

## **Matlafous Kartotéka v0.74**

Technická dokumentace



## Obsah:

### 1. Základní informace

1. Použité technologie
2. Dostupné zdroje

### 2. Technický popis Kartotéky

1. Řešení s použitím SQL
  1. Princip fungování Kartotéky
    - A) Databáze
    - B) Soubory
    - C) Mezivrstva
    - D) GUI
    - E) Nástroje
  2. Schéma
  3. Objekt TKartoteka
2. Řešení bez použití SQL

### 3. Soubory

1. Seznam a vysvětlení souborů
2. Jednotlivé unity a seznam jejich procedur

### 4. Poznámky



## 1. Základní informace

### 1.1 Použité technologie

#### Pascal

Pro naprogramování celého projektu byl použit jazyk Pascal, a to ve svém objektovém rozšíření ObjectPascal.

Jako vývojové prostředí, a tedy i překladač, jsem nakonec zvolil Borland TurboDelphi ve verzi 2005. Jedná se o IDE volně šiřitelné, přeložené binární soubory nejsou nijak technicky ani právně omezeny – toto IDE se tedy dá použít i pro vývoj komerčních aplikací. Druhou možností překladače byl FreePascal a jeho objektové IDE Lazarus. Kód FreePascalu je ale mírně odlišný od BorlandPascalu, a je proto i méně podporovaný doplňky třetích stran. Výsledný binární soubor Lazarusu je navíc zpravidla mnohem větší a pomalejší, než výsledek stejného kódu z Borland Delphi. Nutno ovšem podotknout, že na rozdíl od Borland Delphi je Lazarus velmi multiplatformní a celý je šířen pod licencí GNU/gpl.

#### Sqlite 3

Ukládání záznamů bylo dříve realizováno do samostatných zip archivů. Po zvážení negativ tohoto řešení a naopak uvědomění si výhod a zjednodušení, která by přineslo použití databází jsem se rozhodl pro kompletní přepsání Kartotéky na využívání databáze typu SQL. Pro tento účel jsem nakonec vybral systém SQLite. Ten, na rozdíl od komplikovanějších systémů (např. mySql), neběží na bázi klient-server, kde je server spuštěn jako samostatný proces. To je pro desktopovou aplikaci, jakou je Kartotéka, ideální.

Komunikaci mezi Kartotékou a sqlite3.dll zprostředkovává Delphi SQLite Wrapper od Lukáše Gebauera ([www.ararat.cz/doku.php/en:sqlitewrap](http://www.ararat.cz/doku.php/en:sqlitewrap)).

#### 7zip

Pro zálohování byla použita komprimační knihovna 7z.dll od Igora Pavlova ([www.7-zip.org](http://www.7-zip.org)), pro přístup k ní z Kartotéky jsem využil unitu 7-zip Delphi api od Henri Gourvesta ([hgourvest@gmail.com](mailto:hgourvest@gmail.com)).

### 1.2 Dostupné zdroje

Zdrojové kódy Kartotéky, zkompilované binární soubory a dokumentace jsou dostupné na: [www.bitbucket.org/xnovk01/mkartoteka](http://www.bitbucket.org/xnovk01/mkartoteka)

Tento repozitář je rozdělen celkem do čtyř složek:

BIN	zkompilované binární soubory + potřebné knihovny a soubory.
SRC	zdrojové kódy Kartotéky i všech pomocných unit.
DOC	dokumentace (Uživatelská příručka, technický manuál, spoupis funkcí a ostatní text – tj. anotace, prohlášení, hlavička a zhodnocení).
DEV	pokusné testovací aplikace, grafické návrhy, slepá větve noSQL.



## 2. Technický popis Kartotéky

Základní otázkou při navrhování takového kartotékového programu je: Jak a kam ukládat záznamy? Určuje to totiž celý další vývoj aplikace i rozšiřujících nástrojů, multiplatformnost, vyplývající omezení, nebo třeba i možnost data načítat v cizích programech. Zvažoval jsem využití vlastního ukládacího systému a použití databází.

Nakonec jsem celou tuto verzi projektu Kartotéky postavil na využití sql databází a o druhé možnosti se tedy zmíním jen okrajově.

### 2.1. Řešení s použitím SQL

#### 2.1.1 Princip fungování Kartotéky

##### A) Databáze (na schématu 2.2 číslo 3)

Jak již bylo řečeno, základním kamenem Kartotéky je SQLite3 databáze. Databáze má následující strukturu:

```
table (tabulka)
    id (integer, ai)
    nazev (string)
    text (string)
    tabulka (text)
```

##### B) Soubory (na schématu 2.2 číslo 4)

Tato sql tabulka pochopitelně musí být uložena do vlastního souboru. Protože je dobré, když může mít uživatel v Matlafous Kartotéce více databází (kartoték), přišla na řadu otázka zda všechny databáze sloučit do jednoho souboru (a dělit je tedy dle tabulek), nebo pro každou kartotéku vytvořit vlastní soubor. Po zvážení obou možností jsem rozhodl tabulky neslučovat a pro příjemnější manipulaci ještě navíc pro každou kartotéku vytvořit vlastní složku.

Každá kartotéka má tedy svoji vlastní složku a v ní svůj databázový soubor data.db. To, oproti sloučení všech databází do jednoho velkého souboru, přináší výhody například při zálohování a zároveň se tak zmenšuje riziko poškození všech kartoték.

Veškerá data jsou ukládána do složky „files“ v kořenové složce Kartotéky.

Obrázky jsou pro jednodušší manipulaci s nimi ukládány mimo databázi, vždy ve složce jednotlivé kartotéky. Mají název id záznamu – id obrázku .jpg

##### C) Mezivrstva (na schématu 2.2 číslo 2)

Komunikaci mezi SQL databází a zbytkem Kartotéky zprostředkovává takzvaná Mezivrstva sestávající z objektu Tkartoteka a SQLite wrapperu. Podrobnější vysvětlení najdete v 2.1.3 Objekt Tkartoteka.

##### D) GUI (na schématu 2.2 číslo 1)

Uživatelské prostředí zprostředkovávající komunikaci mezi uživatelem a programem.

##### E) Programové jádro (na schématu 2.2 číslo 6)

Všechny tyto části jsou zpracovány a sjednoceny v tzv. programovém jádře. To obsahuje funkce a procedury propojující jednotlivé části tohoto schématu. Jádro například z GUI (D) dostane požadavek na otevření konkrétní kartotéky. Zjistí opravdovou lokaci databázového souboru v systému souborů (B), kterou následně předá Mezivrstvě (C). Ta jí otevře v databázi (A), a mezivrstvě (C) vrátí například počet záznamů. Ta ho pošle jádru (E), které ho předá k následnému

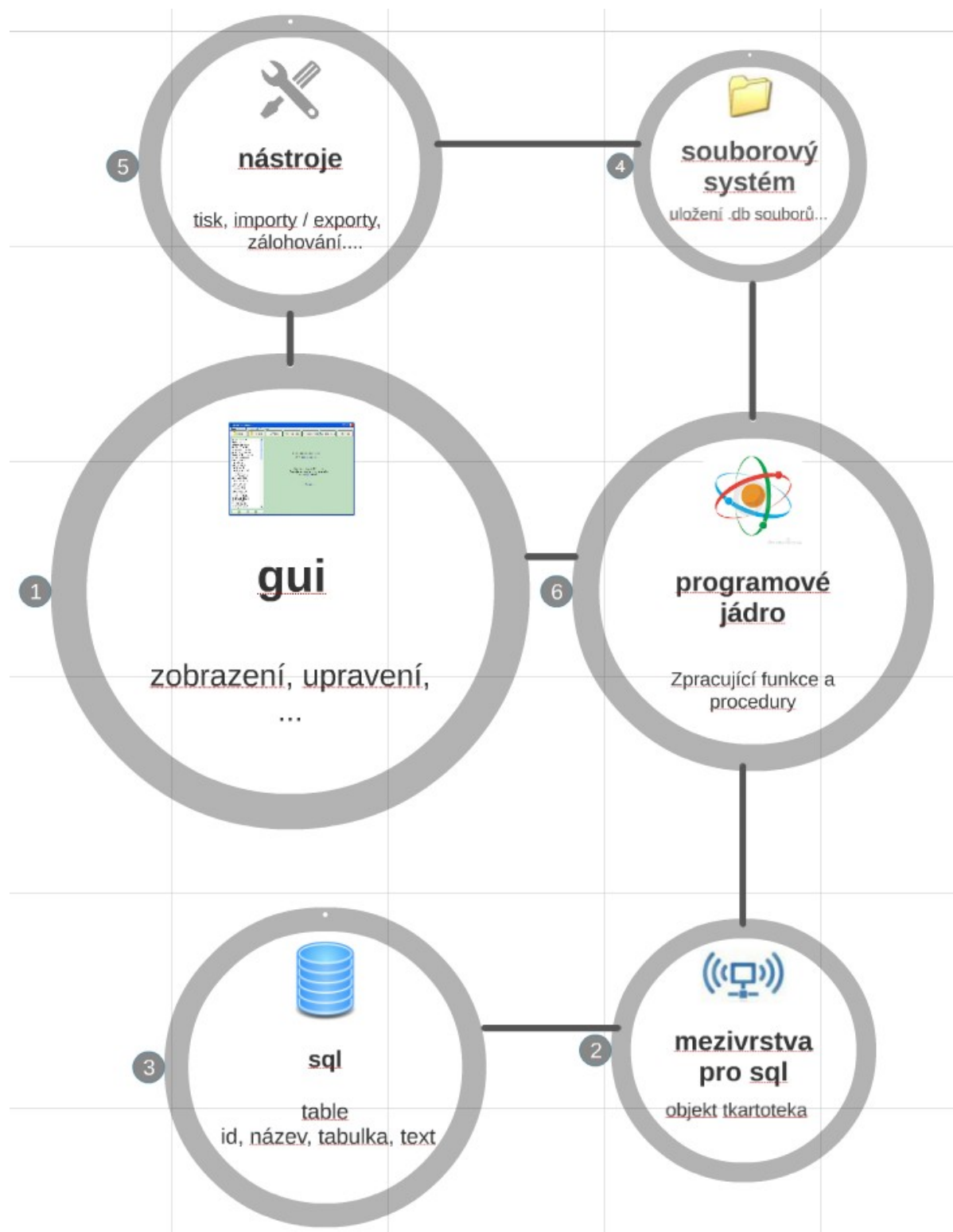


zobrazení v GUI (D).

Programové jádro sestává vždy z funkcí a procedur uvedených v první části tzv. „Unitů spojených s vlastním formulářem“ v sekci 3.1. K těmto funkcím a procedurám také náleží přiložené tabulky F1 a F2.

Jejich speciální podčástí jsou také nástroje, které někdy usnadňují jen práci s GUI, jindy vyvíjejí sofistikovanější činnost kombinující více částí Kartotéky (na schématu 2.2 číslo 5).

## 2.2 Schéma



## 2.3 Objekt Tkartoteka

Objekt v samostatném souboru KartoObj.pas. Zprostředkovává komunikaci mezi GUI Kartotéky a SQLite wrapperem. Spolou se SQLite wrapperem vytváří tzv. mezivrstvu (dle výše uvedeného schématu č. 2). Jsou v něm zapsány stěžejní sql příkazy s využitím sql parametrů. Tyto příkazy jsou dále předány SQLite wrapperu, který je však poměrně přímo předá knihovně sqlite3.dll a následně vrátí výsledky. V Unit1Form je tento objekt deklarován pod názvem Krt.

Použití tohoto univerzálního komunikačního objektu má za následek, že pokud bych chtěl v budoucnu změnit systém ukládání záznamů, bude potřeba přepsat pouze tento objekt a zbytek Kartotéky může zůstat nezměněn.

Objekt Tkartoteka obsahuje následující funkce:

```
procedure   OpenDB(InFile: String);
procedure   AttachDB(InFile: String);

function    Count: Integer;
function    FileList: String;

function    Add(InName: String): Integer;
procedure   Delete(InId: Integer);

function    GetId(InName: String): Integer;
function    GetTable(InId: Integer): String;
function    GetText(InId: Integer): String;
function    GetName(InId: Integer): String;
procedure   SetTable(InId: Integer; InTable: String);
procedure   SetText(InId: Integer; InText: String);
procedure   SetName(InId: Integer; InName: String);
function    IsName(InName: String): Boolean;
function    IsId(InId: Integer): Boolean;

function    FindInTables(InStr: String; WantIDs: boolean): String;
function    FindInTexts(InStr: String; WantIDs: boolean): String;
function    FindInNames(InStr: String; WantIDs: boolean): String;
```

## 2.2 Řešení bez použití SQL

Ze začátku nebyla Matlafous Kartotéka založena na SQL databázích ale pro svůj běh využívala souborového ukládání záznamů. Kartotéky v ní jsou, stejně jak je tomu i v SQL verzi, roztrženy do vlastních složek ale každý záznam je ukládán do samostatného souboru. Tyto soubory (s koncovkou .krt) jsou ve skutečnosti lehce zkomprimované zip soubory obsahující .txt text, .tbl tabulku a .bmp obrázky.

Toto řešení sice umožňuje jednoduše uživateli nebo jiným aplikacím načítat Kartotékové záznamy ale je velmi neefektivní při vyhledávání, hromadném exportu nebo i při načítání a ukládání.

Na druhou stranu je spekulativní, jak moc je toto zpomalení znát pokud Kartotéka neobsahuje tisíce záznamů, natožpak když je spuštěna na modernějším počítači (i na iPentiu 133MHz je tato verze Kartotéky dobře použitelná). Spustitelný soubor této verze je k dispozici na Bitbucketu ve složce DEV/noSQL, anebo jeho „nerozšířená“ verze pro veřejnost na internetu:

<http://www.matlafous.xf.cz/DOWNLOAD/mkartotzip.php>





- SevenZip

Unita pro komunikaci se 7z.dll

#### Ostatní soubory

- 7z.dll, sqlite3.dll, msimg32.dll Knihovny třetích stran
- model.db Vzorový soubor s již vytvořenou strukturou databáze
- mkupdt.exe Program pro aktualizaci Kartotéky – v parametru dostane aktuální verzi Kartotéky, z [www.matlafous.xf.cz](http://www.matlafous.xf.cz) stáhne číslo aktuální verze, porovná, pokud je k dispozici novější verze, tak nabídne její instalaci. Pokud to uživatel dovolí, ukončí běžící verzi, spustí instalaci a ukončí se.

### 3.2 Jednotlivé unity a seznam jejich procedur

Viz příloha F1 – F2 (soubor funkce.pdf).

### 4. Poznámky:

- Kartotéka = Matlafous Kartotéka = název programu
- kartotéka = jedna databáze ( souborově: .db soubor a uskupení náležejících obrázků)
- Používám raději FileListBox.Count, než Krt.Count. Ptát se totiž existující komponenty na její stav je rychlejší, než volat Krt, které následně nechá spočítat SQL počet záznamu a pak až nám vrátí hodnotu.
- Pro přehlednost bylo označení soubory (v kartotékách) změněno pro uživatele na záznamy. V komentářích zdrojových kódů se však stále může vyskytnou označení soubory.
- Namísto IDE TurboDelphi 2005 lze použít verzi Delphi 7 PE, která je též zadarmo a její instalace způsobuje mnohem méně problému. Je však potřeba změnit kódovou sadu zdrojových souborů.
- Ačkoliv to tak z uživatelské příručky nevypadá, jediným členem Matlafous Company, a tedy i autorem Kartotéky, jsem já. Název company však zůstal (z historických důvodů) a proto by psaní manuálu v jednotném čísle mohlo být pro uživatele dezorientující.

