Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Diplomová práce

Systém pro automatickou kontrolu samostatných prací vytvořených v MS Access

Plzeň, 2018 Vojtěch Kinkor

Zadání

1. Seznamte se s formátem souboru MS Access .accdb a možnostmi jeho čtení.
2. Seznamte se validátorem na portálu ZČU.
3. Vytvořte konfigurovatelný systém pro kontrolu samostatných prací vytvořených v MS Access se zaměřením na splnění zadání a plagiarismus.
4. Část systému pro kontrolu splnění zadání adaptujte pro validátor.
5. Celý systém pečlivě otestujte.

**< tato strana bude nahrazena originálním zadáním / kopií ! >**

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne <TODO: 0. 0. 2018>

Vojtěch Kinkor

Abstract

**System for Automatic Checking of Student Works Created in MS Access**

text

Abstrakt

**Systém pro automatickou kontrolu samostatných prací vytvořených   
v MS Access**

text

Obsah

[1 Úvod 1](#_Toc512982327)

[2 Databázový software Microsoft Access 2](#_Toc512982328)

[2.1 Základní informace 2](#_Toc512982329)

[2.2 Objekty uložené v databázi 2](#_Toc512982330)

[2.2.1 Tabulky 2](#_Toc512982331)

[2.2.2 Dotazy 5](#_Toc512982332)

[2.2.3 Formuláře 6](#_Toc512982333)

[2.2.4 Sestavy 7](#_Toc512982334)

[2.2.5 Skryté systémové tabulky 7](#_Toc512982335)

[2.3 Metadata databázových souborů 7](#_Toc512982336)

[2.4 Formát ACCDB a možnosti jeho čtení 8](#_Toc512982337)

[2.4.1 ODBC 9](#_Toc512982338)

[2.4.2 Microsoft Office Interop 9](#_Toc512982339)

[2.4.3 MDB Tools 10](#_Toc512982340)

[2.4.4 MDB Tools Java 10](#_Toc512982341)

[2.4.5 Jackcess 11](#_Toc512982342)

[2.4.6 JDBC 11](#_Toc512982343)

[3 Portál ZČU 13](#_Toc512982344)

[3.1 Základní informace 13](#_Toc512982345)

[3.2 Validátor studentských prací 13](#_Toc512982346)

[3.3 Vytvoření nové validační domény 15](#_Toc512982347)

[4 Analýza kontroly prací 16](#_Toc512982348)

[4.1 Požadavky na řešení 16](#_Toc512982349)

[4.2 Případy užití 16](#_Toc512982350)

[4.3 Validace databáze 16](#_Toc512982351)

[4.4 Vyhodnocení plagiarismu 16](#_Toc512982352)

[5 Implementace systému pro automatickou kontrolu prací 17](#_Toc512982353)

[5.1 Použité technologie 17](#_Toc512982354)

[5.2 Struktura aplikace 17](#_Toc512982355)

[5.3 Validace databáze 17](#_Toc512982356)

[5.4 Implementovaná validační pravidla 17](#_Toc512982357)

[5.5 Hledání podobností a detekce plagiarismu 17](#_Toc512982358)

[5.6 Grafické rozhraní 17](#_Toc512982359)

[5.7 Adaptace pro validátor portálu ZČU 17](#_Toc512982360)

[6 Testování vytvořeného systému 18](#_Toc512982361)

[6.1 Validační pravidla 18](#_Toc512982362)

[6.2 Konzolová aplikace pro validátor portálu ZČU 18](#_Toc512982363)

[6.3 Grafické rozhraní 18](#_Toc512982364)

[7 Závěr 19](#_Toc512982365)

[Reference 20](#_Toc512982366)

[Přílohy 22](#_Toc512982367)

[A Uživatelská příručka 22](#_Toc512982368)

[Spuštění a kompilace nástroje 22](#_Toc512982369)

[Obsluha nástroje 22](#_Toc512982370)

[B Obsah přiloženého média 22](#_Toc512982371)

# Úvod

# Databázový software Microsoft Access

## Základní informace

Microsoft Access je nástroj řadící se mezi takzvané systémy řízení báze dat (SŘBD či DBMS – database management system). Jedná se o software, který umožňuje práci s relačními databázemi. Je součástí kancelářského balíku Microsoft Office, případně prodáván i samostatně. Pro vytváření a správu databáze nabízí uživatelům přehledné grafické rozhraní [[1](#_toc_1)].

Aplikace používá pro ukládání dat technologii Microsoft Jet Database Engine, v novějších verzích poté nazývanou Access Database Engine. Jednotlivé databáze jsou typicky uloženy v jediném souboru ve formátu ACCDB, nebo MDB [[2](#_toc_2)].

## Objekty uložené v databázi

V následujících podkapitolách jsou uvedeny různé objekty, které mohou být součástí Microsoft Access databází.

### Tabulky

Tabulky jsou stěžejní součást každé databáze. Lze je definovat jako strukturovanou kolekci dat. Skládá se ze sloupců a řádků (též záznamů) a v rámci databáze má unikátní název [[1](#_toc_1) str. AC 4].

Sloupce tabulky

Struktura tabulky je definována sloupci, které mají specifikovaný název (unikátní v rámci tabulky) a datový typ. Microsoft Access[[1]](#footnote-2) podporuje následující datové typy [[1](#_toc_1) str. AC 57, [3](#_toc_3)]:

* *Automatické číslo* – pro každý nový záznam se automaticky nastaví na následující hodnotu posloupnosti, nebo na náhodné číslo (dle nastavení). Typicky se využívá pro sloupec označený jako primární klíč (viz dále).
* *Číslo* – rozsah a typ (celočíselné/s desetinnou čárkou) lze zvolit ve vlastnostech sloupce.
* *Krátký text* (dříve Text) – text do délky 255 znaků.
* *Dlouhý text* (dříve Memo) – text do velikosti 1 GB.
* *Datum a čas*.
* *Měna*– specializovaný případ číselného datového typu s fixní desetinnou čárkou (uchovává 4 desetinná místa).
* *Ano/ne* – uchovává hodnotu -1 (Ano) nebo 0 (Ne); v rámci Microsoft Access zobrazeno jako zaškrtávací pole (*checkbox*).
* *Hypertextový odkaz*.
* *Objekt OLE*– umožňuje vložit speciální objekty, například obrázek, jiný dokument, či odkaz na soubor.
* *Příloha* – umožňuje vložit libovolný soubor jako součást záznamu. Jedná se o univerzálnější možnost k předchozímu.
* *Počítané* – automatické vložení hodnoty vypočítané na základě zadaného vzorce.

Každému sloupci lze dále nastavit různé vlastnosti dle vybraného datového typu – typicky se jedná o ověřovací pravidla (validace vstupu od uživatele ještě před přidáním záznamu do databáze), výchozí hodnotu a dále nastavení zobrazení v tabulce (formátování, zarovnání, titulek po najetí myší, atp.).

Primární klíč

Tabulka může mít primární klíč – typicky se jedná o sloupec, jehož hodnoty jsou unikátní a vždy zadané (tzv. *not null*). V případě, že vytvoříme primární klíč pomocí více sloupců, nazýváme jej složeným primárním klíčem [[1](#_toc_1) str. AC A4].

Primární klíč slouží k jednoznačnému určení konkrétního záznamu v tabulce, čehož se využívá při vytváření dotazů nebo tvoření relačních vazeb mezi tabulkami. Pro vytváření primárních klíčů se obvykle používá datový typ Automatické číslo, který každému záznamu přiřadí unikátní celé číslo. Často bývá takový sloupec pojmenován „ID“ (*identification*) [[1](#_toc_1) str. AC A3-A4].

Relace mezi tabulkami a cizí klíče

V případě, že chceme propojit více tabulek mezi sebou, využijeme tzv. relačních vazeb, zkráceně relací. Jedná se o situaci, kdy záznam v tabulce A odkazuje („má referenci“) na jeden konkrétní záznam v tabulce B. To lze obecně zajistit přidáním tzv. *cizího klíče*do tabulky A – sloupce, který bude obsahovat pouze hodnoty primárního klíče tabulky B (příp. skupiny sloupců, pokud se jedná o složený primární klíč). Rozlišují se tři typy relačních vazeb [[1](#_toc_1) str. AC A5-A8, [4](#_toc_4) str. 419]:

* *Relace typu 1:1* – jednomu záznamu v tabulce A odpovídá žádný, či právě jeden záznam v tabulce B. Typicky je tato relace vhodná v situaci, kdy jen málo záznamů odkazuje do druhé tabulky. Pro cizí klíče zde platí, že jejich hodnoty jsou v rámci tabulky unikátní. Alternativně lze pro referencování použít primární klíče obou tabulek (viz obrázek 2.1). [[4](#_toc_4) str. 420].



Obrázek . – model relace 1:1 využívající   
v obou tabulkách primární klíč.

* *Relace typu 1:N* – k více záznamům v tabulce A lze přiřadit jeden záznam z tabulky B. Tato vazba je vždy realizována pomocí již zmíněných cizích klíčů. Jedná se o nejčastěji využívanou vazbu [[4](#_toc_4) str. 421].



Obrázek . – model relace 1:N; cizí klíč *Kolej ID* v tabulce *Student* referencuje primární klíč tabulky *Kolej*.

* *Relace typu M:N* – k M záznamům v tabulce A lze přiřadit N záznamů z tabulky B. Relace se realizuje pomocí *spojové tabulky* (též rozkladové) a dvojice relací 1:N. Spojová tabulka obvykle obsahuje pouze sloupce cizích klíčů [[4](#_toc_4) str. 422].



Obrázek . – model relace M:N, která je tvořena dvěma   
relacemi 1:N na spojovou tabulku *StudentPředmět*.

Relace mezi tabulkami mohou zajišťovat *referenčním integritu*. Cílem je zabránit odkazování na neexistující záznam (a rovněž tedy vzniku osiřelých záznamů, na které byly všechny reference zrušeny). Integritní pravidlo může dále zajistit kaskádovou aktualizaci polí – pokud se změní hodnota primárního klíče, změní se automaticky hodnota u všech záznamů, které na záznam odkazují. Stejně tak může zajistit kaskádové odstranění souvisejících záznamů – v případě smazání záznamu budou smazány i všechny další, které na právě tento záznam odkazovaly [[1](#_toc_1) str. AC A11].

### Dotazy

Dotazy slouží k získávání, přidávání, mazání či upravování dat v databázi. Microsoft Access umožňuje ukládání dotazů do databáze – lze tedy vytvořit dotazy pro usnadnění následné práce s daty [[1](#_toc_1) str. AC 124].

Dotazy mohou mít parametry, které lze využít např. pro filtrování záznamů v rámci tabulky nebo nové hodnoty při vkládání/upravování záznamů. Uživatel je pak při spuštění dotazu vyzván k zadání konkrétních hodnot parametrů [[1](#_toc_1) str. AC 249].

Podporovány jsou následující druhy dotazů:

* *Výběrové (SELECT)* – jedná se o dotaz, jehož výsledkem je množina vybraných záznamů. Struktura je dána dotazem – jednotlivé sloupce mohou pocházet z různých tabulek, či být spočítané „za běhu“. Obecně lze považovat výběrový dotaz za analogii k databázovým pohledům [[1](#_toc_1) str. AC 124-128].
* *Vytvářecí (MAKE TABLE)* – pracuje na stejném principu jako výběrový, výsledek dotazu však není ihned zobrazen uživateli, ale uložen do nové tabulky [[1](#_toc_1) str. AC 500, 501].
* *Přidávací (APPEND)*– slouží pro vkládání nových záznamů do existujících tabulek [[1](#_toc_1) str. AC 500, 506].
* *Aktualizační (UPDATE)*– umožňuje úpravu hodnot již existujících záznamů v tabulkách [[1](#_toc_1) str. AC 500, 512].
* *Křížový (CROSSTAB)*– výsledkem dotazu je tzv. kontingenční tabulka zobrazující data v kompaktní podobě. Typicky se používá například pro sumarizaci hodnot, nalezení průměrů, maximálních hodnot, atp. [[1](#_toc_1) str. AC 254, 256].

### Formuláře

Formuláře poskytují přívětivé rozhraní pro vkládání či editaci záznamů v tabulkách. Grafické rozhraní je plně konfigurovatelné a umožňuje tedy jednotlivá pole záznamů různě seskupovat, přidat popisky, či některá úplně skrýt. Formuláře jsou v databázi opět uloženy pod unikátním názvem [[1](#_toc_1) str. AC 179].

Access umožňují vytvořit formuláře různých druhů [[1](#_toc_1) str. AC 180-200]:

* Formuláře pro editaci jednotlivých záznamů (dále označované jako standardní).
* Navigační formuláře, které poskytují možnost přepínání mezi různými formuláři a umožňují tak vytvořit komplexní rozhraní pro správu celé databáze.
* Formuláře zobrazující více položek (záznamů) najednou.
* Datové listy, které vypadají podobně jako zobrazení tabulky (tedy tabulka, kde každý řádek odpovídá jednomu záznamu), ale zachovávají možnost upravovat zobrazená pole.
* Rozdělené formuláře, které jsou kombinaci standardních formulářů v jedné části a datového listu v druhé části obrazovky.
* Modální dialogová okna, která mají stejné možnosti jako standardní formuláře, ale zobrazují se v samostatném okně a jsou

### Sestavy

Sestavy slouží pro vytváření výpisů dat z databáze v přívětivé podobě, zobrazující typicky více záznamů na jedné straně, na rozdíl od formulářů ale neumožňuje editaci dat. Často se využívá pro následné vytisknutí. Při návrhu se definuje záhlaví a zápatí stránek a rozložení prvků pro každý záznam („řádek“ sestavy) [[1](#_toc_1) str. AC 208].

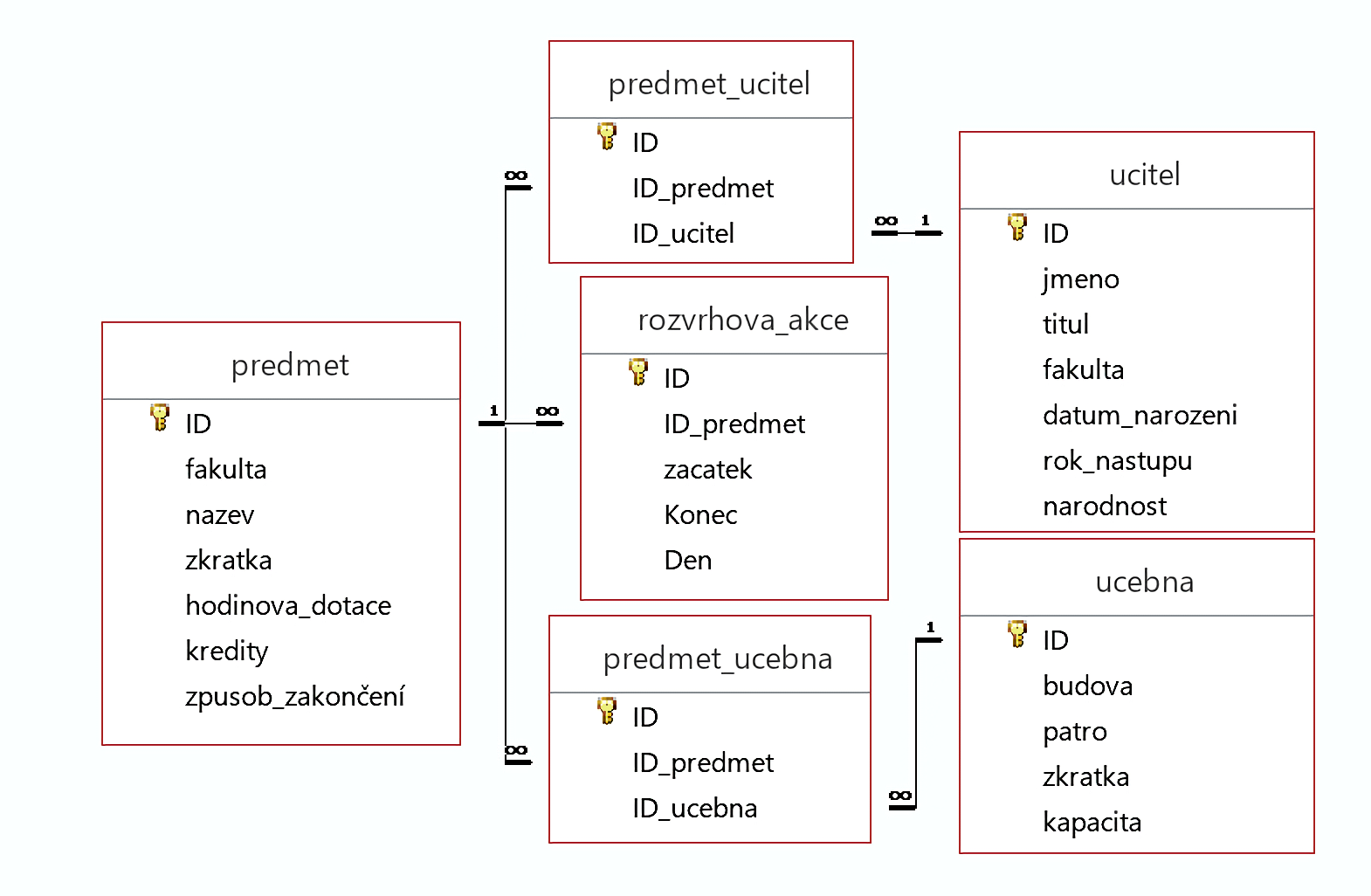
### Skryté systémové tabulky

Databáze dále obsahují několik skrytých systémových tabulek, jejichž název vždy začíná prefixem MSys. Například v tabulce MSysRelationships jsou uloženy všechny relace mezi objekty (tabulkami a dotazy). Jednou ze zajímavějších tabulek je MSysObjects obsahující seznam všech objektů databáze a k nim i různé další údaje, čehož lze využít při hledání metadat (viz kapitola 2.3) [[1](#_toc_1) str. AC 582, [5](#_toc_5)].

## Metadata databázových souborů

Kromě samotných uživatelských dat a databázových objektů lze v databázi nalézt i další doplňující údaje, obecně označované jako metadata [[5](#_toc_5), [6](#_toc_6)]:

* *Autor databáze* – jméno uživatele a název organizace.
* *Formát souboru* – verze aplikace, ve které byla databáze vytvořena, nastavení režimu kompatibility, atp.
* *Údaje o databázi a objektech*– datum vytvoření a poslední úpravy.
* *Uživatelské nastavení aplikace*– nastavení navigačního panelu (řazení/seskupení/šířka/…) a jiných prvků GUI až např. grafické rozvržení relací (viz obr. 4).



Obrázek . – grafické rozvržení relací mezi tabulkami v aplikaci Microsoft Access 2016,   
které lze rovněž považovat za metadata uložená v databázi.

## Formát ACCDB a možnosti jeho čtení

Nativním formátem pro ukládání Access databází je od verze 2007 ACCDB, v předchozích verzích byl hlavním formátem MDB. Oba jsou založeny na technologii Jet (u formátu ACCDB také označované jako Access Database Engine) a jsou si tedy technologicky podobné. Z uživatelského hlediska jsou rozdíly zejména v možnostech zabezpečení dat [[2](#_toc_2), [6](#_toc_6), [7](#_toc_7)].

Jedná se o proprietární binární formáty vyvíjené společností Microsoft bez dostupné specifikace. Jediným oficiálním nástrojem pro správu je právě Microsoft Access, pro přístup k datům z jiných aplikací pak technologie ODBC a OLE DB [[6](#_toc_6)].

To velmi omezuje možnosti programového přístupu k databázím – pokud bychom vzali v potaz pouze oficiální nástroje, jsme limitováni na systémy s nainstalovanou aplikací Microsoft Access (a tím pádem i operačním systémem Microsoft Windows). V současné době jsou však dostupné i nástroje vzniklé na základě reverzního inženýrství formátů ACCDB/MDB bez závislosti na programovém vybavení počítače.

V následujících podkapitolách jsou zmíněny všechny možnosti čtení souborů ACCDB včetně výhod a nevýhod, jaké přináší.

### ODBC

ODBC (Open Database Connectivity) je standardizované API pro přístup k datům uloženým v databázích. Připojení ke konkrétním databázím je zajištěno speciálními ovladači, které lze do systému doinstalovat. Pro komunikaci skrze ODBC se typicky využívá jazyk SQL (Structured Query Language), ovladač poté zajistí vykonání příkazu nad konkrétní databází [[8](#_toc_8)].

Pro přístup k ACCDB databázím v rámci OS Microsoft Windows se využívají ovladače Access Database Engine nainstalované spolu s aplikací Microsoft Access, případně ze samostatného distribučního balíku. Pro další platformy existují komerční Access ODBC ovladače[[2]](#footnote-3). Vzniká zde tedy závislost na dostupnosti ovladače, přičemž v určitých případech může být problém jej do systému doplnit [[2](#_toc_2)].

Zásadní nevýhodou přístupu k datům přes ODBC API jsou omezení vyplývající z univerzálnosti metody. Jednoduše lze pracovat pouze s daty v tabulkách a není možné přímo přistupovat k dalším uloženým objektům. Jedinou možnost je využít skryté systémové tabulky, pomocí kterých lze zjistit alespoň existenci objektů [[8](#_toc_8)].

Novější obdobnou technologií je OLE DB (*Object Linking and Embedding, Database*), vyvinuté firmou Microsoft původně jako nástupce ODBC; a dále technologie ADO a ADO.NET stavící nad ODBC, resp. OLE DB [[9](#_toc_9)]. Z hlediska způsobu použití a nabízených funkcí pro čtení souboru ACCDB jsou však všechny technologie shodné.

### Microsoft Office Interop

Aplikace z balíku Microsoft Office lze programově ovládat pomocí technik obecně označovaných jako *interoperability* (zkráceně *interop*). Typicky se využívají proprietární technologie COM (Common Object Model) a OLE (Object Linking and Embedding) vyvinuté firmou Microsoft [[9](#_toc_9)]. Dále jsou poskytovány *Primary Interop Assemblies*– knihovny určené pro použití na platformě .NET (tedy v tzv. řízeném kódu) obalující COM volání do objektového rozhraní. V současné době jsou tyto knihovny nejjednodušší cestou pro programové ovládání aplikací Microsoft Office (mj. se využívají i pro psaní doplňků, *plug-inů*, pro jednotlivé Office aplikace) [[10](#_toc_10)].

Tato technika oproti ODBC umožňuje kompletní správu databáze vč. všech dostupných objektů bez nutnosti analyzovat obsah systémových tabulek. Zůstává zde však omezení na systémy, kde je nainstalovaný Microsoft Access. Jde rovněž o poměrně pomalý přístup, jelikož „interop kód“ de-facto jen ovládá aplikaci Microsoft Access spuštěnou na pozadí.

### MDB Tools

Jedná se o open-source sadu nástrojů pro práci se soubory Microsoft Access, respektive Jet databázemi ve formátu MDB, jejíž vývoj započal již v roce 2000. Vzhledem k uzavřenosti formátu vznikla většina nástrojů technikami reverzního inženýrství, není tedy zaručena stoprocentní funkčnost a kompatibilita [[11](#_toc_11)].

Nástroje jsou napsány v jazyce C a mají konzolové rozhraní, dále existuje několik grafických nadstaveb pro prohlížení Access souborů. Součástí projektu je i dokument popisující strukturu a klíčové části Jet databází [[5](#_toc_5)]. V posledních letech probíhá vývoj pomalým tempem a podpora novějších verzí Access databází včetně formátu ACCDB není zaručena [[12](#_toc_12)]. Hlavní výhodou nástrojů je nezávislost na konkrétní platformě a externích knihovnách.

### MDB Tools Java

V roce 2004 začala *portace* nástrojů MDB Tools pro jazyk Java, vývoj však již po roce ustal [[13](#_toc_13)]. Následně vzniklo několik *forků* (projektů založených na kódu původního projektu), nejaktuálnější z nich lze nalézt pod názvem OME MDB Tools. Vývoj těchto projektů je ale spíše pomalý – poslední větší aktualizace proběhla v roce 2016 [[14](#_toc_14)].

Oproti dále uvedené knihovně Jackcess nabízí méně možností a použité je značně komplikované [[15](#_toc_15)]. Překážkou je chybějící dokumentace jak samotných nástrojů, tak i jejich programového kódu.

### Jackcess

Jackcess je Java knihovna poskytující čisté objektové rozhraní pro práci s Microsoft Access databázemi. Její vývoj započal v roce 2005 v rámci open-source projektu OpenHMS zaštítěného firmou Health Market Science, Inc a funguje na stejných principech jako nástroje MDB Tools, kterými se vývojáři na počátku inspirovali [[16](#_toc_16), [17](#_toc_17)].

Kromě čtení dat z tabulek umožňuje i základní editaci struktury databáze, výpis všech relací mezi tabulkami a výpis uložených dotazů. Díky přístupu ke skrytým systémovým tabulkám lze vyhledat i uložené formuláře a sestavy (reálně však lze zjistit pouze jejich existenci). Podporuje Access databáze ve verzích 2000 až 2016 (ve formátu MDB i ACCDB) a ve verzi 97 v režimu pro čtení. Knihovna neobsahuje rozhraní pro spouštění SQL dotazů, neumožňuje tedy ani vyhodnocení uložených dotazů [[17](#_toc_17)].

Zásadního výhodou pro potřeby této práce je přenositelnost knihovny (nezávislost na platformě) a aktivní vývoj – tím pádem i podpora nejnovějších verzí Access databází. Vzhledem k distribuci v podobě samostatné Java knihovny (rovněž dostupné v Maven repozitářích) je její použití ve vlastní aplikaci jednoduché [[18](#_toc_18)].

Pro knihovnu existuje rozšíření nazvané Jackcess Encrypt umožňující správu databází zašifrovaných heslem. Podporuje některé formy šifer aplikací Microsoft Access a Microsoft Money [[19](#_toc_19)].

### JDBC

Java Database Connectivity je API pro přístup k relačním databázím a tedy obdobou technologie ODBC pro programovací jazyk Java. API je standardní součástí platformy Java SE. Připojení ke konkrétní databázi je opět zajištěno ovladači určenými pro konkrétní typ databází [[20](#_toc_20)].

Microsoft neposkytuje vlastní JDBC ovladač pro práci s Access/Jet databázemi, existují ale speciální ovladače, tzv. *JDBC-ODBC bridge,* umožňující použití ODBC ovladačů. V rámci platformy Java SE do verze 1.7 byl takový ovladač standardní součástí; do novějších verzí jej lze překopírovat (jde ale o neoficiální postup bez záruky funkčnosti) nebo nahradit komerčními alternativami [[8](#_toc_8), [21](#_toc_21)].

Dále existuje několik JDBC „nativních“ ovladačů pro práci s Microsoft Access databázemi. Jde zejména o open-source projekt UCanAccess, který využívá již zmíněnou knihovnu Jackcess [[22](#_toc_22)]. K dispozici jsou rovněž komerční ovladače [[3]](#footnote-4),[[4]](#footnote-5),[[5]](#footnote-6).

Stejně jako u technologie ODBC jsou největší nevýhodou omezení v důsledku univerzálnosti přístupu; jinými slovy – jednoduše pracovat je možné pouze s daty v tabulkách a k ostatním objektům databází je přístup obtížný či nemožný [[20](#_toc_20)]. Výhodou je lepší přenositelnost na jiné platformy.

# Portál ZČU

## Základní informace

Obecně termínem portál označujeme webovou aplikaci, která uživateli poskytuje jednotným způsobem a centralizovaně informace z různých zdrojů, které uživatele zajímají nebo se ho nějakým způsobem týkají [[23](#_toc_23)].

Portál ZČU[[6]](#footnote-7) je informační zdroj pro studijní účely používaný studenty i zaměstnanci Západočeské Univerzity. Zastřešuje různé klíčové aplikace univerzity – z pohledu studentů se jedná zejména o *IS/STAG* (Informační systém studijní agendy, umožňující zápis předmětů, zkoušek, prohlížení rozvrhů, hodnocení kvality výuky, atp.) a *Courseware* (místo s materiály používanými v rámci výuky předmětů)[[23](#_toc_23)].

Jednotlivé stránky portálu se skládájí z více či méně nezávislých částí, které se nazývají portlety. Jedním z nich je *aplikace pro správu semestrálních prací, jejich odevzdávání a hodnocení*. Vyučujícím slouží k vypisování témat semestrálních prací, do nichž se studenti posléze mohou přihlašovat a odevzdávat své práce. Vyučující si může práce jednotlivě či hromadně stáhnout, schvalovat je nebo vracet k přepracování a zanechávat studentům poznámky a hodnocení[[23](#_toc_23), [24](#_toc_24)].

## Validátor studentských prací

Studenti často po odevzdání své práce prostřednictvím portálu ZČU čekají delší dobu na zkontrolování práce vyučujícím. Pokud práci shledá nevyhovující, vrací ji studentovi k přepracování. Tento proces se navíc může i několikrát opakovat, což znamená pro vyučujícího velkou časovou zátěž a pro studenta čekání v nejistotě.

Cílem validátoru je eliminovat tento proces a již při nahrání automatizovaně kontrolovat, zda je práce „vyhovující“ – to může mít mnoho podob, například [[25](#_toc_25)]:

* **Práce splňuje formální náležitosti:**
  + soubor má požadovaný formát,
  + soubor je vhodně pojmenován,
  + v odevzdaném archivu jsou požadované soubory ve správné adresářové struktuře,
  + v nahraném archivu nejsou žádné nadbytečné soubory.
* **Práce je vyhovující kvalitativně:**
  + dokument splňuje stanovený rozsah (typicky minimální počet slov či stran),
  + zdrojový kód lze zkompilovat,
  + program má totožný výstup, jako referenční program připravený vyučujícím,
  + odevzdaný soubor obsahuje všechny prvky stanovené zadáním.

Automatizace této kontroly poté přináší mnoho výhod [[25](#_toc_25)]:

* Student se okamžitě dozví, zda jeho práce splňuje základní kritéria stanovená učitelem.
* Protože je nevyhovující práce okamžitě odmítnuta, nemůže student zneužít nahrání takové práce k získání dalšího času.
* Pokud není omezen počet odevzdání, může student u určitých úloh zkoušet i různá „inovativní“ řešení, u kterých si není jist správností.
* Vyučující není zatěžován mechanickým kontrolováním formální správnosti.
* K hodnocení se dostanou jen práce s určitou minimální mírou kvality.

<TODO: nevýhody?>

### Validační servery

Validátor je realizován jako služba běžící na samostatném serveru, se kterou aplikace pro odevzdávání semestrálních prací komunikuje. Příprava validačních pravidel a různých nástrojů pro kontrolu prací tedy probíhá odděleně na tomto serveru, který kromě konfigurace přes terminál poskytuje i jednoduché webové rozhraní[[7]](#footnote-8) umožňující správu pravidel a následné testování nad vzorovými pracemi. Souběžně s tímto serverem je dostupný testovací validační server[[8]](#footnote-9) určený pro vývoj nových funkcí a přípravu validačních pravidel, kde nehrozí nebezpečí poškození aktuálně používaných dat, a je tedy vhodný i pro využití v rámci realizace této práce. Služba validátoru studentských prací je napsána v jazyce Java a je spouštěna ve webovém serveru Tomcat 8 na serveru s OS Linux [[25](#_toc_25), [26](#_toc_26)].

### Validační domény

Konkrétní postup pro vyhodnocení, zda je práce „vyhovující“, je určen tzv. *validační doménou*. Ta se skládá z libovolného počtu kroků, jimiž se zajistí validace odevzdané práce, a tedy určení, zda bude práce přijata, či vrácena studentovi. Domény jsou uloženy na validačním serveru, kde je má vyučující možnost libovolně přidávat, upravovat a odebírat. Každá je uložena pod vlastním názvem – pomocí něj ji lze následně nastavit v rámci portletu pro odevzdávání prací a tím zajistit validaci odevzdaných prací [[25](#_toc_25), [27](#_toc_27)].

Jednotlivé kroky validace představují konkrétní akce, které se mají provést. V rámci domény mají unikátní název a volitelný popis. Pro každý krok lze určit podmínku, za které se má akce provést [[27](#_toc_27)]:

* *Vždy*.
* *Nikdy*.
* *Validace ještě neobsahuje chybu* – neboli žádný předchozí krok neskončil chybou.
* *Validace již obsahuje chybu* – alespoň jeden předchozí krok skončil chybou.
* *Vlastní skript* – lze napsat vlastní skript v jazyce JavaScript, který například vyhodnotí výsledek předchozího kroku a určí, zda se má krok provést.

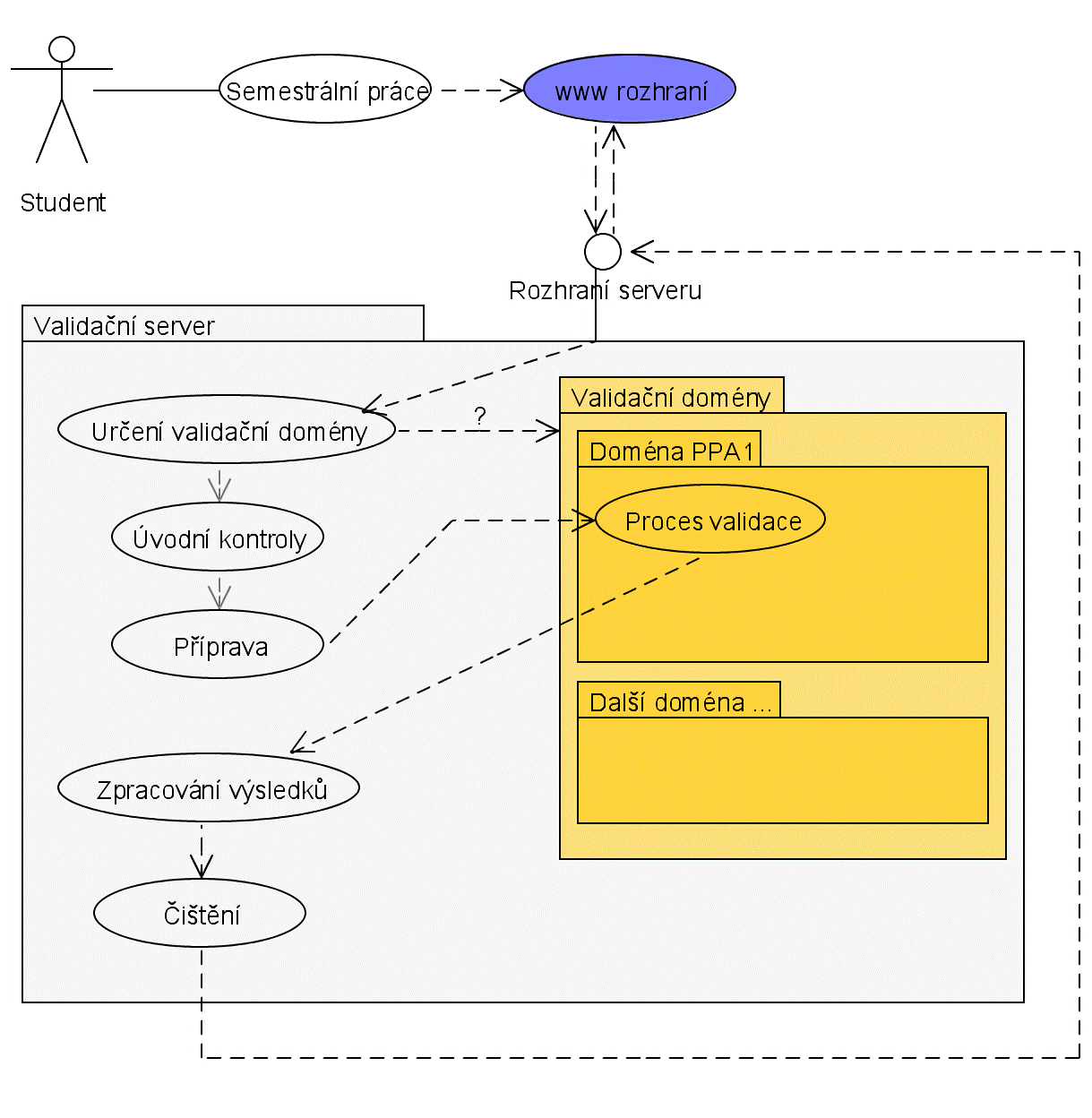
Akce lze rozdělit do několika kategorií [[28](#_toc_28)]:

* *Textové výpisy*– pro zlepšení orientace je vhodné vypisovat relevantní informace k právě probíhané kontrole a jejím výsledkům (tedy např. *„Kontrola počtu stran“* nebo *„Chyba – očekáváno alespoň 20 stran, odevzdaný soubor obsahuje 17“*). Výpisy mohou být informativní, varovné či chybové.
* *Spouštění vlastních akcí* – jedná se o předem definované komplexnější validace, které lze ovládat pomocí parametrů. V současné době je dostupných 31 akcí rozdělených do několika skupin, např.: kompilace souboru, rozbalení ZIP souboru, spočítání slov/stran dokumentu, spuštění Java programu, kontrola názvu souboru, nalezení souboru v adresáři, porovnání PNG souborů, … Validátor lze rozšířit o další vlastní akce naprogramováním a nahráním tzv. *validačních modulů* na validační server [[29](#_toc_29), [26](#_toc_26)].
* *Vlastní skript* – opět lze napsat vlastní skript v jazyce JavaScript. Pro spouštění skriptů se používá knihovna Rhino, která umožňuje přistupovat i k běžným funkcím platformy Java [[27](#_toc_27), [26](#_toc_26)].
* *Skok na jiný krok validace* a *ukončení validace* – ve spojení s podmínkami lze takto řídit průběh validace.

### Princip validace práce

Postup validace práce odevzdané studentem zahrnuje mnoho dílčích úkonů, stručně jej lze popsat následovně (viz též diagram na obrázku 3.1) [[26](#_toc_26)]:

1. Validační server práci přijme a načte přiřazenou validační doménu.
2. Proběhne úvodní kontrola dle konfigurace domény.
3. Příprava na validaci – vytvoření pracovního adresáře, do kterého je práce nahrána a ve které probíhají akce definované doménou (např. rozbalení archivu, atp.).
4. Spuštění validačního procesu validační domény – tedy zpracování všech definovaných kroků. Průběžně se generuje výsledek validace, který může obsahovat informativní nebo chybová hlášení.
5. Ukončení validačního procesu, vyčištění pracovního adresáře (závisí na nastavení validační domény).
6. Výsledek validace je vrácen volající aplikaci – obvykle tedy aplikaci pro odevzdávání studentských prací, který jej zobrazí studentovi. Ten má v tu chvíli možnost zjistit, zda byla jeho práce vyhovující, či nikoliv.



Obrázek . – diagram znázorňující proces validace práce odevzdané studentem [[26](#_toc_26)].

# Analýza řešení kontroly prací

Cílem této práce je vytvoření systému pro kontrolu samostatných prací vytvořených v MS Access, konkrétně pro jejich validaci a rozpoznání plagiarismu; v návaznosti poté systém adaptovat pro validátor studentských prací. Následující podkapitoly se zabývají požadavky, úvodní analýzou a návrhem řešení systému.

## Požadavky na řešení

Požadavky na řešení systému byly stanoveny s ohledem na využití vyučujícími, s důrazem na intuitivní použití výsledného systému a snadnou konfiguraci ve spojení s validátorem studentských prací.

Hlavní části požadovaného systému je aplikace umožňující *validaci Access databází* – tedy kontrolu databází s ohledem na splnění zadání. Vzhledem k různorodosti jednotlivých zadání musí aplikace umožnit pohodlnou konfiguraci této kontroly. To bude spočívat v možnosti zahrnout do kontroly různá pravidla testující obsah databází. Výsledkem kontroly bude označení zadaného databázového souboru jako *vyhovujícího* či *nevyhovujícího*; v druhém případě by měla aplikace zároveň poskytnout informaci o pravidle, které „selhalo“ (a zapříčinilo tedy vyhodnocení databáze jako *nevyhovující*).

Pravidla musí umožňovat kontrolu existence tabulek a relací mezi nimi, případně i dalších objektů uložených v databázi. Dále musí být možné prověřit strukturu tabulek – existenci sloupců konkrétního názvu či datového typu, atp. Bude možné i ověřit počet řádků v tabulce či tabulkách (zejména v podobě kontroly, zda všechny tabulky obsahují alespoň určitý počet řádků).

Další částí aplikace bude detekce plagiarismu, respektive vyhledávání podobností mezi databázovými soubory. Pro databáze může být obtížné definovat, kdy se již jedná o plagiát – například dvě databáze obsahující stejně pojmenované tabulky mohou, ale nemusí být plagiátem. V takovém případě může aplikace pouze upozorňovat na podobnost; pokud se ale podaří na základě dalších informací jednoznačně určit, že se o plagiát jedná, budou tak takové soubory označeny.

Aplikace bude uživatelům nabízet přívětivé grafické rozhraní, pomocí kterého bude moci spravovat aktuální konfiguraci validace – tedy nastavení jednotlivých pravidel. Validaci musí být možné spustit hromadně na více databázemi, výsledek poté bude pro každou z nich určený zvlášť.

Součástí systému bude dále aplikace přizpůsobená pro použití v rámci validátoru studentských prací – konkrétním řešením může být konzolová aplikace, případně nový validační modul. S ohledem na jednoduchou konfiguraci je nutné, aby hlavní část systému (tedy již popsaná aplikace s grafickým rozhraním) umožňovala export konfigurace (pravidel validace) pro použití v části adaptované pro validátor. Výstupem exportu může být konfigurační soubor nebo textový řetězec.

## Případy užití

Diagram na obrázku 4.1 (strana 20) znázorňuje případy užití celého systému z pohledu uživatele i validátoru studentských prací.

Popis diagramu:

* *Import validačních pravidel* – uživatel může načíst dříve vytvořenou sadu pravidel pro validaci databázových souborů ve formátu ACCDB. Stejnou funkcionalitu využívá i část adaptovaná pro validátor studentských prací, které jsou automatizovaně předávány odevzdané studentské práce.
* *Export validačních pravidel* – uživatel může exportovat vytvořenou konfiguraci validace, tedy sadu aktivních validačních pravidel.
* *Nakonfigurování validačních pravidel* – uživatel může vytvořit sadu pravidel ověřující, zda jsou zadané databáze vyhovující. Bližší popis konfigurace je uveden v kapitole 4.3.
* *Validace jednoho či více databázových souborů ­*– uživatel může pomocí aplikace provést validaci databázových souborů. Výstupem akce je informace, zda jednotlivé soubory vyhovují aktuálně nakonfigurovaným pravidlům.
* *Detekce plagiarismu mezi datovými soubory* – uživatel může při zadání více databázových souborů spustit vyhledání plagiátů.
* *Validace jednoho databázového souboru* – komunikace se systémem ze strany validátoru studentských prací; do systému je předán databázový soubor a konfigurace validačních pravidel. Výstupem je opět informace, zda je soubor vyhovující pravidlům.



Obrázek . – diagram případů užití systému pro kontrolu samostatných prací.

## Metoda čtení databázových souborů

Jak již bylo probráno v kapitole 2.4, existuje několik možností pro čtení databázových souborů ve formátu ACCDB. Na základě požadavků na řešení systému lze stanovit kritéria pro výběr metody čtení souborů v podobě následujících vyžadovaných schopností:

* Vypsání názvů tabulek v databázi.
* Vypsání struktury tabulky – zjištění názvů a datových typů sloupců, informace o primárních klíčích.
* Zjištění počtu řádků v tabulce, případně získání všech dat z tabulky.
* Nalezení relací mezi tabulkami v podobě, aby bylo možné určit typ.
* Vypsání dotazů uložených v databázi včetně druhu.

Dalšími omezujícími kritérii jsou funkční požadavky na výsledný systém:

* Spustitelnost na osobních počítačích s OS Microsoft Windows.
* Spustitelnost části adaptované pro validátor na validačním serveru, tj. kompatibilita se službou napsanou v jazyce Java na serveru s OS Linux.

Některé nalezené metody pro čtení souborů ve formátu ACCDB (konkrétně ODBC, Microsoft Office Interop, částečně JDBC) je nutné na základě stanovených kritérií vyřadit. Nesplňují zejména poslední zmíněný bod, tedy spustitelnost na serveru s OS Linux (na kterém není dostupný Microsoft Access ani ODBC ovladače). U metod přístupu prostřednictvím API ODBC a JDBC je navíc problematické pracovat s relacemi a dotazy uloženými v databázích.

Ze zbývajících se jako ideální metoda jeví knihovna *Jackcess* pro platformu Java SE. Vývoj systému v jazyce Java přinese výhodu v multiplatformnosti a bude možné vyvíjet obě části (hlavní aplikaci a část adaptovanou pro validátor) nad společnými základy. Ze všech metod navíc knihovna Jackcess poskytuje nejucelenější přístup k Access databázím a rovněž podporu nejnovějších verzí díky aktivnímu vývoji.

## Validace databázových souborů

<TODO: popsat jednotlivé vyžadované schopnosti a jak se toho docílí v jackcessu>

## Vyhodnocení plagiarismu

## Návrh struktury systému

# Implementace systému pro automatickou kontrolu prací

## Použité technologie

## Struktura aplikace

## Validace databáze

## Implementovaná validační pravidla

## Hledání podobností a detekce plagiarismu

## Grafické rozhraní

## Adaptace pro validátor portálu ZČU

# Testování vytvořeného systému

## Konzolová aplikace pro validátor portálu ZČU

## Grafické rozhraní

## Validační pravidla

# Závěr

# Reference

[1] ADAMSKI, Joseph J.; FINNEGAN, Kathy T. ; SCOLLARD, Sharon. *New perspectives on Microsoft Access 2013: comprehensive.* Stamford, CT: Cengage Learning, 2014. ISBN 978-1-285-09920-0.

[2] Introduction to the Access 2007 file format. *Microsoft Office help and training - Office Support.* [Online] [Citace: 20. 3. 2018]. Dostupné z: <https://support.office.com/en-us/article/Introduction-to-the-Access-2007-file-format-8cf93630-0b68-4a40-a13c-7528b9f074b6>

[3] Data types for Access desktop databases. *Microsoft Office help and training - Office Support.* [Online] [Citace: 22. 3. 2018]. Dostupné z: <https://support.office.com/en-us/article/data-types-for-access-desktop-databases-df2b83ba-cef6-436d-b679-3418f622e482>

[4] CONNOLLY, Thomas; BEGG, Carolyn. *A Practical Approach to Design, Implementation, and Management.* 6. Harlow: Pearson Education Limited, 2014. ISBN 978-1-292-06118-4.

[5] BRUNS, Brian. HACKING. *MDB Tools repository.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://github.com/brianb/mdbtools/blob/master/HACKING>

[6] Microsoft Access ACCDB File Format Family. *Digital Preservation at the Library of Congress.* [Online] [Citace: 20. 3. 2018]. Dostupné z: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000462.shtml>

[7] Which Access file format should I use? *Microsoft Office help and training - Office Support.* [Online] [Citace: 20. 3. 2018]. Dostupné z: <https://support.office.com/en-us/article/which-access-file-format-should-i-use-012d9ab3-d14c-479e-b617-be66f9070b41>

[8] KYLE, Geiger. *Inside ODBC.* Redmond, WA: Microsoft Press, 1995. ISBN 978-1556158155.

[9] ROFF, Jason T. *ADO: ActiveX Data Objects.* místo neznámé: O'Reilly Media, 2001. ISBN 9781491935576.

[10] Office Primary Interop Assemblies. *Microsoft Developer Network.* [Online] [Citace: 02. 04. 2017]. Dostupné z: <https://msdn.microsoft.com/en-us/library/15s06t57.aspx>

[11] BRUNS, Brian. *MDB Tools repository.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://github.com/brianb/mdbtools/>

[12] —. Access 2013 support. *MDB Tools repository.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://github.com/brianb/mdbtools/issues/77>

[13] SMITH, Calvin R. mdbtools is being ported to java. *MDB Tools Discussion.* [Online] 2. 5. 2004 [Citace: 20. 4. 2018]. Dostupné z: <https://sourceforge.net/p/mdbtools/discussion/6688/thread/a543445a/>

[14] Open Microscopy Environment. *OME MDB Tools.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://github.com/ome/ome-mdbtools>

[15] —. ColumnTest source code (ukázka použití). *OME MDB Tools.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://github.com/ome/ome-mdbtools/blob/master/src/main/java/mdbtools/tests/ColumnTest.java>

[16] *Jackcess.* [Online] Health Market Science, 31. 3. 2018 [Citace: 20. 4. 2018]. Dostupné z: <http://jackcess.sourceforge.net/>

[17] Frequently Asked Questions. *Jackcess.* [Online] Health Market Science, 31. 3. 2018 [Citace: 20. 4. 2018]. Dostupné z: <http://jackcess.sourceforge.net/faq.html>

[18] Cookbook. *Jackcess.* [Online] Health Market Science, 31. 3. 2018 [Citace: 20. 4. 2018]. Dostupné z: <http://jackcess.sourceforge.net/cookbook.html>

[19] *Jackcess Encrypt.* [Online] Health Market Science, 9. 10. 2017 [Citace: 20. 4. 2018]. Dostupné z: <http://jackcessencrypt.sourceforge.net/>

[20] MAYDENE FISHER, Jon Ellis, Jonathan Bruce. *JDBC™ API Tutorial and Reference.* Boston, MA: Addison Wesley, 2003. ISBN 0-321-17384-8.

[21] ORACLE. JDBC-ODBC Bridge. *Java SE Documentation.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://docs.oracle.com/javase/7/docs/technotes/guides/jdbc/bridge.html>

[22] AMADEI, Marco. *UCanAccess.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <http://ucanaccess.sourceforge.net/site.html>

[23] Centrum informatizace a výpočetní techniky. *Referenční příručka portálového rozhraní IS/STAG.* Plzeň: Západočeská univerzita, 2009. ISBN 978-80-7043-807-7.

[24] —. Aplikace pro správu semestrálních prací, jejich odevzdávání a hodnocení. *IS/STAG - Helpcentrum.* [Online] Západočeská univerzita [Citace: 23. 4. 2018]. Dostupné z: <https://is-stag.zcu.cz/napoveda/stag-v-portalu/spnew-studium_odevzdavani-praci.html>

[25] HEROUT, Pavel. *Validační server pro studentské projekty.* [Online] [Interní dokument] [Citace: 20. 4. 2018]. Dostupné z: <https://validator-test.zcu.cz/vs/auth/doc/doc/validacni-server-uzivatelsky-popis-2.pdf>

[26] VALENTA, Lukáš; DUDOVÁ, Veronika. *Validační server - manuál.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://validator-test.zcu.cz/vs/auth/doc/index.html>

[27] DUDOVÁ, Veronika. *Webová konfigurace validačního serveru.* Plzeň, 2010. Bakalářská práce. Západočeská univerzita. Fakulta aplikovaných věd. Katedra informatiky a výpočetní techniky. Vedoucí práce Pavel HEROUT.

[28] *Testovací validační server pro studentské projekty.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://validator-test.zcu.cz/>

[29] *Wiki - Validační server a jeho moduly - Redmine.* [Online] [Citace: 20. 4. 2018]. Dostupné z: <https://students.kiv.zcu.cz:3443/projects/validator/wiki>

# Přílohy

## A Uživatelská příručka

### Spuštění a kompilace nástroje

### Obsluha nástroje

Typický postup práce s nástrojem je následující:

1. ...

## B Obsah přiloženého média

Součástí práce je přiložené paměťové médium (DVD) obsahující tyto adresáře a soubory:

* Project/ – adresář obsahující projekt vytvořené aplikace,
* Kinkor\_A16N0040P\_DP.pdf – text této práce ve formátu PDF,
* readme.txt – textový soubor obsahující popis struktury DVD.

Obsah přiloženého média (včetně aktuální verze nástroje) je možné najít též v repozitáři projektu v rámci služby GitHub na adrese:

<https://github.com/ikeblaster/access-validator/>

1. Aktuálně ve verzi 2016. [↑](#footnote-ref-2)
2. Např. viz WWW: https://www.easysoft.com/products/data\_access/odbc-access-driver/ [↑](#footnote-ref-3)
3. Viz WWW: <https://www.easysoft.com/products/data_access/jdbc-access-gateway/> [↑](#footnote-ref-4)
4. Viz WWW: <http://www.hxtt.com/access.html> [↑](#footnote-ref-5)
5. Viz WWW: http://sesamesoftware.com/relational-junction/jdbc-database-drivers-↩ products/relational-junction-mdb-jdbc-driver/ [↑](#footnote-ref-6)
6. Dostupný na adrese <https://portal.zcu.cz/> [↑](#footnote-ref-7)
7. Viz WWW: <https://validator.zcu.cz/> [↑](#footnote-ref-8)
8. Viz WWW: <https://validator-test.zcu.cz/> [↑](#footnote-ref-9)