátu txt. K převodu bude třeba také načíst kodtom tabulku, která je formátu tag. Výstup bude textový soubor formátu txt.

7. 2. 1. Jazyk Java

Jazyk Java vznikl v roce 1995. Původně to byl malý jazyk, který obsahoval pouze 211 tříd. Po svém vydání sklidila Java veliký úspěch, protože v ní začalo programovat zhruba 3 milióny vývojářů. Tento ohlas motivoval vývojáře, aby pokračovali. Java je dnes nejrozšířenějším programovacím jazykem. Není proto divu, že se Java používá na většině univerzit pro výuku základů programování.

Hlavních důvodů, díky kterým je Java tolik populární, je několik. Prvním je objektově orientovaný jazyk. Na rozdíl od jiných jazyků Java neumožňuje psaní jinak než objektově. Druhým důvodem je jednoduchost. Javu se programátor dokáže naučit za několik hodin. Jednoduchost, ale neznamená omezenost. V Javě se vytvářejí nejen malé, ale i velké projekty. Java je multiplatformní. To znamená, že běží prakticky na

jakémkoliv počítači. To je možné pomocí takzvaného Java Virtual Machine. Program se nejprve rozloží na „bajtkód“, který Java Virtual Machine interpretuje pro specifický hardware. Pro každý hardware lze definovat vlastní Java Virtual Machine. Proto Java funguje na operačních systémech Windows, Linus, MacOS a dalších. Java není jen jazyk, ale zároveň i platforma. Java platformy jsou čtyři: J2SE, J2EE, J2ME a Java CARD. Java CARD je pro aplikace chytrých karet. Java ME je pro mobilní aplikace. Java SE pro aplikace na stolní počítače a Java EE pro rozsáhlé podnikové systémy. (Pecinovský, 2005; Pecinovský, 2004)

7. 3. Návrh

Dalším krokem byl návrh struktury převodníku. Struktura tříd je zobrazena v class diagramu v kapitole 7. 3. 1. UML class diagram. Plánovaný algoritmus převodníku je nastíněn v kapitole 7. 3. 2. Vývojový diagram.

7. 3. 1. UML class diagram

Převodník se skládá z pěti tříd. Hlavní třídou je AppGUI, která spouští samotný program. Obsahuje tyto atributy:

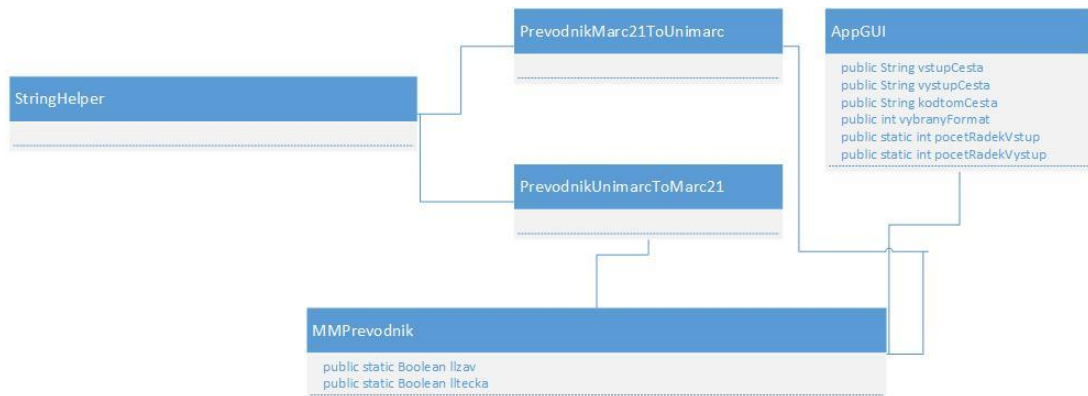
```
public String vstupCesta = ""; //sem se uklada cesta k vstupnímu souboru
public String vystupCesta = ""; // sem se uklada cesta kam se uloží výstupní soubor
public String kodtomCesta = ""; //sem se ukládá cesta ke kodtom tabulce
public int vybranyFormat = 0; //pro uložení rozhodnutí z kterého formátu převádím
public static int pocetRadekVstup = 0; //počet vstupních řádek
public static int pocetRadekVystup = 0; //počet výstupních řádek
```

PrevodMarc21ToUnimarc a PrevodUnimarcToMarc21 jsou hlavní třídy pro převod. Tyto třídy neobsahují atributy. V převodníku se zavolá třída podle toho, který formát hodlám převést. MMPrevodnik obsahuje metody společné pro oba převody.

StringHelper obsahuje metody pro práci se Stringy a nejsou zde atributy.

Na obrázku 10 je ukázán zjednodušený UML class diagram. Kompletní je v příloze pod číslem 1.

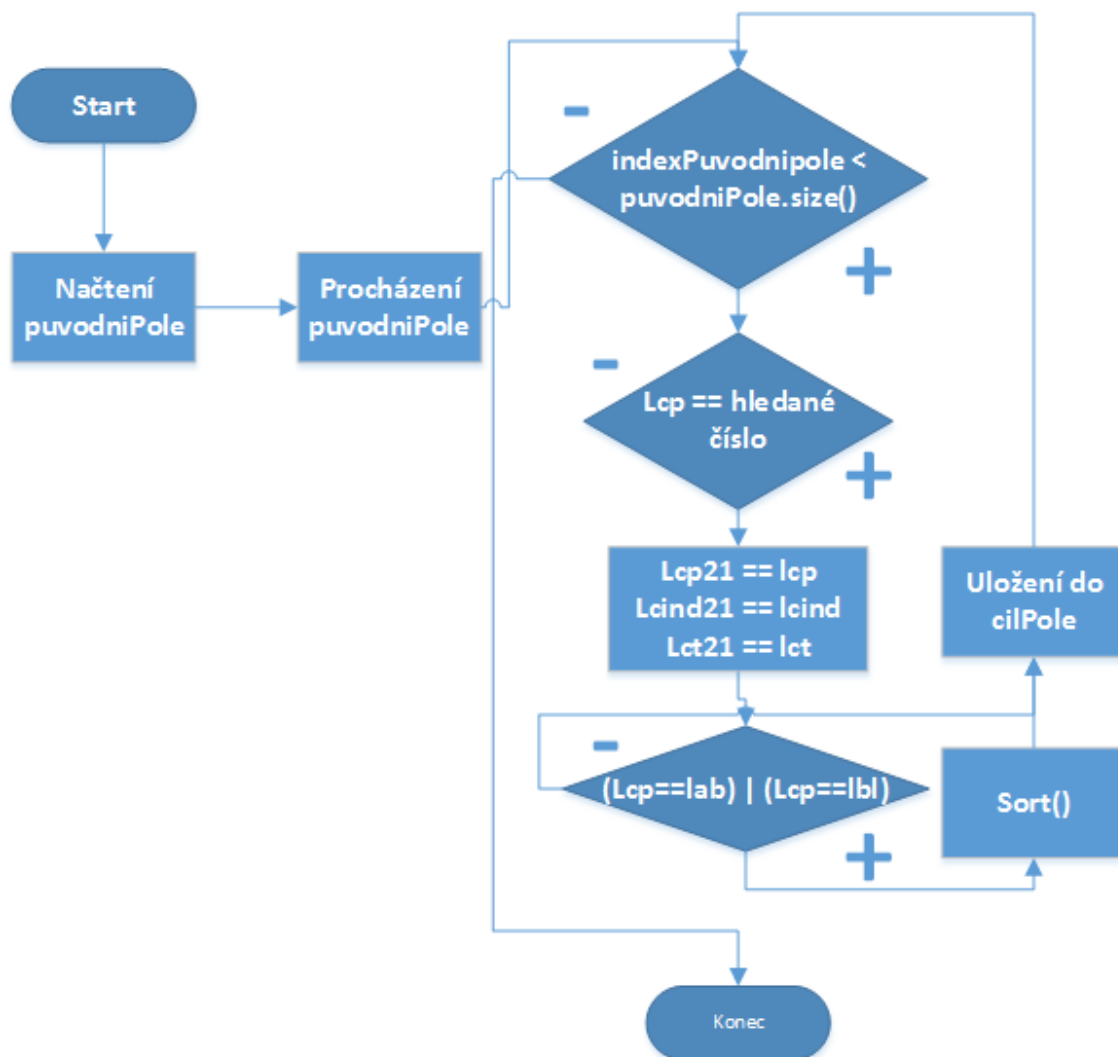
Obrázek 10: UML class diagram převodníku:



7. 3. 2. Vývojový diagram

Zde je co nejjednodušeji nastíněn průběh programu. Aplikace nejdříve načte jednotlivé záznamy do pole puvodniPole, pak ho po jednom prochází, dokud $indexPuvodniPole < puvodniPole.size()$. Když se proměnná lcp rovná konkrétní podmínce hledaného pole, upraví se tři hlavní proměnné a uloží do nových (lcp21, lcind21 a lct21). Pokud se proměnná lcp rovná lab nebo lbl, seřadí se celý záznam a uloží do pole cilPole. V opačném případě se pouze uloží. Po uložení běží cyklus znovu, dokud platí jeho podmínka. Když podmínka cyklu přestane platit, znamená to, že byly převedeny všechny záznamy a program skončí.

Obrázek 11: Vývojový diagram:



7. 4. Implementace

Pro přehlednost jsou zde popsány pouze nejdůležitější části kódu. Celé kódy metod kodToM21() a sort() jsou uloženy v přílohách. Kompletní převodník je uložený jako příloha 8, která se nachází na přiloženém CD.

7. 4. 1. Metoda kodToM21()

V této části bude osvětleno, proč aplikace před spuštěním vyžaduje tabulku kodtom. Některá podpole obsahují kód země původu. Není překvapením, že v UNIMARCU a MARC21 se tyto kódy liší. Například UNIMARC obsahuje aut, ale v MARC21 to je 070. Tato metoda vyměňuje tyto kódy právě podle tabulky kodtom. Samozřejmě by tato tabulka mohla být nahraná v samotné aplikaci a uživatel by to ušetřilo práci s obsluhou převodníku. Je to z důvodu změn, které by mohly v tabulce

nastat. Proto je tabulka umístěna mimo kód převodníku. Tabulka obsahuje tři sloupce. První sloupec obsahuje číslo, podle kterého se určí, pro která pole jsou další dva sloupce určeny. Tím se tabulka dělí na dvě poloviny. První půlka obsahuje hodnotu 7XX, druhá 102. První tedy znamená, že je určena pro různá pole začínající na 7. Druhá polovina se používá pouze pro pole 102. Hledaný řádek se určí podle původní hodnoty. Poté se původní hodnota nahradí za novou, která je uložena v nalezeném řádku. Tedy řádek v tabulce vypadá takto: 7XX, 070, aut. UNIMARC podpole obsahuje aut, takže se původní hodnota nahradí za 070.

Metoda je celá k nahlédnutí v příloze 3 metoda kodToM21().

7. 4. 2. Metoda sort()

Některá pole po převedení už nebudou správně seřazena z důvodu změny čísla, a proto se musí záznam seřadit. Klasické dostupné metody pro řazení, které Java obsahuje, nelze použít, protože krom seřazení jedné číslice, s nimi je potřeba seřadit i dva další Stringy. Je zde proto popsáno, jak funguje metoda na seřazení. Cilpole se s každým záznamem zvětšuje, proto je důležité při převádění každého záznamu, pracovat pouze se záznamem, který je potřeba seřadit. Kdyby bylo záznamů např.: 20 a metoda by měla seřadit jen 18. záznam, Cilpole by se řadilo od 1. do 18., ale bylo by to zbytečné, protože předchozí záznamy už byly seřazeny. Proto má metoda sort() tři parametry. První je Cilpole, kde jsou uloženy již převedené záznamy. Druhý je startRazeni, který má hodnotu čísla prvního řádku v právě převáděném záznamu. Poslední parametr index je číslo, které mi udává poslední řádek v záznamu.

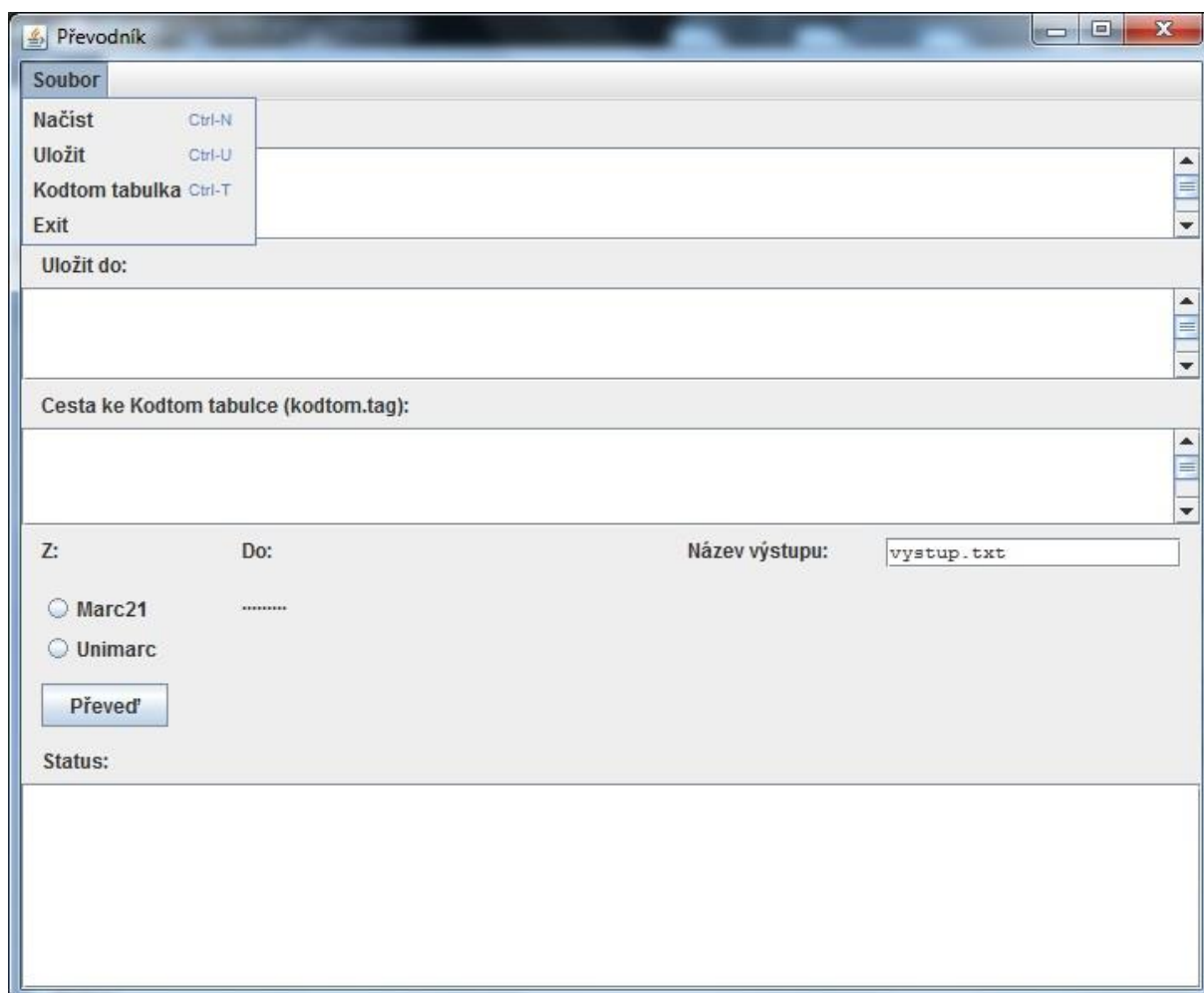
Metoda sort() nejdříve zkontroluje, jestli záznam neobsahuje hodnotu null, která by mohla později způsobovat chyby. To se provede tak, že se Cilpole projde for cyklem a pokud zde bude null, nahradí se hodnotou "" (prázdný string).

Samotné řazení probíhá tak, že jsou vnořené dva cykly do sebe. První cyklus vypadá takto: for (int i = startRazeni; i < index; i++) a druhý: for (int j = startRazeni; j < index; j++). Tím se docílí toho, že každý řádek v záznamu se porovná s každým. Pokud řádek začíná L nebo l, uloží se rovnou na první místo, protože je hlavičkou záznamu. Pokud tedy momentální číslo řádku je větší než to, s kterým se právě porovnává, uloží se na současnou pozici. Pokud není, prohodí se s tím, které je větší. Takto se řadí stále dokola, dokud se neseřadí všechny řádky. Metoda sort() je zobrazena v příloze 2.

7. 4. 3. Uživatelské rozhraní převodníku

Interface hotového převodníku vypadá následovně. Nebylo zapotřebí ho nějak komplikovat. Nahoře se nachází menu s názvem „Soubor“. Tato nabídka nabízí čtyři možnosti. První je „Načíst“. Pokud uživatel nechce psát cestu k souboru, který bude převádět, ručně vybere soubor přes procházení. Druhá možnost je „Uložit“. Tato volba provede stejné, ale s tím rozdílem, že místo souboru vybere složku, kam se nový soubor uloží s výsledným převodem. V „Kodtom tabulce“ se vybere soubor, ve kterém je uložena tabulka obsahující kódy důležité pro převod. Poslední možnost „Exit“ okamžitě ukončí aplikaci převodník. V horní polovině programu se nachází text boxy, kam lze ručně zadat cestu k souborům. Zde se poté zobrazí cesta k souborům, pokud se použijí možnosti z nabídky v menu „Soubor“. Pod text boxy pro cesty k souborům, se nalevo nachází nabídka formátů. Pokud se zvolí například Marc21, tak se vedle pod nápisem „Do:“ doplní Unimarc. Když se vybere Unimarc, doplní se Marc21. Vpravo od této nabídky se nachází text box, obsahující název výsledného souboru. Ten se mění v závislosti na zvoleném formátu. Pokud není zvolen žádný formát, je název výstup.txt. Po zvolení např.: Marc21 se obsah text boxu změní na unimarcVystup.txt. Tato funkce funguje i obráceně. Tedy při výběru Unimarc se doplní marc21Vystup.txt. Název souboru lze libovolně přepsat, ale musí končit na příponu txt. Pod možností výběru formátu se nachází tlačítko „Převod“. Toto tlačítko zahájí převod. Aplikace má ošetření proti následujícím možnostem: nevyplnění cest ke všem souborům, nezvolení formátu nebo název výstupního souboru není ve správném formátu. Po aktivaci tlačítka začne převod. Po dokončení převodu se shrnuté výsledky testu objeví v text boxu, který je nadepsaný „Status“.

Obrázek 12: Uživatelské rozhraní převodníku:



7. 4. 4. Algoritmus převodníku

Z čeho jsou složeny jednotlivé záznamy monografie, bylo napsáno v kapitole 5. 2. Struktura záznamu v MARC21/UNIMARC.

Po zmáčknutí tlačítka „Převéd“ se hodnoty v proměnných `vystupNazevPole`, `vstupCesta`, `vytupCesta`, `kodtomCesta` předají dál převodu a spustí se samotný převod. Buď ze třídy `PrevodMarc21ToUnimarc` nebo z `PrevodUnimarcToMarc21`.

Zdrojový kód 1: metoda `Převéd()`:

```
if ((!vstupCesta.equals("")) & (!vystupCesta.equals("")) &
(!kodtomOkno.getText().toString().equals(""))) {
    String path = AppGUI.class.getProtectionDomain().getCodeSource().getLocation().getPath();
    String decodedPath = "";
    statusText.setText("Převádím....." + '\n');
```



```

statusText.append("-----" + "\n");

statusText.update(statusText.getGraphics());

try {

    decodedPath = URLDecoder.decode(path, "UTF-8");}catch(Exception vyjimka) {

    statusText.setText("Nenalezen Soubor kodtom21" + vyjimka);}

StringBuilder prevedeno = new StringBuilder();

if(vystupNazevPole.getText().indexOf(".") == -1){

    statusText.append("Převod přerušen -> Výstupní soubor nemá příponu!" + "\n");

    else{

        if(vybranyFormat==1) {

            PrevodMarc21ToUnimarc.prevod(vystupNazevPole.getText() ,vstupCesta, vystupCesta ,
kodtomCesta); }

            if(vybranyFormat==2) {PrevodUnimarcToMarc21.prevod(vystupNazevPole.getText(),
vstupCesta, vystupCesta, kodtomCesta); }

            statusText.append("Převod dokončen" + "\n");

            statusText.append(pocetRadekVstup + " řádků bylo převedeno na " + pocetRadekVystup +
".");}

        } else {

            statusText.append("Nejdříve zadejte cesty k souborům!" + "\n");}

```

Převodník po řádcích načte záznamy. Po řádcích je poté prochází pomocí for cyklu, kde každý řádek rozloží do tří proměnných. První je StringBuilder lcp, který obsahuje číslo pole. Indikátory obsahuje String lcind a podpole jsou uloženy v StringBuilder lct.

Zdrojový kód 2: metoda nactiSoubor():

```

/**

* Nacte obsah souboru do StringBuilderu.

* @param cestaKeZdroji Cesta ke zdrojovemu souboru - např. C:\\test.txt

* @return Vraci obsah souboru ve StringBuilderu.

*/

```

```

public static List<StringBuilder[]> nactiSoubor(String cestaKeZdroji){
    List<StringBuilder[]> sr = new ArrayList<StringBuilder[]>();
    try {
        Scanner scanner = new Scanner(new File(cestaKeZdroji));
        while (scanner.hasNextLine()) {
            String zdroj = scanner.nextLine();
            StringBuilder[] sb = new StringBuilder[2];
            sb[0] = new StringBuilder(zdroj.substring(0, 3));
            sb[1] = new StringBuilder(zdroj.substring(3));
            sr.add(sb);}
        scanner.close();
    } catch (FileNotFoundException ex) {
        Logger.getLogger(MMPrevodnik.class.getName()).log(Level.SEVERE, null, ex); }
    return sr;
}

```

Když pole začíná na hledané číslo, neboli lcp, tedy obsahuje hledané číslo pole, vykoná se dotyčná podmínka if(), která obsahuje příkazy potřebné pro upravení všech tří proměnných pole do jiného formátu.

Zdrojový kód 3: příklad – if podmínky:

```

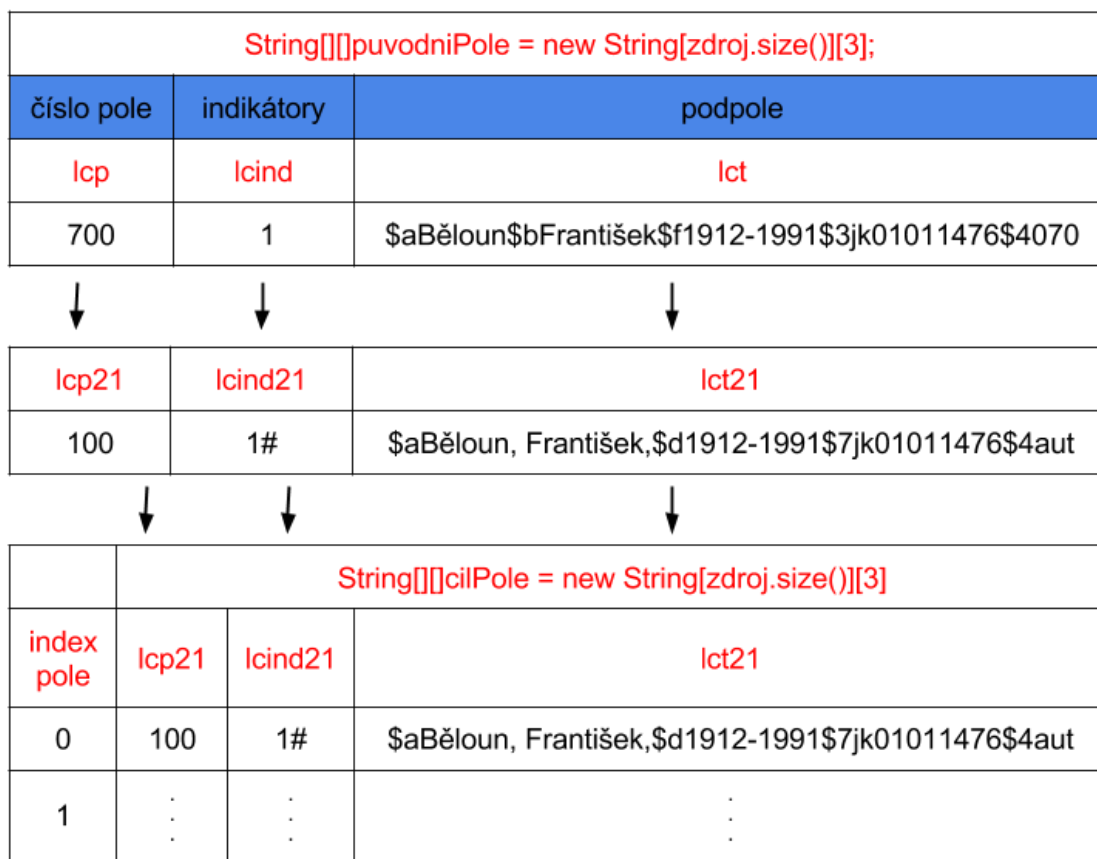
if (lcp.toString().equals("012")) { //Fingerprint
    lcp21.append("026");
    lct = StringHelper.javaSTRTRAN(lct, inzn1 + "a", " " + inzn1 + "e");
    lct = StringHelper.javaSTRTRAN(lct, inzn1 + "b", " " + inzn1 + "2");
    bylaspravne = true;}

```

Po upravení a dojetí cyklu na konec se proměnné uloží do dvourozměrného pole. První rozměr jsou řádky a druhý rozměr jsou sloupce.

For cyklus tedy posouvá řádky, kde řádky jsou pole a sloupce jsou proměnné. Pro lepší pochopení je průběh jednoho cyklu vysvětlen na obrázku.

Obrázek 13: Převedení názvu díla z UNIMARC do MARC21:



Na obrázku 13 je vidět, že pokud má proměnná lcp hodnotu 700, převede se na 100 a uloží do lcp21. Indikátory se změní na 1#, podpole z proměnné lct se upraví a uloží do lct21. Převedené proměnné se ukládají do pole Cilpole. Cilpole má tolik řádků, jako původní záznam. Sloupce má tři, protože ukládáme 3 proměnné. Takže pole 700 1 \$aBělouň\$bFrantišek\$f1912-1991\$3jk01011476\$4070 se uloží do prvního řádku (programovací jazyk Java, ale indexuje od 0, proto se proměnné ukládají na řádek s indexem 0). Je tedy vidět, že lcp21 je v prvním sloupci, lcind21 ve druhém a lct21 ve třetím.

Po načtení se pole nechá seřadit funkcí sort () podle čísla polí. Po seřazení následuje vytvoření souboru pomocí metody vytvořSoubor().

Zdrojový kód 4: metoda vytvořSoubor():

```
/**
 * Metoda vytvoří soubor
 * @param nazev - řetězec s názvem vytvářeného souboru
```

```

* @param cestaKVystupu - řetězec, kde je uložena cesta kde se má vytvořit soubor
*/

public static void vytvorSoubor(String nazev, String cestaKVystupu){

    File dir = new File(cestaKVystupu);

    try{

        dir = dir.getCanonicalFile();

    } catch (IOException e){
Logger.getLogger(MMPrevodnik.class.getName()).log(Level.SEVERE,
"Došlo k chybě při vytváření souboru", e);}

        File f = new File(dir, nazev);

        try {

            if (!f.createNewFile()) {}

        } catch (IOException e) {

Logger.getLogger(MMPrevodnik.class.getName()).log(Level. SEVERE,
"Došlo k chybě při vytváření souboru", e);}
}
}

```

Do souboru se po řádku uloží obsah pole Cilpole pomocí metody zapsani(). Pole Cilpole má délku původního záznamu krát 2, protože při převodu z MARC21 do UNIMARC se některá pole dělí na více polí.

Zdrojový kód 5: metoda zapsani():

```

/**

* Metoda která zapíše do souboru

* @param cilPole - pole z kterého se bude zapisovat

* @param nazevSouboru - název souboru kam se bude zapisovat

* @param cestaKVystupu - řetězec s cestou k souboru kde bude uložen výstup
*/

public static void zapsani(String[][] cilPole, String nazevSouboru, String cestaKVystupu) throws
IOException{

    System.out.println("Zapisuji....");

    String soubor = nazevSouboru;
}

```

```

PrintWriter vystupuj = new PrintWriter(new FileWriter(cestaKVystupu + "\\\" + soubor));

for(int c = 0; c < cilPole.length; c++ ){

    if(((cilPole[c][0] != null) & (cilPole[c][1] != null) & (cilPole[c][2] != null))){

        if(((!cilPole[c][0].trim().equals("")) | (!cilPole[c][1].trim().equals("")) |
(!cilPole[c][2].trim().equals("))))){

            if(cilPole[c][1].toString().equals(" ")){

                cilPole[c][1] = " ";}

            if(StringHelper.inlist(sB(cilPole[c][0].trim()), "LAB", "001", "005", "LBL", "lab", "lbl", "",
"", "", "")){

                vystupuj.println(cilPole[c][0].trim() + " " + cilPole[c][1] + " " + cilPole[c][2].trim());} }

        vystupuj.close();}

```

7.5. Test

Test byl nejdříve proveden na krátkém souboru, který měl pouze 4 záznamy. Tento test dopadl uspokojivě, a proto se přešlo k dalšímu kroku testu. Finální test byl soubor o velikosti 64 MB, který obsahoval 2500 000 řádků. Tento test dopadl neúspěšně. Nejdříve bylo nutné opravit chyby při převodu, aby se podařilo soubor převést celý. Po opravení chyb převod trval 8 hodin. Tento problém bylo třeba opravit změnou algoritmu. Test byl nakonec proveden za jednu hodinu. Konečný výsledek byl shledán uspokojivým.

8. ZÁVĚR

Cílem práce bylo popsat knihovní systémy Clavius, Tritius a rozebrat problematiku bibliografických formátů. Knihovní systémy zde byly rozebrány a pro porovnání byly uvedeny i jiné. K bibliografickým formátům musely být rozebrány i standarty, které je upravují, a to z důvodu pochopení dané problematiky.

V praktické části měl vzniknout převodník pro převod formátů MARC21 a UNIMARC. Převodník byl nejprve vyzkoušen na záznamu o cca 2500 000 řádcích. Za předpokladu, že průměrná knížka obsahuje 8 řádků to tedy je 312 500 knih. Po několika opravách a vylepšeních tento test prošel. Chyby byly především kvůli neočekávaným znakům ve stringu, které musely být escapovány.

Původní převod trval neuvěřitelných 8 hodin, což je velice špatné. Po úpravě části algoritmu, především pak ukládání dat do pole před převodem, zde docházelo ke zbytečnému hromadění dat a tím narůstajícímu nedostatku paměti počítače. To vedlo k narůstajícímu zpomalení počítače s každým dalším záznamem. Data byla při načtení uložena do jednoho pole. Následně se ukládala při každém záznamu do puvodniPole. Po převedení, se výsledná převedená data uložila do cilPole. Takže zde vznikalo jedno pole navíc, které zabíralo zbytečně paměť počítače. Na krátkém záznamu nebylo nic znát. Až tento rozsáhlý test objevil tento veliký nedostatek.

Po vyřešení tohoto nedostatku se délka převodu zkrátila na jednu hodinu. Následně byl převodník vyzkoušen v ostrém provozu firmou Lanius s. r. o., kde se převodník ukázal provozu schopný a byl nasazen do provozu. Hotový převodník má zhruba 5500 řádků kódu a výslednou velikost 127 kb.

Všechny předem stanovené cíle byly splněny.

Summary and keywords

This work tries to describe information systems Clavius and Tritius. Also, there trying to delineate the issue bibliographic MARC21, UNIMARC and Bibframe. With this knowledge was created an application for the transfer of records between Clavius and Tritius. Specifically, between MARC21 and UNIMARC formats.

Keywords: Clavius, Tritius, MARC21, UNIMARC, BIBFRAME, Java

Seznam literatury

1. AMERICAN LIBRARY ASSOCIATION, CANADIAN LIBRARY ASSOCIATION, AND THE CHARTERED INSTITUTE OF LIBRARY AND INFORMATION PROFESSIONALS. (2006). *Aacr2*. Dostupné z WWW: <<http://www.aacr2.org/about.html>>.
2. AMERICAN LIBRARY ASSOCIATION, CANADIAN LIBRARY ASSOCIATION, AND THE CHARTERED INSTITUTE OF LIBRARY AND INFORMATION PROFESSIONALS. (2006). *Aacr2*. Dostupné z WWW: <http://www.aacr2.org/canada/products_aacr2.html>.
3. BALÍKOVÁ, M. (2008). *Katalogizace ve formátu MARC21: stručná instrukce a příklady pro knihy a některé typy pokračujících zdrojů*. (2th ed.). Praha: Národní knihovna ČR.
4. BARTOŠEK, M. (© 2011). Aleph: nový knihovní systém pro MU. *Zpravodaj ÚVT MU*. 13(2), 6-10. Dostupné z WWW: <<http://www.ics.muni.cz/bulletin/articles/263.html>>.
5. BIZER, CH., HEATH, T., & BERNERS-LEE, T. (2009). *Linked Data – The Story So Far*. pp. 26. Dostupné z WWW: <<http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf>>.
6. EFFECTIVA SOLUTIONS. (© 2012). *Funkce – Tritius*. Dostupné z WWW: <<http://www.tritius.cz/funkce>>.
7. EFFECTIVA SOLUTIONS. (© 2012). *Moduly – Tritius*. Dostupné z WWW: <<http://www.tritius.cz/moduly>>.
8. EFFECTIVA SOLUTIONS. (© 2012). *Tritius*. Dostupné z WWW: <<http://www.tritius.cz/>>.
9. EHRLICHOVÁ, K. (2007). Henriette D. Avramová: její osobní život a kariéra. *IKAROS*. 11(4). Dostupné z WWW: <<http://www.ikaros.cz/node/4041>>.
10. EX LIBRIS. (n. d.). *Integrated Library System*. pp. 6. Dostupné z WWW: <http://www.exlibrisgroup.com/files/Solutions/Aleph_A4_low.pdf>.
11. GABRIELA, O. (2010). *ISBD*. Dostupné z WWW: <<http://kisk.phil.muni.cz/wiki/ISBD>>.

12. GEORGIA PUBLIC LIBRARY SERVICE. (© 2008-2014). *About Us / Evergreen ILS*. Dostupné z WWW: <http://www.aacr2.org/canada/products_aacr2.html>.
13. HUTHWAITE, A., BIBLIOGRAPHIC SERVICES MANAGER, QUEENSLAND UNIVERSITY OF TECHNOLOGY LIBRARY, GROVE CAMPUS, K., PARK, V., & GROVE QLD, K. (2001). *AACR2 and Its Place in the Digital World: Near-Term Solutions and Long-Term Direction*. Dostupné z WWW: <http://www.loc.gov/catdir/bibcontrol/huthwaite_paper.html>.
14. INTERNATIONAL FEDERATION OF LIBRARY ASSOCIATIONS AND INSTITUTIONS. (2014). *UNIMARC formats and related documentation*. Dostupné z WWW: <<http://www.ifla.org/publications/unimarc-formats-and-related-documentation>>.
15. ISO. (2011). *ISO 2709:2008 Information and documentation – Format for information exchange*. Dostupné z WWW: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41319>.
16. JANÍČEK, M. (2013). *Linked data (nejen) v knihovnách*. pp. 36. Dostupné z WWW: <http://invenio.nusl.cz/record/156682/files/idr-664_1.pdf>.
17. KABINET INFORMAČNÍCH STUDIÍ A KNIHOVNICTVÍ. (2011). *Henriette D. Avram*. Dostupné z WWW: <http://kisk.phil.muni.cz/wiki/Henriette_D._Avram>.
18. KARPÍŠEK, H. (2013). *AKS Evergreen „v praxi“ turnovské knihovny*. Dostupné z WWW: <<http://www.slideshare.net/cernin/aks-evergreen-v-praxi-turnovsk-knihovny>>.
19. LANIUS. (n. d.). *Clavius – základní informace*. Dostupné z WWW: <<http://www.clavius.cz/clavius/info.htm>>.
20. LANIUS. (n. d.). *Knihovní systém Clavius – přechod na interní formát MARC21*. Dostupné z WWW: <<http://www.lanius.cz/clavius/marc21.htm>>.
21. LIBRARY OF CONGRESS. (2001). *MARC21: Harmonized USMARC and CAN/MARC*. Dostupné z WWW: <<http://www.loc.gov/marc/annmarc21.html>>.
22. LIBRARY OF CONGRESS. (2012). *Bibliographic Framework as a Web of Data: Linked Data Model and Supporting Services*. pp. 42. Dostupné z WWW: <<http://www.loc.gov/bibframe/pdf/marclid-report-11-21-2012.pdf>>.

23. LIBRARY OF CONGRESS. (n. d.). *BIBFRAME Frequently Asked Questions*. Dostupné z WWW: <<http://www.loc.gov/bibframe/faqs/#q01>>.
24. MATULÍK, P., & PITNER, T. (2011). Sémantický web a jeho technologie. *Zpravodaj ÚVT MU*. 14(3), 15-17. Dostupné z WWW: <<http://www.ics.muni.cz/bulletin/articles/296.html>>.
25. MINTY, W. (1998). *AACR2 a UNIMARC: příručka pro praktickou katalogizaci s příklady*. (1th ed.). Praha: Národní knihovna ČR.
26. NÁRODNÍ KNIHOVNA ČESKÉ REPUBLIKY. (© 2012). *Zhodnocení vývoje formátu UNIMARC*. Dostupné z WWW: <<http://text.nkp.cz/o-knihovne/odborne-cinnosti/zpracovani-fondu/informativni-materialy/doporady>>.
27. PECINOVSKÝ, R. (2004). *Myslíme objektově v jazyku Java 5.0*. Praha: Grada Publishing.
28. PECINOVSKÝ, R. (2005). *Java 5.0 : novinky jazyka a upgrade aplikací*. Praha: CP Books.
29. RDF WORKING GROUP. (2014). *Resource Description Framework (RDF)*. Dostupné z WWW: <<http://www.w3.org/RDF/>>.
30. STOKLASOVÁ, B. (2001). Přežije formát UNIMARC rok 2003?. *IKAROS*. 5(9). Dostupné z WWW: <<http://ikaros.cz/prezije-format-unimarc-rok-2003>>.

Seznam obrázků a zdrojových kódů

Obrázek 1: Clavius (katalog a vyhledávání):	13
Obrázek 2: www katalog - Clavius:	14
Obrázek 3: výpůjční protokol:.....	15
Obrázek 4: Tritius - katalogizace:	19
Obrázek 5: Katalog - Carmen:	20
Obrázek 6: Aleph - katalog:	21
Obrázek 7: Evergreen - katalog:.....	22
Obrázek 8: Bibframe model:	29
Obrázek 9: Bibframe model (Annotation): (Library of Congress, 2012)	30
Obrázek 10: UML class diagram převodníku:	36
Obrázek 11: Vývojový diagram:	37
Obrázek 12: Uživatelské rozhraní převodníku:.....	40
Obrázek 13: Převedení názvu díla z UNIMARC do MARC21:	43
Zdrojový kód 1: metoda Převed'():	40
Zdrojový kód 2: metoda nactiSoubor():.....	41
Zdrojový kód 3: příklad – if podmínky:.....	42
Zdrojový kód 4: metoda vytvořSoubor():	43
Zdrojový kód 5: metoda zapsani():	44

Seznam příloh

Příloha 1: UML class diagram

Příloha 2: metoda sort()

Příloha 3: metoda kodToM21()

Příloha 4: Clavius

Příloha 5: www katalog – Clavius

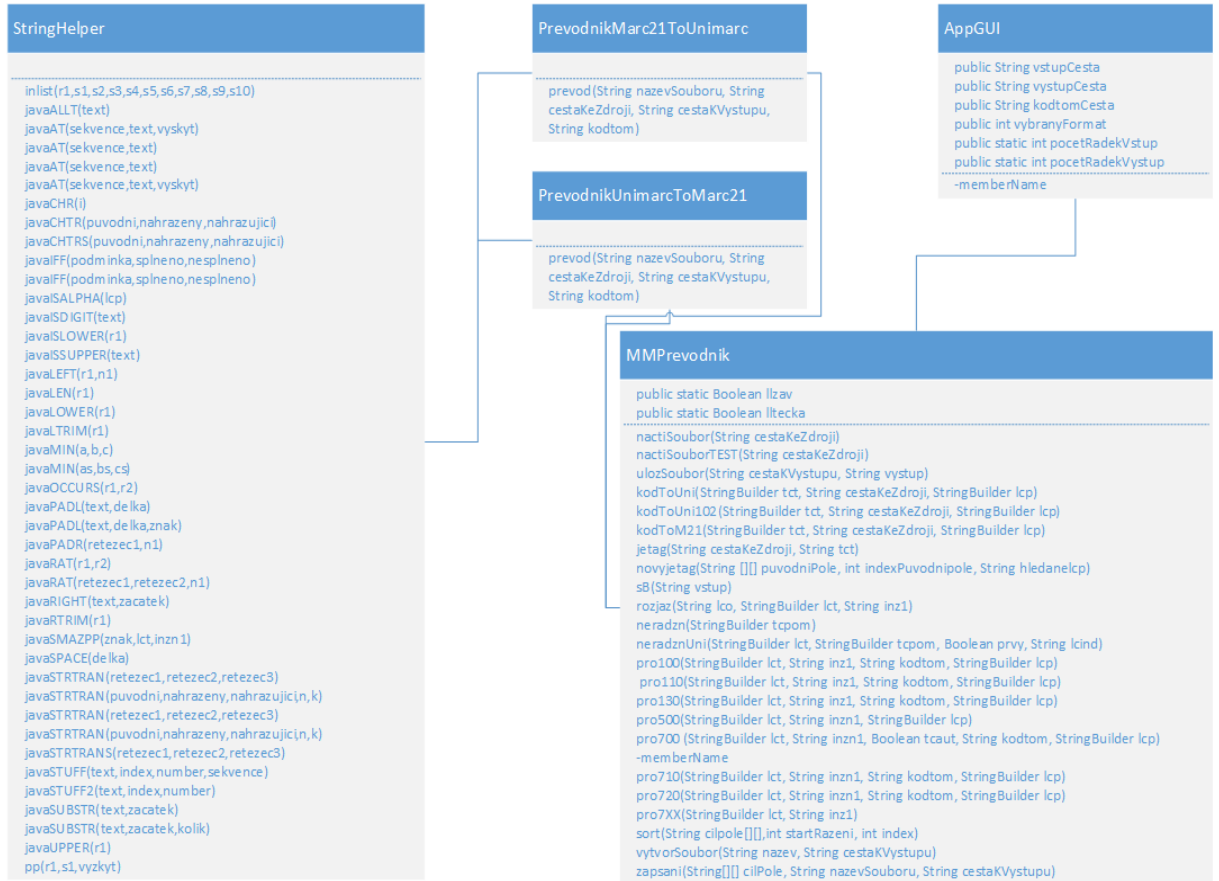
Příloha 6: www katalog – Carmen

Příloha 7: Tritius

Příloha 8: UnimarcToMarc21Converter – na přiloženém CD

Přílohy

Příloha 1: UML class diagram




```

        kolikrat++;

        StringBuilder pomocna = new StringBuilder(cilpole[i][0]);

        pomocna.deleteCharAt(0);

        cilpole[i][0] = pomocna.toString();}

    if(cilpole[i][0].startsWith("0")){

        kolikrat++;

        StringBuilder pomocna = new StringBuilder(cilpole[i][0]);

        pomocna.deleteCharAt(0);

        cilpole[i][0] = pomocna.toString();}

    if(cilpole[j+1][0].startsWith("0")){

        kolikrat2++;

        StringBuilder pomocna2 = new StringBuilder(cilpole[j+1][0]);

        pomocna2.deleteCharAt(0);

        cilpole[j+1][0] = pomocna2.toString();}

    if(cilpole[j+1][0].startsWith("0")){

        kolikrat2++;

        StringBuilder pomocna2 = new StringBuilder(cilpole[j+1][0]);

        pomocna2.deleteCharAt(0);

        cilpole[j+1][0] = pomocna2.toString();}

    if((!cilpole[i][0].equals("")) & (!cilpole[j+1][0].equals(""))&
(!cilpole[j+1][0].equals("LAB"))& (StringHelper.obsahujePismena(cilpole[j+1][0]) == false)&
(StringHelper.obsahujePismena(cilpole[i][0]) == false)){

        int o = Integer.valueOf(cilpole[i][0]);

        int p = Integer.valueOf(cilpole[j+1][0]);

        if(o < p){

            int tmp = o;

            o = p;

            p = tmp;

            StringBuilder pomocnas1 = new StringBuilder();

            pomocnas1.append(o);

```

```
StringBuilder pomocnas2 = new StringBuilder();

pomocnas2.append(p);

if(kolikrat == 1){

    String pom = pomocnas1.toString();

    pom = "0" + pom;

    StringBuilder pom2 = new StringBuilder();

    pom2.append(pom);

    pomocnas1 = pom2;}

if(kolikrat ==2 ){

    String pom2 = pomocnas1.toString();

    pom2 = "00" + pom2;

    StringBuilder pom3 = new StringBuilder();

    pom3.append(pom2);

    pomocnas1 = pom3;}

if(kolikrat2 == 1){

    String pom4 = pomocnas2.toString();

    pom4 = "0" + pom4;

    StringBuilder pom5 = new StringBuilder();

    pom5.append(pom4);

    pomocnas2 = pom5;}

if(kolikrat2 == 2){

    String pom5 = pomocnas2.toString();

    pom5 = "00" + pom5;

    StringBuilder pom6 = new StringBuilder();

    pom6.append(pom5);

    pomocnas2 = pom6;}

cilpole[i][0] = pomocnas1.toString();

String prvek2 = cilpole[i][1];

String prvek3 = cilpole[i][2];
```



```

        cilpole[i][1] = cilpole[j+1][1];

        cilpole[i][2] = cilpole[j+1][2];

        cilpole[j+1][0] = pomocnas2.toString();

        cilpole[j+1][1] = prvek2;

        cilpole[j+1][2] = prvek3;}

    else{

        cilpole[i][1] = cilpole[i][1];

        cilpole[i][2] = cilpole[i][2];}

    }}}

    catch(NullPointerException ex){

Logger.getLogger(MMPrevodnik.class.getName()).log(Level.SEVERE,

"Neočekávaná chyba1 ---- Nulová hodnota ve výstupním poly", ex);}

        }catch(NumberFormatException ex){

Logger.getLogger(MMPrevodnik.class.getName()).log(Level.SEVERE,

"Neočekávaná chyba", ex);}}}

    //vraceni nul

    for(int u = 1; u < index; u++){

        //System.out.println("Zbavuji null...");

        if(cilpole[u][0] == null){

            cilpole[u][0] = "";

            cilpole[u][1] = "";

            cilpole[u][2] = "";}

        try{

            if(!cilpole[u][0].isEmpty()){

                if(cilpole[u][0].length() == 1){

                    String pomocna = cilpole[u][0];

                    pomocna = "00" + pomocna;

                    cilpole[u][0] = pomocna;}

                if(cilpole[u][0].length() == 2){

```

```
String pomocna = cilpole[u][0];  
  
pomocna = "0" + pomocna;  
  
cilpole[u][0] = pomocna;}}  
  
catch(NullPointerException ex){  
Logger.getLogger(MMPrevodnik.class.getName()).log(Level.SEVERE,  
"Neočekávaná chyba", ex);}}}}
```



```

        pom = StringHelper.javaSTRTRAN(StringHelper.pp(tct, "a", 1), kodtom[j][2],
kodtom[j][1]);

        if(pom.toString().compareTo(StringHelper.pp(tct, "a", 1).toString()) != 0){
            //ret = StringHelper.javaSTRTRAN(tct, StringHelper.pp(tct, "a", 1).toString(),
pom.toString());

            ret = sB(tct.toString().replace(StringHelper.pp(tct, "a", 1).toString(),
pom.toString()));

            }}}}

if((lcp.toString().equals("110") | (lcp.toString().equals("111"))){

    StringBuilder pom2 = new StringBuilder();

    ret = sB(tct.toString().replace("$" + "4" + StringHelper.pp(tct, "4", 1), ""));

    for(int k = 0; k < kodtom.length; k++){

        if(kodtom[k][0].startsWith("7")){

            if(tct.indexOf(kodtom[k][2]) > 0){

                pom2 = StringHelper.javaSTRTRAN(StringHelper.pp(tct, "4", 1), kodtom[k][2],
kodtom[k][1]);

                if(pom2.toString().compareTo(StringHelper.pp(tct, "4", 1).toString()) != 0){

                    ret = StringHelper.javaSTRTRAN(tct, StringHelper.pp(tct, "4", 1).toString(),
pom2.toString());

                    }}}}

        StringBuilder pom2 = new StringBuilder();

        if(lcp.toString().trim().startsWith("7")){

            ret = sB(tct.toString().replace("$" + "4" + StringHelper.pp(tct, "4", 1), ""));

            for(int k = 0; k < kodtom.length; k++){

                if(kodtom[k][0].startsWith("7")){

                    if(tct.indexOf(kodtom[k][2]) > 0){

                        pom2 = StringHelper.javaSTRTRAN(StringHelper.pp(tct, "4", 1), kodtom[k][2],
kodtom[k][1]);

                        if(pom2.toString().compareTo(StringHelper.pp(tct, "4", 1).toString()) != 0){

                            ret = StringHelper.javaSTRTRAN(tct, StringHelper.pp(tct, "4", 1).toString(),
pom2.toString());

```

```
}}}}}
```

```
return ret; }
```

Příloha 4: Clavius

Knihovní systém CLAVIUS uživatel: LA
 Systém Záznam Úpravy Akvizice Katalog Vyhledávání Výstupy Úbýtky Pomůcky Olno Nápvěda

Knihovni katalog - knihy
 Nastavovat Formulář Kopírování

Dneska už se tomu směju

Počet vybraných titulů : 715
 Počet titulů celkem : 726

Věcný popis Soubhrm

Tituly	Swazky	Autoři	Názvové údaje	Údaje o vydání	Údaje o edici	Věcný popis	Soubhrm
Autor	Název	Část	Místo vydání	Nakladatel	Rok		
Němcová, Božena, 1820-1861	Babička	Praha		Academia	2008		
Němcová, Božena, 1820-1861	Babička	Praha		Československý národní divadlo	1980		
Bradbury, Ray, 1920-2012	Martianská kronika	Praha					
Toole, John Kennedy, 1937-1	Spočtení hlupců	Praha					
Blyton, Enid, 1897-1968	Tajemství opuštěného domu	Praha					
Courtier, Jane, 1954-	Zahrádka v květináčích a závěsných koších	Praha					
Hlaváčková, Iva, 1962-	Lepší pozdě než nikdy	Praha					
Tremayne, Peter, 1943-	Zvon malomocného	Praha					
Tremayne, Peter, 1943-	Pán duší	V Praze					
Mandlová, Adina, 1910-1991	Dneska už se tomu směju	Praha					

Omezení : Datum uložení titulu před dnešním datem dnů "9999"

Zadání kombinovaného dotazu

Vyhledávat ve všech dokumentech

Výsledky vyhledání

Zadáání dotazu

Ulož

Načti

Priority má Zároveň

Start hledání

Smaž hodnoty

Vyhledávání

Zadání kombinovaného dotazu

Vyhledávat ve všech dokumentech

Autor začíná Zároveň Nebo

Název začíná Zároveň Nebo

Téma začíná Zároveň Nebo


Signatura svazku začíná Zároveň Nebo

Libovolné pole obsahuje slova začínající Zároveň Nebo

Start hledání


Smaž hodnoty

↑ AAA
Novinky
Výběr katalogu
Rejstřík
Náповěda
Čeština
Přihlášení




Městská knihovna Tábor
Online katalog knihovny


Novinky fondu




Matka a dívky pohádky



Ságařka, ty jsi krásná!



Ságařka, ty jsi krásná!



50 odstínů šedi

Vyhledávání

Hledat

ve všech dokumentech

▼

Vše

Název

Autor


Téma

Kombinované


Regionální autority

Kalendárium


Dokumenty s nejvyšším hodnocením



Harry Potter a věžeň z Azkabanu. 3.
Část: 3
Rowling, J. K., 1965-
Vydáno: Praha : Albatros, 2003
Signatury: LSM, M, PSM, SF



Fiorella a Bratrstvo křišťálu
Vondruška, Vlastimil, 1955-
Vydáno: Praha : Albatros, 2006
Signatury: LSM, M



Květy z půdy. 1. O zradě, pomstě...
Část: 1.

Hodnoceno: 7x

Dostupné

Vybrat

Rezervovat

Hodnoceno: 4x

Dostupné

Vybrat

Rezervovat

Hodnoceno: 4x

Dostupné

Vybrat

Rezervovat

Uživatel

Uživatel

Vybrané položky (0)

Nehledanější

Nehledanější

50 odstínů šedi AIDS dejin
divlá bára farma zvířat
FYZIOTERAPIE Hanson
husitské malířství indiáni jo
nesbo komiks komu zvoní
hrana Kroniky kulička
lakomec malý princ petr a lucie
podle skutečne
udalosti podle skutečné
události Prezidenti proměna
průvodce ztracených
psychologie revizor **rock**
romány pro ženy, 2. svět.
válka, židé středověk vývoj
dítěte **vývojová psychologie**
vývoj socharství

Příloha 7: Tritius

Díla - Procházení - vše										
Id	Typ dokume...	Autorka	Název	Část	Místo vydání	Nakladatel	Rok vydání	Datum vytv...	Vytvořeno ...	Edice
134546	PD	ARTECO - B...	- dílčí průzkumová zpráva				1992	28.08.2011	lanadmin lan...	
136052	PD		- polní náčrt - měřítko 1:500				1922	28.08.2011	lanadmin lan...	
134772	PD	CORONA M...	- průzkumová zpráva				1995	28.08.2011	lanadmin lan...	
134750	PD		- rekonstrukce plynovodu				1994	28.08.2011	lanadmin lan...	
134048	PD	Státní projek...	- situace				1984	28.08.2011	lanadmin lan...	
135830	PD	CORONA M...	- stavebně historický průzkum s...				2000	28.08.2011	lanadmin lan...	
132462	PD	Siavopodnik...	.Oprava fasády - rozpis prací a do...				1957	28.08.2011	lanadmin lan...	
133067	PD	Okresní sta...	.Pohled východní, jižní a severní				2011	09.04.2013	Martina Nov...	
1058403	PP		[Chýnov čp. 35 - stav před opravo...				1870-1952	09.04.2013	Martina Nov...	
1058425	IIP	Hansová, J...	[Katastrální úřad v Českém Krum...				2012	29.10.2012	Martina Nov...	
722292	PD	Dolejšová, L...	[Rekonstrukce venkovské usedl...		Husinec		2005-2010	28.08.2011	lanadmin lan...	
132368	RZ	Matějček, Jiří	[Restaurová zpráva - závěsný...				2009-2010	28.08.2011	lanadmin lan...	
132373	RZ	Kobylika, Pavel	[Restaurová zpráva barokní k...				2009-2010	28.08.2011	lanadmin lan...	
132377	RZ	Kobylika, Pavel	[Restaurová zpráva barokní k...				2009-2010	28.08.2011	lanadmin lan...	
132374	RZ	Kobylika, Pavel	[Restaurová zpráva barokní k...				2009-2010	28.08.2011	lanadmin lan...	
132376	RZ	Kobylika, Pavel	[Restaurová zpráva barokní k...				2009-2010	28.08.2011	lanadmin lan...	
132378	RZ	Kobylika, Pavel	[Restaurová zpráva barokní k...				2009-2010	28.08.2011	lanadmin lan...	
132379	RZ	Kobylika, Pavel	[Restaurová zpráva konzola s...			[s. d.]	2009-2010	28.08.2011	lanadmin lan...	
132375	RZ	Kobylika, Pavel	[Restaurová zpráva konzola s...				2009-2010	28.08.2011	lanadmin lan...	
132380	RZ	Kobylika, Pavel	[Restaurová zpráva konzola s...				2009-2010	28.08.2011	lanadmin lan...	
132346	RZ	Fiala, Luděk	[Restaurová zpráva paláš, tes...				2010	28.08.2011	lanadmin lan...	
132366	RZ	Náprstková...	[Restaurová zpráva soubor g...				2010	28.08.2011	lanadmin lan...	
132370	RZ	Trmalová, O...	[Restaurová zpráva soubor o...				2010	28.08.2011	lanadmin lan...	

Zobrazují 1 - 23 z 16041

Uživatel: lanadmin lanadmin (Poslední přihlášení: 18.02.2014 09:19:49) Knihožna: NPÚ ÚOP v Českých Budějovicích

Odhlásit