

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

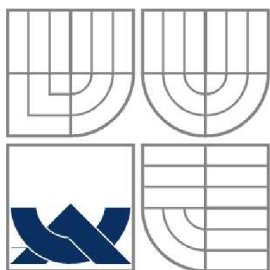
PROGRAM PRO SPRÁVU PSÍCH RODOKMENŮ

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

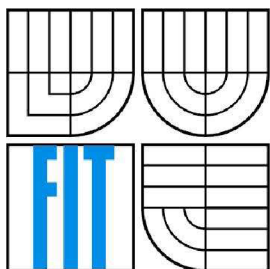
AUTOR PRÁCE  
AUTHOR

LUKÁŠ GORZOLKA

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## PROGRAM PRO SPRÁVU PSÍCH RODOKMENŮ PROGRAM FOR DOG PEDIGREE MANAGEMENT

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

LUKÁŠ GORZOLKA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. JAN KAŠTIL

BRNO 2010

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a implementací programu pro vytváření a správu psích rodokmenů v jazyce Java. Navržený program je přenositelný a je schopen provádět operace nad databází psů, umožňuje procházet rodokmeny vybraných psů a pro vybranou fenu vyhledat vhodného psa ke krytí. V závěru práce jsou zhodnoceny přínosy této aplikace a nastíněny její další možnosti rozšíření.

## **Abstract**

This bachelor's thesis deals with a project and system implementation of a program for dog pedigree management in Java. The suggested program is cross-platform and also able to realise operations with the database of dogs. It is enable to browse family trees of dogs and find out the best dog for mating a selected female dog. In the conclusion benefits of this application are evaluated. Other possibilities of its extension have also been mentioned.

## **Klíčová slova**

Rodokmen, pes, krytí, Java, XML, zpracování XML dokumentů v Javě, swing, jDOM, Uživatelské rozhraní, GUI, zobrazení psích rodokmenů

## **Keywords**

Pedigree, dog, Java, XML, jDOM, processing XML documents, GUI, graphical user interface, dog pedigree management.

## **Citace**

Gorzolka Lukáš: Program pro správu psích rodokmenů, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Program pro správu psích rodokmenů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Kaštila. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Lukáš Gorzolka  
18. května 2010

## Poděkování

Rád bych poděkoval Ing. Janu Kaštilovi, za vedení bakalářské práce, odborné konzultace a vstřícné jednání při konzultacích.

© Lukáš Gorzolka, 2010

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah .....	5
1 Úvod .....	7
1.1 Důkladnost správného krytí.....	7
2 Použité technologie.....	8
2.1 Objektově orientované programování .....	8
2.1.1 Objekty.....	8
2.1.2 Třídy.....	9
2.2 Programovací jazyk Java.....	10
2.2.1 Přenositelnost programů .....	10
2.2.2 Virtuální stroj jazyka Java (JVM).....	11
2.2.3 Java API.....	11
2.2.4 Grafické rozhraní GUI a balíček java.swing.....	12
2.3 Vyvojové prostředí NetBeans IDE a sestavovací system ANT .....	13
2.3.1 Vyvojové prostředí NetBeans IDE .....	13
2.3.2 Sestavovací system Apache ANT .....	14
2.4 Jazyk XML.....	14
2.5 Java a XML, parsování XML dokumentů pomocí balíčku JDOM .....	15
2.5.1 DOM .....	15
2.5.2 SAX .....	16
2.5.3 JDOM.....	16
3 Návrh řešení a implementace.....	17
3.1 Požadavky na program.....	17
3.2 Návrh aplikace.....	18
3.2.1 Systém ukládání záznamů a XML databáze .....	19
3.3 Implementace .....	22
3.3.1 Implementace.....	22
3.3.2 Úvodní obrazovka programu .....	22
3.3.3 Vložení nového záznamu do databáze programu.....	23
3.3.4 Editace záznamu .....	25
3.3.5 Procházení záznamu.....	25
3.3.6 Vyhledávání záznamů .....	27
3.3.7 Zobrazení detailu záznamu .....	28
3.3.8 Procházení rodokmenem.....	29
3.3.9 Rozšíření programu o možnost vkládání dalších údajů uložených záznamů.....	29

3.3.10	Implementace hledání vhodného psa ke krytí s vybranou fenou.....	31
4	Závěr.....	32
	Použitá literatura.....	35
	Příloha.....	36
	Obsah CD.....	38

# 1 Úvod

Tématem této práce je vytvoření programu pro správu psích rodokmenů. Takový program umožňuje chovateli vybudovat databázi plemen psů, což v případě hledání krycího psa pro vybranou fenu značně usnadní práci. Program byl proto navržen tak, aby mimo jiné uměl i navrhnout psa, který se svými vlastnostmi co nejvíce hodí ke krytí s vybranou fenou. V této práci se věnuji pouze plemenům pudlů, ale program lze rozšířit tak, aby byl použitelný i pro další plemena psů. Program je určen především pro chovatele, kterým záleží na kvalitě výsledných vrhů.

## 1.1 Důkladnost správného krytí

Nejdůležitějším předpokladem pro chov kvalitních vrhů, je správná volba chovné feny nebo chovného psa. Chovatelé se při výběru vhodného jedince ke krytí, jenž odpovídá standardu daného plemene, řídí jak znalostí exteriérových vlastností psa, tak genetikou.

**Například u pudlů, na které je tato práce zaměřena, se posuzují následující kritéria:**

- chrup
- skus
- barva
- ráz
- výška
- výsledky vyšetření očí (PRA, PRA-prcd)
- luxace patel
- předkové
- věk

Kvalita vybraného jedince ke krytí je podmíněna množstvím shodných bodů z výše uvedeného seznamu. Čím více faktorů má vybraný jedinec s vybranou fenou společných, tím je pro ni tento krycí pes vhodnějším partnerem. Důraz je také kladen na to, aby chovný pár neměl společného předka.

Dalším důležitým kritériem je genetika. Tato věda se zabývá problematikou dědičnosti a vrozených vad. Při výběru vhodného krycího partnera bývají sledovány genetické vady předků mnoho generací zpět.

Výběr kvalitního psa, který by se co nejvíce hodil k vybrané feně, je poněkud komplikovaný. Z tohoto důvodu se doporučuje navštívit poradce chovu, který méně zkušeným chovatelům pomůže vyhledat jedince nejvíce odpovídající požadavkům dané feny.

Myslím si, že výsledný program pro správu psích rodokmenů bude užitečným nástrojem jak pro laiky, tak i pro zkušené chovatele psů. Díky své jednoduchosti snadno a rychle vyhledá ideálního psa ke krytí a nahradí tak návštěvu poradce chovu.

## 2 Použité technologie

Tato kapitola se zabývá použitými technologiemi při tvorbě programu. Tyto technologie jsou základním kamenem celé aplikace. Jedná se o objektově orientované programování, programovací jazyk Java, značkovací jazyk XML a balíček JDOM pro zpracování XML dokumentů.

### 2.1 Objektově orientované programování

S rozšiřováním počítačů stoupají také nároky na kvalitu a komplexnost jejich programového vybavení. Moderní programy obsahují desetitisíce řádků kódu, který je nutno udržovat.

Problém nepřehledných zdrojových kódů lze řešit využitím technik procedurálního programování, které člení kód do jednotlivých procedur a funkcí. Tyto funkce jsou znovupoužitelné a případnou chybu pak stačí odstranit uvnitř jedné funkce.

I když nástup procedurálního programování byl velkým pokrokem v psaní programů, neřešilo další důležitý problém, a to sice objektové uspořádání světa. Toto znamenalo, že pokud programátor vytvořil jakýkoliv nový program, musel pak znovu vytvořit objektově orientovaný svět pomocí procedurálního programovacího jazyka. James Keogh a Mario Giannini ve své knize *OOP objektově orientované programování bez předchozích znalostí* přirovnávají tento postup k pokusu „...vsunout osmihranný kolík do kulaté dírky“. Konstatují, že i kdyby se to podařilo, nikdy to nebude sedět perfektně.[2] Východiskem z tohoto problému se stalo využívání tzv. objektově orientovaného programování, které se objevuje v osmdesátých letech. Pomocí definování tříd napodobuje objekty skutečného světa.

*„Třída se skládá z datových členů (prvků třídy), k nimž patří atributy (někdy též nazývané datové položky nebo vlastnosti) a metody. Atributy se používají k ukládání atributů objektu. Metody se někdy nazývají členské funkce a definují chování objektu.“ [2]*

#### 2.1.1 Objekty

Objekt, též instance třídy, představuje základní stavební prvek objektově orientovaného programování. Každý objekt má své atributy a metody. Atributy jsou data spojená s objektem. Metody definují chování objektu. Příkladem skutečného objektu z reálného světa je například člověk.



Takový člověk má své atributy, kterými jsou například jméno a pohlaví. V případě, že tento člověk provádí určitou činnost, má určité chování.

*„Úkolem programátora je použít objektově orientovaného jazyka k převedení atributů a chování skutečného objektu do třídy skládající se z atributů a metod, kterým počítač rozumí.“ [2]*

## 2.1.2 Třídy

*„...definice třídy se podobá vykrajovátku a nikoliv perníčku. To znamená, že definice třídy se používání při vytváření skutečné kopie třídy, kterou pak označujeme jako instanci třídy. Abychom toho dosáhli, musíme uvnitř programu deklarovat instanci třídy.“ [1]*

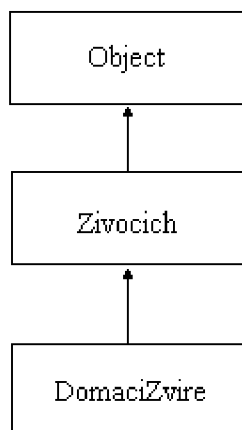
„Perníčky“ tedy v programování chápeme jako objekt, který je instancí třídy „vykrajovátko“. Každá instance třídy (objekt) má shodné atributy a metody, jako ty které jsou definovány ve třídě objektu, přičemž každý objekt má vlastní kopii atributů. [1]

Definice třídy má tři části [1]:

- **Klíčové slovo** - ve většině objektově orientovaných jazycích to bývá slovo „class“
  - informuje překladač, že se definuje třída
- **Název třídy** - vhodný název třídy zvolený programátorem
  - je unikátní pro každou třídu
- **Tělo třídy** - jedná se o část definice třídy
  - je oddělená počáteční a ukončující složenou závorkou
  - v těle třídy se definují atributy a metody třídy

Objektově orientované jazyky disponují velmi užitečnou vlastností nazývanou „dědičnost“. Pokud chceme definovat třídu, která přejímá vlastnosti již existující třídy, použijeme v její deklaraci klíčové slovo „**extends**“ následované jménem třídy, z níž se budou vlastnosti dědit. Nová třída zdědí vlastnosti původní třídy. Tuto nazýváme potomkem nebo podtřídou, případně dceřinou třídou výchozí třídy. Také říkáme, že nová třída rozšiřuje původní třídu. [3]

*„Každá třída, u níž sami explicitně neuvedeme, odkud dědí, je vždy potomkem třídy Object (z balíku java.lang, jehož jméno nemusíme uvádět). Díky tomu je přeneseně úplně každá třída potomkem třídy Object“, tj. např.: [3]*



Obr. 1 - Dědičnost – šipka směřuje od třídy potomka ke třídě předka [3]

- „Zivocich“ je bezprostředním potomkem třídy „Object“
- „DomaciZvire“ je bezprostředním potomkem třídy „Zivocich“ a potomkem třídy „Object“ v další generaci

## 2.2 Programovací jazyk Java

Programovací jazyk Java je univerzálním programovacím jazykem, který je produktem společnosti Sun Microsystems. Patří mezi objektově orientované jazyky. Tento programovací jazyk byl představen v roce 1995 a stejně jako jazyky Basic, Pascal, C/C++ se zařadil mezi tzv. třetí generaci programovacích jazyků. Jazyk Java vychází z objektové verze jazyka C, tedy C++, avšak při jeho vyvíjení se podařilo eliminovat spoustu chyb a vynechaly se složité konstrukce, jako jsou například ukazatelé, a proto bývá jazyk Java označován za jazyk „čistší“ než jeho předchůdce C++. Java díky automatické správě paměti (Garbage collector) usnadňuje programátorům práci s odstraňováním nepotřebných objektů z paměti, podporuje mechanismus vláken nebo vytvářet GUI rozhraní vyvíjených aplikací.

### 2.2.1 Přenositelnost programů

Jednou z velkých předností Javy je přenositelnost programů mezi všemi počítačovými platformami (Windows, Unix, Linux, Solaris), které obsahují virtuální stroj jazyka java (JVM). Program se po napsání jednou zkompiluje (tzv. bytekód) a na jakékoli platformě interpretuje virtuálním strojem. Nerozhoduje ani typ hardwaru, na němž program spustíme.

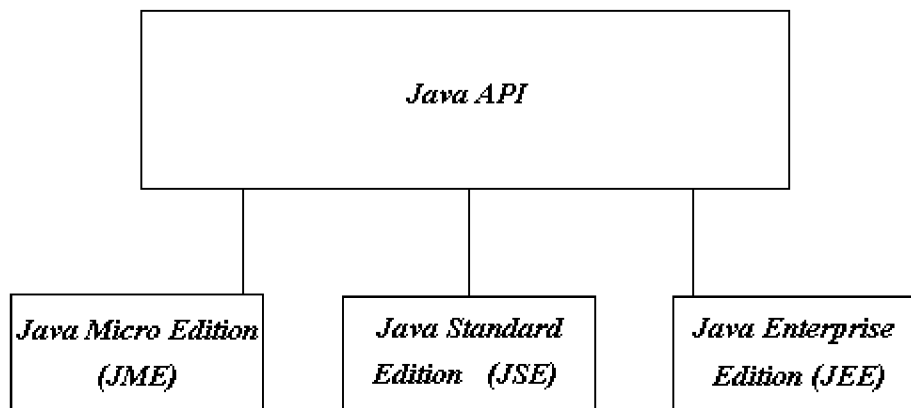
## 2.2.2 Virtuální stroj jazyka Java (JVM)

Jak už bylo zmíněno, hlavní výhodou jazyka Java je přenositelnost. Program se zkompiluje do tzv. „**Java bytekódu**“, který je dále interpretován pomocí **virtuálního stroje jazyka Java (JVM)**. Takovýto virtuální stroj jazyka Java je tvořen počítačovými programy a datovými strukturami, využívající modul virtuálního stroje ke spuštění dalších počítačových programů a skriptů vytvořených v jazyce Java. Tímto způsobem dosáhneme přenositelnosti kódu na systémy s jinou architekturou, takže narozdíl od programu, který je napsán v jazyce C nebo C++, se nemusí kód kompilovat zvlášť pro různé architektury. Důležité tedy je, že aplikace neběží přímo ve strojovém kódu počítače, na němž jsou spuštěny, ale jen díky JVM.

## 2.2.3 Java API

S JVM souvisí i další důležitá část jazyka Javy. Jedná se o Java API (Application Programming Interface), což je sada standardních knihoven, která je dodávána společně s JVM. Tyto dvě části tvoří celek, poskytovan jako **Java Runtime Environment (JRE)**.

Java API je uspořádán do balíčků (packages) a dělí se do tří částí [4]:



Obr. 2 – Uspořádání balíčků Java API

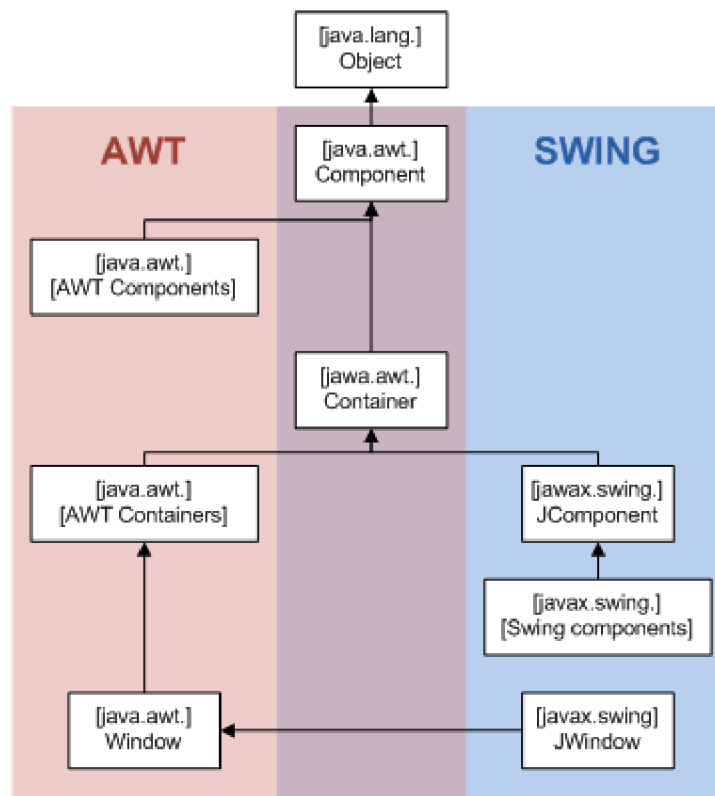
- **Java Micro Edition (JME)** – platforma pro vývoj aplikací např.: pro mobilní telefony, navigační systémy motorových vozidel, atd.
- **Java Standard Edition (JSE)** – platforma obsahující základní sadu tříd, které jsou potřebné pro tvorbu GUI a standardních aplikací
- **Java Enterprise Edition (JEE)** – platforma se sadou tříd a rozhraní pro vývoj webových aplikací (např.: Java Server Pages – JSP, Enterprise Java Beans - EJB)

## 2.2.4 Grafické rozhraní GUI a balíček java.swing

V současné době se v každém moderním programu setkáváme s grafikou. Jeden z hlavních požadavků na programy je jednoduchost, aby se s nimi uživatelé naučili pracovat co nejrychleji. Proto je důležitá interakce mezi programem a uživatelem. V takovém případě je grafické rozhraní (GUI) nejlepší volbou. Zkratka GUI pochází z anglického „graphical user interface“, což v překladu znamená grafické uživatelské rozhraní. Základní stavební prvky grafického uživatelského rozhraní jsou obvykle okna, tlačítka, popisky, přepínače a další prvky. S takovými prvky se setkáváme ve většině moderních programů, což výrazně napomáhá uživateli naučit se s programem pracovat. V jazyce Java existuje pro práci s GUI několik balíčků, jedním z nich je balíček java.swing.

*„Balíček java.swing obsahuje velkou kolekci tříd GUI, které jsou známější pod názvem Java Foundation Classes (JFC). Balíček java.swing obsahuje bohatší kolekci grafických tříd než třeba balíčky tříd Abstract Window Toolkit (AWT). Knihovny AWT a JFC se navzájem doplňují. Programátoři používají při tvorbě grafického rozhraní ve svých programech obě.“ [1]*

### 2.2.4.1 Hierarchie grafických component



Obr. 3 - Pseudo-class diagram hierarchie AWT a Swingu v Javě [4]

Jak již bylo zmíněno, každá třída v jazyce Java má společného předka, třídu „Object“. Stejně tak i třídy GUI. Ze třídy „Object“ vychází třída „Component“, která představuje všechny grafické prvky uživatelského grafického rozhraní (tlačítka, přepínače, popisky, textové pole, atd.). Další důležitou třídou je třída „Container“, což je třída, která může obsahovat i jiné grafické prvky. Například instance třídy „Container“ může obsahovat textová pole pro zadávání hodnot a tlačítka pro vyvolání akcí s těmito daty. Rozložení grafických komponent v instanci třídy „Container“ lze nastavit pomocí správce rozložení (Layout manager). Hlavní třídou knihovny Swing je třída „JComponent“, která je potomkem třídy „Container“, což tedy znamená, že všechny instance této třídy dědí vlastnosti třídy „Container“ popsané výše. Další třídy grafických komponent jako jsou například „JButton“, „JPanel“, atd. jsou potomky třídy „JComponents“.

Součástí každého grafického uživatelského rozhraní je okno, které lze minimalizovat, maximalizovat, nebo lze měnit jeho velikost podle požadavků uživatele počítače. V tomto okně je umístěn kontejner obsahu, ve kterém jsou umístěna tlačítka, textová pole a další grafické prvky tvořící grafické uživatelské rozhraní. Programátor vytvoří dané rozhraní tak, že *„nadefinuje svou vlastní třídu okna, které obsahuje definice všech ostatních prvků okna, včetně kontejneru a prvků GUI.“* [1] Ke všem změnám uvnitř okna pak dochází ve třídě okna.

## 2.3 Vyvojové prostředí NetBeans IDE a sestavovací systém ANT

V této kapitole se budu věnovat nástrojům, které jsem použil pro vývoj aplikace. Jedná se o vyvojové prostředí Netbeans IDE a sestavovací systém ANT.

### 2.3.1 Vyvojové prostředí NetBeans IDE

Vývojové prostředí NetBeans IDE (IDE – Integrated Development Environment) je Open Source projekt založený a financovaný firmou Sun Microsystems, který slouží pro vývoj a distribuci aplikací. Toto prostředí původně vzniklo na MFF UK v Praze jako školní projekt. Následně byla založena firma s názvem Netbeans, která byla zakoupena firmou Sun Microsystems. Netbeans IDE je multiplatformní vyvojové prostředí. Lze s ním pracovat na všech počítačích, na kterých funguje JDK. Jedná se o tzv. „**Java Development Kit**“, což je soubor základních nástrojů pro vývoj aplikací pro platformu Java. Vývojové prostředí je vytvořeno v jazyce Java, avšak podporuje jakýkoliv jiný programovací jazyk. NetBeans IDE poskytuje programátorovi širokou škálu nástrojů a komponent pro vývoj kvalitních multiplatformních desktopových, webových i mobilních aplikací. Pomocí plugin modulů lze toto vyvojové prostředí velmi snadno rozšiřovat o nové komponenty nebo nové programovací jazyky.

## 2.3.2 Sestavovací systém Apache ANT

Sestavovací systém ANT slouží ke kompilaci a sestavování programu napsaných především v Javě, ale existují i nástavby, které umožňují kompilovat i jiné jazyky. Tento sestavovací systém byl vytvořen v Javě, proto je plně přenositelný. Mnoho integrovaných prostředí přijalo sestavovací systém ANT jako rozšíření formou plug-in modulů nebo jako pevnou součást prostředí (Eclipse, IDEA).

Pro definici a nastavení musíme připravit sestavovací soubor `build.xml`. Pomocí XML značek v souboru `build.xml` lze určit soubory potřebné ke kompilaci a nastavit systém ANT podle potřeby. Hlavní značkou je `project`, který specifikuje projekt ke kompilaci. Z toho plyne, že v souboru `build.xml` může být i více projektů. Další důležitou značkou je `targets`, tedy cíle sestavování. Jednotlivé cíle lze vázat do závislostí. Tyto cíle mohou být například:

- **prepare** – příprava ke kompilaci
- **compile** – kompilace
- **dist** – vytvoření distribučních souborů
- **javadoc** – vytvoření dokumentace

Velikou výhodou sestavovacího systému ANT je závislost „class“ souborů. Při změně konkrétní třídy se zkompilují i třídy, které jsou na kompilované třídě závislé. ANT zvládá kromě sestavování programů i jiné operace jako je kopírování, přesouvání souborů a vytváření adresářů.

## 2.4 Jazyk XML

XML je značkovací jazyk vyvinutý skupinou XML Working Group pod záštitou konsorcia W3C. Tato speciální skupina odborníků vyvinula standard jazyka XML jakožto zjednodušenou verzi jazyka SGML. Jejím úkolem bylo kromě definování standardu také vyhledávání řešení problematiky linkování dokumentů mezi sebou a specifikace designu dokumentů napsaných v XML. Zkratka XML pochází z anglického „eXtended Markup Language“. Značkovací jazyky slouží k popisu struktury dat. Jazyk XML používá speciální značky, tzv. tagy, pomocí kterých umožňuje zachytit povahu dat. První verze tohoto značkovacího jazyka byla uveřejněna roku 1998, zatím poslední verze v roce 2000.

Značkovací jazyk XML je nezávislý na platformě a stal se jednoduchým standardem. Snaží se zachytit co nejdokonalejší popis logické struktury dat.

„XML už rovnou počítá s podporou všech možných jazyků, takže není tak úzce svázáno s angličtinou jako většina předchozích počítačových technologií. Navíc je syntaxe zápisu dokumentů v XML oproti SGML poměrně přísná, což umožní mnohem snazší a levnější vývoj aplikací, které umožňují s tímto jazykem pracovat.“ [6]

## Formální struktura XML dokumentu

Jak má být sestaven správný XML dokument určují jednoduchá pravidla. Dodržení základní struktury dokumentů je jedním z častých problémů při hromadném zpracování.

Příklad struktury XML dokumentu:

```
<clovek>
  <jmeno>Karel</jmeno>
  <pohlavi>Muž</pohlavi>
</clovek>
```

Prvek v XML dokumentu musí být uveden počáteční značkou a ukončen značkou ukončovací. V našem případě je počáteční značka `<clovek>` a ukončující značka `</clovek>`. Prvky v takovém dokumentu se mohou vnořovat, jak lze vidět v příkladu. `jmeno` a `pohlavi` jsou prvky vnořené prvku `clovek`.

## 2.5 Java a XML, parsování XML dokumentů pomocí balíčku JDOM

V této podkapitole se budu zabývat zpracováním XML dokumentů. K těmto účelům existují standardizovaná rozhraní (API) DOM a SAX. Obě rozhraní plní stejný účel, v mnohém se však velmi liší. Z těchto dvou rozhraní vychází JDOM, který byl použit při implementaci programu.

### 2.5.1 DOM

V rozhraní DOM (Document Object Model) je dokument reprezentován jako stromová hierarchická struktura, kdy každému elementu odpovídá jeden uzel stromu. Tyto uzly obsahují i komentáře, instrukce pro zpracování, atd. Tento způsob reprezentace se nazývá „grove“ (Graph Representation Of property Values). Rozhraní DOM poskytuje nástroje, díky nimž je možné procházet celý strom, modifikovat jeho uzly, mazat nebo přidávat.

*„Rozhraní DOM je standardem z dílny konsorcia W3C. Původně byl DOM vytvořen zejména proto, aby nové verze prohlížečů podporující XML používaly stejný objektový model pro přístup k dokumentu ze skriptových jazyků jako JavaScript.“ [6]*

Tento způsob je vhodný pro menší dokumenty a aplikace, které provádějí náročnější operace.

## 2.5.2 SAX

Rozhraní SAX (Simple API for XML) je založeno na zcela odlišném přístupu než DOM. Pracuje na principu řízení pomocí událostí (event-driven). Rozhraní SAX vytváří vazbu mezi událostmi, které generuje parser, a kódem který se parsuje.

*„V praxi to znamená, že si definujeme funkce, které se zavolají v okamžiku, kdy parser narazí na začátek elementu, na obsah elementu, na konec elementu, na komentář, na instrukce pro zpracování apod. Naši funkci jsou pak předány všechny potřebné parametry jako např. název elementu.“ [6]*

Hlavní výhodou tohoto přístupu je rychlost zpracování a nízká spotřeba paměti. Události jsou volány průběžně při procházení stromem. V případě, že nás zajímá konkrétní část dokumentu, nemusíme zpracovávat celý strom, ale pouze jeho určitou část. SAX má ale také velké nevýhody, např. neumožňuje upravovat dokumenty nebo neumožňuje vrátit se k již dříve zpracovaným datům.

## 2.5.3 JDOM

Dalším projektem pro zpracování XML dokumentů je Open Source projekt JDOM. Jedná se o balíček programovacího jazyka Java. JDOM vychází z obou předchozích - DOM a SAX, což mu dovoluje s nimi spolupracovat. JDOM je vyvíjen pod licencí LGPL a stane se součástí specifikace Javy.

**Hlavní strukturu JDOM tvoří 4 balíky[7]:**

- **org.jdom** - v tomto balíku jsou třídy základních entit XML (Attribute, Element, Text, ...) a také třída, která reprezentuje JDOM dokument - Document
- **org.jdom.input** - v tomto balíku jsou třídy, které vytvářejí JDOM dokumenty ze souboru nebo jiných zdrojů.
- Nejdůležitější je třída SAXBuilder, která vytváří JDOM dokument pomocí SAX parseru. Třída DOMBuilder vytváří JDOM dokument z DOM dokumentu.
- **org.jdom.output** - třídy v tomto balíku zajišťují převedení JDOM dokumentu na jiné dokumenty. Nejdůležitější třídou je XMLOutputer, který vypíše dokument v textové podobě.
- **org.jdom.transform** - třídy z tohoto balíku zajišťují XSLT transformace.



## 3 Návrh řešení a implementace

Tato kapitola se zabývá návrhem řešení a implementací programu. Jsou zde popsány požadavky na program a funkce, kterými by měl program disponovat. Další část se věnuje implementaci těchto funkcí a požadavků.

### 3.1 Požadavky na program

Jako každý program obsahující grafické uživatelské rozhraní by měl být co nejjednodušší. Pro uživatele, který s programem bude pracovat, by měl být přehledný a snadno ovladatelný. Toto bylo prvním kritériem, které jsem se snažil při tvorbě tohoto programu splnit. Proto jsem do programu vložil co nejvíce pomocných grafických prvků, které by uživateli usnadnily orientaci a práci v aplikaci. Například ikona lupy, která představuje akci prohlížení detailních informací o konkrétním jedinci v databázi psů.

Druhým kritériem bylo, že program bude schopen provádět základní operace, které mu umožní vytvořit si databázi psů a následně s ní pracovat. Program by měl disponovat možnostmi jako jsou:

- procházení záznamů v databázi
- přidávání nových záznamů
- úpravu a editaci vložených záznamů
- odstranění záznamů z databáze psů
- prohlížení detailů a fotografie psa
- řazení a filtrování záznamů podle různých kritérií
- vyhledávání záznamů psů podle vložených informací
- možnost zobrazování uživatelského manuálu
- možnost prohlížení všech potomků vybraného jedince
- možnost procházení rodokmenem vybraného psa a jeho sestavování
- vyhledání vhodných krycích psů pro vybranou fenu

V neposlední řadě je třeba myslet také na grafické zpracování uživatelského grafického rozhraní. Toto rozhraní by mělo při spuštění aplikace jasně vyjadřovat účel, za jakým byl program vytvořen a svým vzhledem působit na uživatele co nejpozitivněji, aby s tímto programem pracoval rád. Jak se mi tento záměr podařilo splnit, můžete snadno sami posoudit na screenshotech v následujících kapitolách.

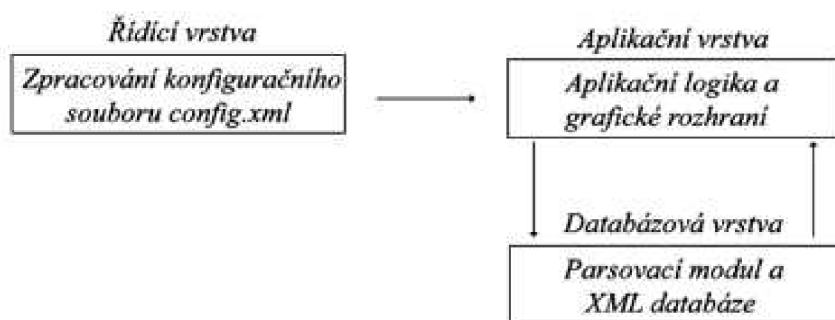
## 3.2 Návrh aplikace

Ještě před samotným návrhem programu pro správu psích rodokmenů bylo třeba určit, jaké konkrétní údaje o jednotlivých psech budou zpracovávány. Z tohoto důvodu bylo třeba pečlivě prostudovat standard pudlů podle FCI a navrhnout položky, které jsou potřebné pro všechny funkce programu. Na základě těchto údajů byla navržena datová struktura. Ta obsahuje tyto položky:

- **ID konkrétního záznamu** (uloženého psa/feny)
- **pohlaví**
- **jméno**
- **ID matky psa/feny**
- **ID otce psa/feny**
- **rok narození**
- **místo narození**
- **ráz** – ráz pudla může nabývat jednu z následujících možností: trpasličí, střední, toy, velký.
- **počet zubů** – uživatel vybírá z možností plnochrupý nebo neplnochrupý
- **skus** – skus pudla může nabývat jednu z následujících možností: nůžkový, klešťový, podkus, předkus
- **tituly** – počet titulů není omezen
- **ocenění** – počet ocenění není omezen, ocenění mohou být následující: BOB, BIS, BIG, CAC, CACIB, rCAC, rCACIB, CAJC, rCAJC
- **bonitace** – Bonitace je odborné posouzení vlastností psa/feny před jeho zařazením do chovu. Záznam o bonitaci může nabývat jednu z následujících možností: chovný/á, chovný/á s omezením, nechovný/á,
- **poměr** – Poměr pudla se uvádí jako výška / délka. V programu se výška a délka zadávají zvlášť.
- **srst** – Druh srsti pudla: kadeřavá nebo šňůrová.
- **barva** – barva pudla může nabývat jednu z následujících možností: černá, bílá, hnědá, stříbrná, modrá, červená a aprikot.
- **vyšetření PRA-prcd** – Jedná se o vyšetření očí. Možnosti: normal/clear A, normal/clear afterparents, Carrier, Affected.
- **vyšetření PRA** – může být negativní nebo pozitivní
- **vyšetření luxace patel** – vyšetření kolenního kloubu psů
- **poznámky** – poznámky uživatele programu
- **fotografie** – cesta k uložené fotografii

Všechny tyto údaje jsou programem upravitelné a lze pomocí nich provádět operace, jako jsou například vyhledávání nebo filtrování.

Celá aplikace je založena na třech vrstvách. Jedná se o řídicí, aplikační a databázovou vrstvu. První a nejvyšší vrstvou je vrstva řídicí. Tato vrstva zpracovává konfigurační soubor config.xml, ve kterém je uchována struktura XML databáze. Po zpracování konfiguračního souboru řídicí vrstvou získáme informace o počtu položek ve stromové struktuře XML databáze a jejich seznam. Ten se následně předá ke zpracování aplikační vrstvě. Aplikační vrstva spolupracuje s databázovou vrstvou pomocí parseru, kterému předává informace z řídicí vrstvy. Těmito informacemi je myšlen počet položek v XML struktuře a jejich názvy. Parser umožňuje zpřístupnit data z XML databáze a načíst je do paměti. Nad těmito daty pak aplikační vrstva provádí nejrůznější operace. Grafické rozhraní je součástí aplikační vrstvy.



Obr. 4 – Vrstvový model programu

Při tvorbě modulu parseru, který zpracovává XML databázi, jsem využil balíček JDOM. Pomocí nástrojů v něm obsažených jsem načel celý strom záznamů z XML dokumentu do paměti počítače, konkrétně do typové kolekce. Vytvoření tohoto parseru a XML databáze bylo základem pro následující tvorbu této aplikace.

### 3.2.1 Systém ukládání záznamů a XML databáze

Po návrhu všech vhodných položek, které mají být programem zpracovávány, přichází na řadu volba vhodné databáze pro tento program. Jak se již zmiňuji v přecházející podkapitole, jedná se o databázi v XML dokumentu. Z důvodu, že jeden z hlavních požadavků byla jednoduchost programu a aby tento program pro správu psích rodokmenů byl pro uživatele co nejméně náročný, jsem zvolil databázi XML. Soubor s databází se nazývá „database.xml“ a je umístěn v balíčku „xmlFiles“.

### 3.2.1.1 Popis struktury XML databáze

Struktura XML databáze je definována v konfiguračním souboru „config.xml“. Základním elementem databáze je element <root>, kterému jsou všechny další záznamy vnořené. Každý do programu vložený záznam, tím se myslí záznam o konkrétním psovi nebo feně, je v databázi definován jako element <zaznam>, který má dva atributy. Jedním z nich je „id“, což je jednoznačný identifikátor tohoto záznamu. Dalším atributem tohoto záznamu je pohlaví. 0 pro tento atribut znamená, že se jedná o psa, 1 znamená fenu. Ostatní údaje jsou v záznamu uloženy jako vnořené prvky.

### 3.2.1.2 Struktura XML databáze

```
<root>
  <zaznam id="1" pohl="0">
    <jmeno></jmeno>
    <matka></matka>
    <otec></otec>
    <narozeni></narozeni>
    <misto_narozeni></misto_narozeni>
    <raz></raz>
    <pocet_zubu></pocet_zubu>
    <skus></skus>
    <tituly></tituly>
    <oceneni></oceneni>
    <bonitace></bonitace>
    <pomer></pomer>
    <srst></srst>
    <barva></barva>
    <poznámka></poznámka>
    <foto></foto>
    <praprčd></praprčd>
    <pra></pra>
    <luxpa></luxpa>
  </zaznam>
</root>
```

Hodnoty se v databázi ukládají s ohledem na možnost budoucího elektronického zpracování. To znamená, že ne všechny položky se ukládají jako textové řetězce, ale pouze číselným označením. A to následovně:

<matka>, <otec> ukládají se pouze identifikátory záznamů, to znamená jejich ID.

<raz> 0 – trpasličí, 1 - střední , 2 - toy, 3 - velký

<skus> 0 – nůžkový, 1 – klešťový, 2 - podkus, 3 - předkus

<srst> 0 – kadeřavá, 1 - šňurová

<barva> 0 - černá, 1 - bílá, 2 - hnědá, 3 - stříbrná, 4 - červená, 5 – aprikot, 6 – modrá,  
7 - krémová

<bonitace> 0 – chovný/á, 1 - chovný/á s omezením, 2 – nechovný/á

<praprcd> 0 - normal/clear A, 1- normal/clear afterparents, 2 - Carrier, 3 – Affected

<pra> 0 – negativní, 1 - pozitivní

<foto> ukládá se číslo fotografie, která je umístěna v adresáři fotografií

### 3.2.1.3 Ukládání titulů

Počet uložených titulu není omezen, tituly se ukládají s atributem „typ“, který říká, o jaký titul se jedná. Dále se uchovává informace jaký stát titul zadal.

```
<tituly>
  <titul typ="Šampion">Česká republika</titul>
  <titul typ="Junior Šampion">Česká republika</titul>
</tituly>
```

### 3.2.1.4 Ukládání ocenění

Stejně jak je to u titulů, ani počet uložených ocenění není omezen. Každý typ ocenění má svou značku stejného jména jako je jméno ocenění. Dále se uchovávají i státy, které ocenění udělily.

```
<oceneni>
  <BOB>CZ</BOB>
  <rCACIB>CZ</rCACIB>
</oceneni>
```

V příloze je uvedena ukázka uložení konkrétních záznamů v databázi.

## 3.3 Implementace

Tato kapitola popisuje implementaci programu a uživatelského rozhraní. Jsou zde uvedeny screenshoty z programu a popsány všechny funkce aplikace. Kapitola je rozdělena do podkapitol a každá z nich popisuje konkrétní operace, tzn. přidávání záznamů, editaci záznamů, procházení rodokmeny, atd.

### 3.3.1 Implementace

Uživatelské rozhraní této aplikace je rozděleno do dvou částí. Jedná se o panel s uživatelským menu a obsahový panel, umístěn v pravé části okna. Tento panel mění svůj obsah podle zvolené operace, kterou se uživatel rozhodl provádět. Program po vyhodnocení akce, kterou provedl uživatel pomocí nabízených možností programu, připojí do obsahové části panel s implementací vybrané funkce. Každý takový panel je kontejnerem, který obsahuje další prvky grafického rozhraní. Těchto panelů je celkem 6. První z nich je panel s úvodním obrázkem. Tento panel neimplementuje žádnou funkci, existuje pouze z estetických důvodů. Dalšími panely jsou panel pro vložení nového záznamu do databáze nebo úpravu již vloženého záznamu, panel pro výpis všech záznamů, panel pro vyhledávání, panel pro procházení rodokmenem, panel pro zobrazení detailů. V následujících podkapitolách se budeme podrobně zabývat každým panelem zvlášť.

Po spuštění programu a načtení struktury XML databáze, program zahájí zpracování dat z databáze pomocí parseru. Tento modul je definován ve třídě *xml.Parser.java*. Pomocí balíčku JDOM načte celý strom záznamů do paměti počítače a všechny záznamy uloží do typové kolekce definované ve třídě *java.util.ArrayList*. Pro každý záznam je vytvořen nový objekt definován ve třídě *xml.Zaznam.java*. Celá databáze se tedy uloží jako kolekce objektů. S touto kolekcí pak program pracuje po celou dobu spuštění. Operace, jako je přidávání nových položek nebo mazání stávajících, se v XML databázi projeví až při ukončení programu, kdy se celá kolekce záznamů, v případě změny modifikovaná kolekce záznamů, uloží zpět do XML souboru.

### 3.3.2 Úvodní obrazovka programu

Program byl vytvořen v programovacím jazyku Java a uživatelské rozhraní pomocí balíčku *java.swing*. V následujícím obrázku (obr. 5) je zachycena úvodní obrazovka programu, tak jak se zobrazí po spuštění. V levé části okna se nachází uživatelské menu, které nabízí 5 možností. První možností je přidání nového záznamu. Tím je myšleno vložení všech potřebných informací o psovi nebo feně do databáze. Další možností je procházení všech záznamů v databázi. Zde jsou k dispozici nástroje pro filtrování a řazení záznamů. Další položkou uživatelského menu je „Najít záznam“. Tato

položka umožňuje vyhledávání jedince podle zadaných kritérií. Nezbytnou součástí aplikace je uživatelský manuál ve formátu PDF rovněž přístupný z uživatelského menu. Poslední položkou menu je ukončení programu. Během ukončování programu dochází k automatickému uložení všech změn, které uživatel během práce s programem učinil.

V levé části okna, pod uživatelským menu je navíc malý informační panel, který informuje o tom, kolik psů a fen se v databázi vyskytuje.



Obr.5 – Úvodní okno programu

### 3.3.3 Vložení nového záznamu do databáze programu

Na následujícím obrázku (obr. 6) je zobrazen panel pro přidávání nových záznamů. Tento panel je definován ve třídě *rodokmen.PanelPridatZaznam.java* a umožňuje vložit nový záznam do databáze programu. Pomocí nabízených nástrojů je možné nastavit všechny potřebné charakteristické vlastnosti, dosažené tituly, ocenění a fotografii. Při vkládání nového záznamu je jediným povinným parametrem jeho jméno. Vytvořenému záznamu se navíc vygeneruje unikátní identifikátor (ID), který není uživateli přístupný. Pomocí tohoto ID je možné provádět operace jako je například sestavování rodokmenu. Vkládané jméno musí být taktéž unikátní, program neumožňuje duplicitu jmen. Pro

výběr matky a otce vkládaného jedince, se používají nabízené roletky s nabídkou jmen. Nastavení rodičů tímto způsobem je podmínkou pro procházení rodokmenem. Program umožňuje vložit psa nebo fenu bez nastavení matky a otce. Je to důležité z důvodu, kdy vkládáme jedince, jehož předci nejsou uloženi v databázi programu. Tito mohou být vloženi později.

Obr. 6 – Vkládání nového záznamu

Panel poskytuje nástroj pro výběr fotografie umístěné na pevném disku počítače nebo jiném externím médiu připojeném k počítači. Po načtení fotografie následuje její velikostní úprava. Fotografie se uloží ve třech velikostech. První velikost je miniatura v rozlišení 80x60 bodů. Druhou velikostí je obrázek v rozlišení 250x188 bodů. Obrázek v této velikosti se používá při zobrazení detailu záznamu. Nakonec se uloží obrázek v původním rozlišení, pro pozdější prohlížení.

Na předchozím obrázku je možné spatřit položky s názvem „Rozšířená položka 1 až 4“. V kapitole 3.2.2 Rozšíření programu o možnost vkládání dalších údajů uložených záznamů, bylo zmíněno, že úpravou konfiguračního souboru config.xml je možné program rozšířit o další položky. Takto se tato změna projeví v panelu přidávání. S obsahem těchto položek se pracuje jako s textovým řetězcem, do databáze se uloží ve stejném tvaru jako text.



Před samotným uložením zadaných informací program automaticky zkontroluje správnost vložených údajů a případnou duplicitu jména jedince. V případě vložení nekorektních údajů program na tuto skutečnost upozorní zprávou v dialogovém okně.

### 3.3.4 Editace záznamu





Obr. 7 - Editace záznamu

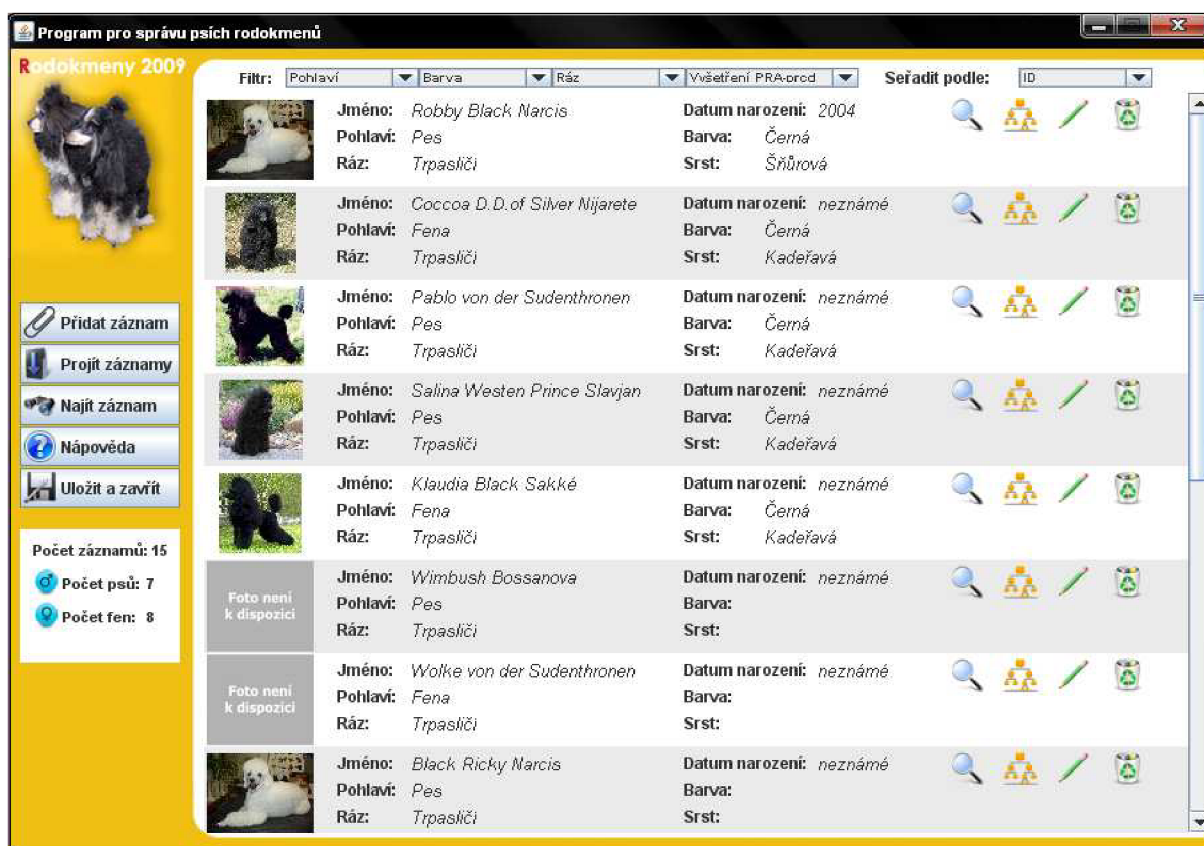
Na obrázku č. 7 je panel pro přidávání nového záznamu v módu pro editaci. Panel se zobrazí již vyplněn a umožní snadnou úpravu záznamu. Před uložením provedených změn dochází k opětovné kontrole správnosti zadaných údajů, avšak duplicita jména se zde řeší pouze v případě změny jména.

### 3.3.5 Procházení záznamu

Při volbě možnosti procházení záznamů, nacházejícím se rovněž v uživatelském menu, se do obsahové části připojí panel, který umožňuje zobrazit seznam všech záznamů. Panel je definován ve třídě *rodokmen.PanelSeznamZaznamu.java*. Každý záznam je reprezentován panelem, který obsahuje miniaturu fotografie, některé údaje (jméno, pohlaví, rás, datum narození, barva, srst) a nabídku operací, které je možné použít na konkrétní záznam. Definice těchto panelů se nachází ve třídě *rodokmen.PanelZaznam.java*.

K dispozici je nabídka těchto operací:

-  **Zobrazení detailu záznamu** – otevře panel pro prohlížení detailů záznamu
-  **Zobrazení rodokmenu** – zobrazí panel pomocí něhož lze procházet rodokmenem
-  **Editace záznamu** – možnost úpravy záznamu
-  **Smazání záznamu z databáze**



Obr. 8 – Procházení záznamů

Panel pro procházení záznamů je navíc vybaven nástroji pro řazení a filtrování záznamů, což umožňuje snadnější vyhledání potřebného záznamu. Řadit lze podle ID, což je unikátní identifikátor záznamů. Záznamy seřazené podle ID po sobě následují v pořadí, v jakém byly do programu postupně zadávány. Další možností řazení je chronologicky podle jména.

K dispozici je filtrování podle pohlaví, barvy, rásy a vyšetření PRA-pred. Pokud ve filtru PRA-pred zvolíme konkrétní volbu pro filtrování, zobrazí se záznamy, které odpovídají zadané volbě, ale i ty, které mají vyšetření lepší než zadané ve filtru. Implementace panelu s nabídkou filtrů a řazení se nachází ve třídě `rodokmen.PanelListaFiltru.java`.

### 3.3.6 Vyhledávání záznamů

Program pro správu psích rodokmenů

Rodokmeny 2009

**Zadejte údaje k vyhledávání:**

Jméno  
Pohlaví  
Ráz  
Matka  
Otec  
Rok narození  
Místo narození  
Výška/Délka (cm) Tol:  
Počet zubů  
Vyšetření PRA  
Vyšetření PRA-prcd  
Vyšetření lux. pately  
Skus  
Bonitace  
Barva srsti  
Druh srsti  
Rozšířená položka 1  
Rozšířená položka 2  
Rozšířená položka 3

Ocenění:  
 BOB  BIS  BIG  CAC  CACIB  rCAC  
 rCACIB  CAJC  rCAJC

Tituly:  
Zadejte hledaný titul

Rozšířená položka 4

Obr. 9 – Zadávání údajů k vyhledávání

Vyhledávat je možné nejen pomocí pevně daných položek, ale i položek, o které je možno program rozšířit. Tyto položky jsou na obrázku označeny jako „Rozšířené položky 1 až 4“. Více o tomto rozšíření v kapitole 3.3.9.

U vyhledávání podle výšky nebo délky je možné zadat toleranci pomocí textového pole „Tol“, která se bere v úvahu při vyhledávání.

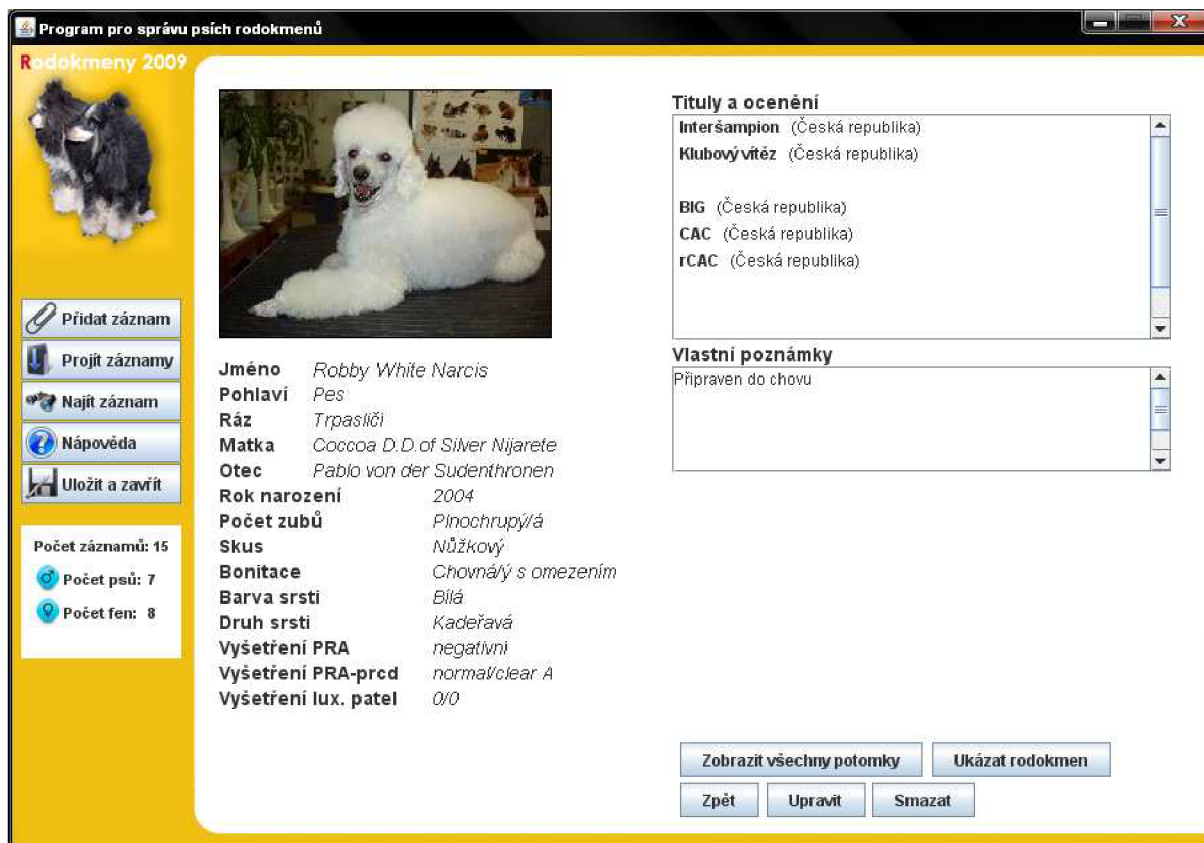
Vyhledávání podle vyšetření PRA-prcd funguje stejně jako filtrování v panelu výpisu (kapitola 3.3.5). Ve výsledku se zobrazí jedinci s výsledky vyšetření PRA-prcd stejnými jako je zadaná volba ve vyhledávání, ale i lepšími.

Vyhledávací panel poskytuje nástroje, které vyhledávají jedince podle ocenění a titulů. Po zadání požadovaných kritérií se ve výsledku zobrazí záznamy, které obsahují alespoň jeden z žádaných titulů.

Implementace panelu s nabídkou vyhledávání se nachází ve třídě *rodokmen.PanelHledani.java*, samotný vyhledávací algoritmus se nachází v balíčku *search* ve třídě *search.vyhledavani.java*.

Výsledky vyhledávání se zobrazí jako seznam položek tak, jak je to popsáno v předcházející kapitole, s možností dalšího filtrování a řazení.

### 3.3.7 Zobrazení detailu záznamu



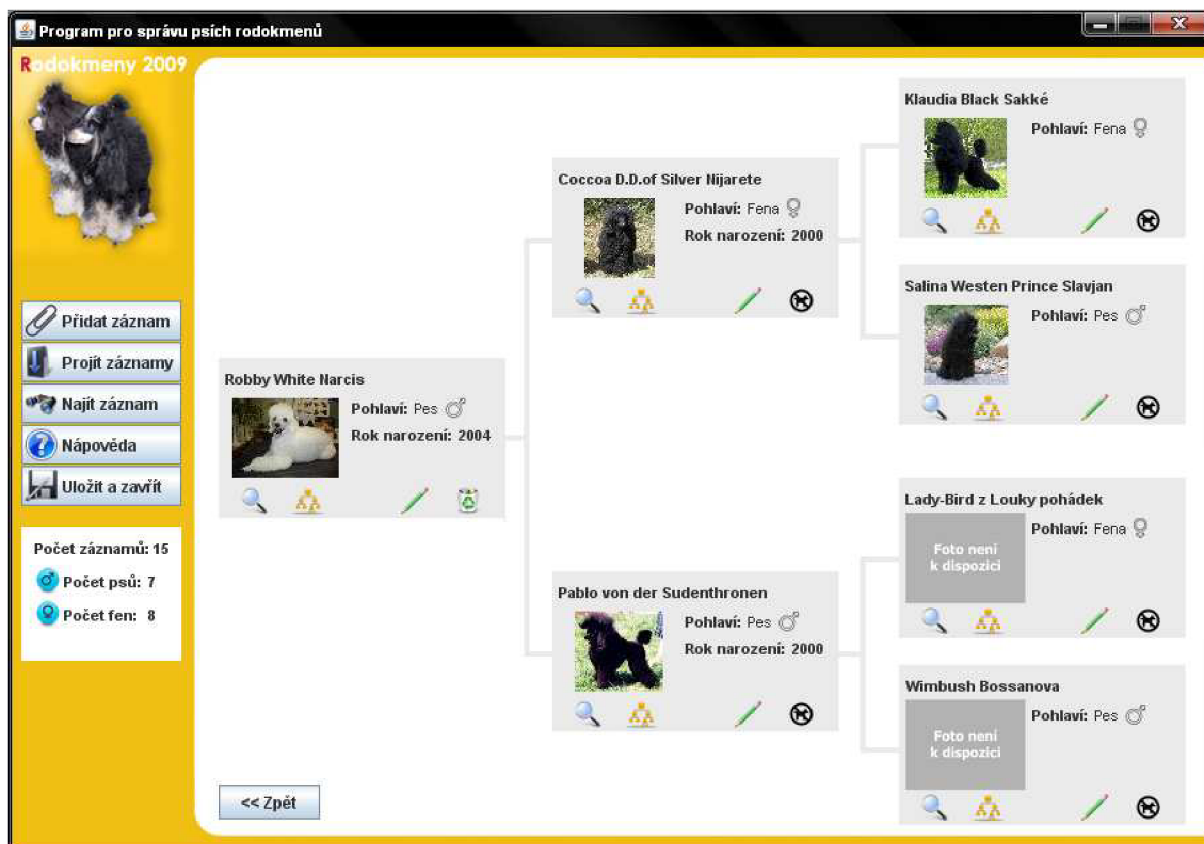
Obr. 10 – Zobrazení detailu záznamu

Obrázku č. 10 poskytuje náhled zobrazení detailu záznamu. Nástroje tohoto panelu umožňují zobrazení všech informací o vybraném jedinci, přehled jeho titulů, ocenění a nabízejí vyvolání dalších akcí jako jsou například: zobrazení všech potomků, zobrazení rodokmenu, editace záznamu nebo smazání. V případě, že prohlížený jedinec je fena, je nabízena navíc možnost vyhledat vhodné krytí pro tuto fenu.

V kapitole 3.3.3 bylo zmíněno, že při vkládání fotografie se vytvoří její tři kopie. První z nich je použita jako miniatura při procházení záznamů, nebo při zobrazení rodokmenů, další kopie se používá zde při zobrazení detailu (obr. 10). A poslední kopie zůstává v originálním rozlišení. Po kliknutí na obrázek se zobrazí okno s fotografií v plné velikosti. To je vhodné například při posuzování určitých exteriérových vlastností psa nebo feny. Je to však podmíněno tím, že uživatel při vkládání záznamu do programu vložil fotografii s velkým rozlišením.

Panel pro zobrazení detailů je implementován ve třídě `rodokmen.PanelUkazDetail.java`.

### 3.3.8 Procházení rodokmenem

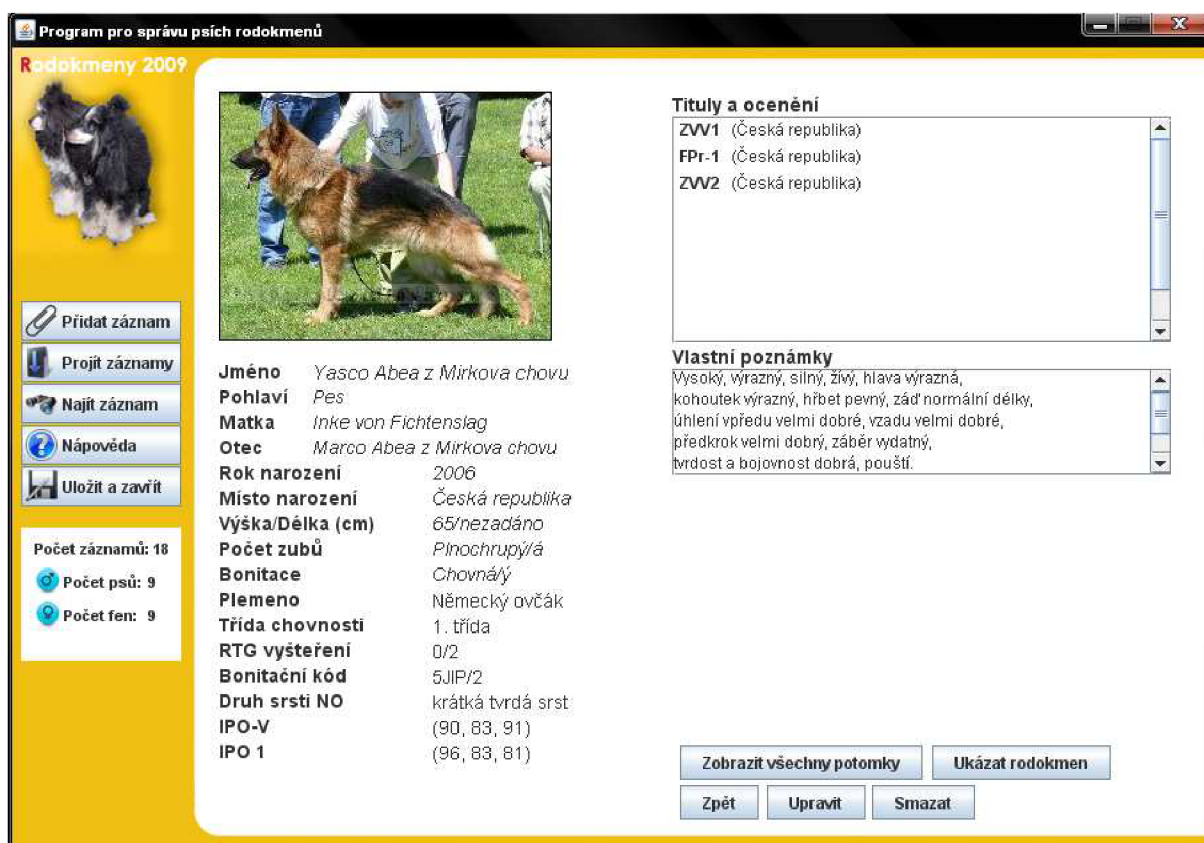


Obr. 11 – Panel s rodokmenem

Jednou z hlavních funkcí programu je procházení rodokmenem vybraných psů a fen. K tomuto účelu slouží panel definovaný ve třídě *rodokmen.PanelRodokmen.java*. Náhled je zobrazen na obrázku výše (obr. 11). Panel pro zobrazení rodokmenu poskytuje nástroje pro procházení rodokmenem oběma směry, tzn. směrem k předkům a zase se vrátit zpět (směrem k potomkům). Panel disponuje možností upravovat větve rodokmenu, upravovat záznamy psů v rodokmenu, nebo prohlížet jejich detaily. Panel pro zobrazování rodokmenů se skládá z menších panelů, ve kterém se zobrazují jednotliví jedinci z rodokmenové linie.

### 3.3.9 Rozšíření programu o možnost vkládání dalších údajů uložených záznamů

Program uchovává a zpracovává údaje, které byly v návrhu zvoleny jako nejvhodnější položky, jako jsou např. jméno, pohlaví, barva, rás, atd... Tyto jsou v programu pevně přednastaveny. V případě, kdy by bylo potřeba program rozšířit o další položky, lze toho dosáhnout změnou v konfiguračním souboru „config.xml“. Tyto rozšířené položky poté program zpracovává jako textový řetězec a je možné podle nich provádět vyhledávání.



Obr. 12 – Ukázka možnosti rozšíření pro jiná plemena

Přidání položky lze tedy provést úpravou konfiguračního souboru. Každá rozšířená položka musí být vložena až za stávající položky obsažené v konfiguračním souboru a musí mít tento tvar:

```
<nazev_tagu popis="Jméno položky"></nazev_tagu>
napr: <PB popis="Původní barva"></PB>
```

Jak je z příkladu zřejmé, atribut `popis` se v programu projeví jako popis k rozšířené položce. Jeho délka je omezená na 22 znaků, v případě že bude delší, automaticky se v programu zkrátí.

Myšlenka rozšiřování o další položky programu vznikla z důvodu potřeby uživatelů uvádět v databázi další informace pro ně nezbytné. Pokud by se uživatel programu nespokojil s uložením této informace jako poznámky a potřeboval by podle této nové informace vyhledávat záznamy, je nutné využít možnosti rozšíření programu o možnost vkládání dalších údajů uložených záznamů.

Tímto je možné docílit i toho, že program je aplikovatelný i na jiná plemena psů. Stačí jen program rozšířit o položky, které jsou pro toto plemeno charakteristické.

### 3.3.10 Implementace hledání vhodného psa ke krytí s vybranou fenou

Algoritmus hledání vhodného krycího psa je založen na srovnávání exteriérových vlastností a výsledků zdravotních vyšetření obou jedinců. Tyto faktory jsou popsány v úvodu této práce. Podle výsledků srovnání se nalezení jedinci řadí za sebou podle vhodnosti ke krytí. Nejvhodnější psi, kteří se svými vlastnostmi co nejvíce hodí, bývají označováni jako „Excelentní krycí pes“, „Výborný krycí pes“.

Program tedy prohledává svou databázi a každého psa podrobí srovnání vlastností s fenou, pro které se vyhledávání provádí. Nalezení psi se zobrazí v programu seřazení od nejvhodnějšího psa po nejméně vhodný. Tedy seznam psů začíná Excelentním krycím psem.

#### **Excelentní krycí pes musí mít tyto vlastnosti:**

- musí být plnochrupý
- musí mít přiměřený věk
- skus musí být nůžkový
- vyšetření PRA: negativní
- vyšetření PRA-prcd: normal/clear A
- luxace patel: 0/0
- výška stejný jako u feny
- ráz stejný jako u feny
- barva stejná jako barva feny
- pes nesmí mít s fenou žádného společného předka v první a druhé generaci.

Pokud má pes s vybranou fenou v první nebo druhé generaci společného předka, nejedná se o „Excelentního krycího psa“, nýbrž „pouze“ „Výborného krycího psa“.

Akci pro vyhledání vhodného psa pro vybranou fenu je možné vyvolat pomocí nástrojů v panelu pro zobrazení detailů psa. Tento panel je popsán v kapitole 3.3.7 Zobrazení detailů záznamů.

## 4 Závěr

Cílem této práce bylo vytvořit program, který by sloužil chovatelům pudlů a umožnil jim vytvořit si databázi psů, se kterou by pak mohli dále pracovat. Myslím si, že tento záměr se mi podařilo splnit. Program je jednoduchý a nabízí spoustu funkcí, které by mohl chovatel potřebovat. Grafické rozhraní aplikace je velmi jednoduché, což je přínosné i pro uživatele, kteří nemají s počítači tolik zkušeností. Databáze programu je vytvořena pomocí XML dokumentu a uživatel se o ní tedy nemusí vůbec starat. Vše zařídí program za něj, což opět potvrzuje to, že program je uživatelsky nenáročný. V případě, že by databáze byla tvořena jinou technologií než tou, která byla využita v této práci, například MySQL, bylo by pro zajištění fungování nutné použít další externí programy, což by možná bylo efektivnější, ale za cenu zvýšení požadavků na uživatele, čímž by se jejich okruh zmenšil. Můj prvořadý cíl, tedy jednoduchost programu, byl po všech stránkách splněn.

Pro uživatele přínosná může být i možnost rozšíření programu o další pro něj důležité položky, což program činí efektivnějším. Tato rozšíření byla implementována s ohledem na snadnost, to znamená že změnu lze provést na jednom místě (v konfiguračním souboru) a program se této změně přizpůsobí ve všech sekcích. Původně využívání tohoto rozšíření mělo být výsadou pouze jakéhosi správce (odborníka), na kterého by se musel uživatel programu odkázat, avšak ve výsledku je rozšiřování programu tímto způsobem tak jednoduché, že to zvládne i běžný uživatel. Na druhou stranu tato výsada spolu přináší i požadavek na zodpovědnost a pečlivost uživatele při práci s těmito rozšířeními. V případě nekorektní změny konfiguračního souboru, může dojít k selhání programu.

Hlavní funkcí programu je procházení rodokmenů vybraných jedinců. Rodokmeny si uživatel v programu vytváří sám a rozsah rodokmenů a přesnost informací v něm obsažených závisí pouze na něm. Program bude pracovat s těmi informacemi, které si do něj uživatel sám vloží. Tyto informace jsou důležité taktéž pro vyhledání vhodného psa ke krytí, což je dalším posláním programu. Funkčnost a spolehlivost aplikace tedy závisí na uživatelem zadaných údajích. V případě špatných výsledků po zadání nekorektních informací, nenese tvůrce programu, tedy já, žádnou zodpovědnost a proto hraje velkou roli rozvážnost a informovanost uživatele.

Při tvorbě programu jsem počítal s jeho budoucím rozšiřováním. Jedním s možných směrů práce je například zdokonalení vyhledávacích algoritmů. Na tuto možnost je program připraven. Už samotný fakt, že je aplikace vytvořena v objektově orientovaném jazyce, naznačuje možnosti snadného rozšiřování. Kód je členěn do tříd, a proto při případných rozšířeních stačí aktualizovat konkrétní třídu. V balíčku „search“ je připravena třída s vyhledávacími algoritmy, kterou stačí upravit podle požadavků. Vyhledávací algoritmy by bylo možné například upravit tak, aby pracovaly s ohledem na genetické informace vybraných jedinců. Přínosem pro uživatele by byla také možnost tisku, kdy by bylo možné vytisknout informace o vybraném jedinci nebo dokonce jeho rodokmen. V úvahu přichází také možnost exportu například do formátu PDF.



Tato práce mi byla přínosem hlavně ve zdokonalení mých programátorských schopností a lepší orientaci v jazyce Java. Naučil jsem se různé techniky zpracování XML dokumentů, čehož můžu využít i v jiných programovacích jazycích, například PHP. V průběhu vývoje programu jsem si potvrdil fakt, jak je důležité vytvořit správné základy a počítat s budoucím rozšířením. U menších programů to není na první pohled znát, ale u větších projektu jako je tento znamenají případná rozšíření velké komplikace pokud je program napsán špatně. Proto je nutné na tuto skutečnost brát ohled již v prvních fázích vývoje. Doufám že tento program bude užitečný mnoha chovatelům a usnadní práci při vybírání vhodných jedinců k chovu. V případě, že dojde ke jeho dalším modifikacím, budu jedině rád a těším se na jeho budoucí verze. Jelikož jsem sám chovatelem psů, není tato verze pro mne konečnou a plánuji její vylepšení ve svůj vlastní prospěch.

# Použitá literatura

- [1] James Keogh, *Java bez předchozích znalostí*, CP Books a.s, Brno, 2005. ISBN 80-251-0839-2
- [2] James Keogh, Mario Giannini: *OOP objektivě orientované programování bez předchozích znalostí*. Computer Press, a.s, Brno, 2006. ISBN 80-251-0973-9
- [3] Tomáš Pitner: *JAVA – ZAČÍNÁME PROGRAMOVAT, podrobný průvodce začínajícího uživatele*, Grada Publishing, spol s.r.o, Praha, 2002, ISBN 80-247-0295-9
- [4] [WWW.PROGRAMUJTE.COM](http://www.programujte.com) Úvod do programovacího jazyka Java [online], dostupné z WWW na <http://programujte.com/?akce=clanek&cl=2006041804-uvod-do-programovacieho-jazyka-java>
- [5] Jakub Závěrka: *Pseudo-class diagram hierarchie AWT a Swingu v Javě* [online], dostupné z WWW na <http://cs.wikipedia.org/wiki/Soubor:AWTSwingClassHierarchy.png>
- [6] Jiří Kosek: *XML pro každého*, Grada publishing, 2000, ISBN 80-7169-860-1
- [7] Vojtěch Patrný: článek na [www.linuxzone.cz](http://www.linuxzone.cz) *Programově na XML, 3.díl - JDOM* [online], dostupné z WWW na <http://www.linuxzone.cz/index.phtml?ids=2&idc=377>

# Příloha

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <zaznam id="1" pohl="1">
    <jmeno>Robby White Narcis</jmeno>
    <matka>2</matka>
    <otec>3</otec>
    <narozeni>2004</narozeni>
    <raz>0</raz>
    <pocet_zubu>0</pocet_zubu>
    <skus>0</skus>
    <bonitace>1</bonitace>
    <pomer>0/0</pomer>
    <srst>0</srst>
    <barva>1</barva>
    <praprcd>0</praprcd>
    <pra>0</pra>
    <luxpat>0/0</luxpat>
    <poznamka>Připraven do chovu</poznamka>
    <foto>1</foto>
    <oceneni>
      <BIG>Česká republika</BIG>
      <CAC>Česká republika</CAC>
      <rCAC>Česká republika</rCAC>
    </oceneni>
    <tituly>
      <titul typ="Interšampion">Česká republika</titul>
      <titul typ="Klubový vítěz">Česká republika</titul>
    </tituly>
  </zaznam>
  <zaznam id="2" pohl="0">
    <jmeno>Coccoa D.D.of Silver Nijarete</jmeno>
    <matka>6</matka>
    <otec>5</otec>
    <narozeni>2000</narozeni>
```

```
<raz>0</raz>
<skus>1</skus>
<bonitace>0</bonitace>
<pomer>0/0</pomer>
<srst>0</srst>
<barva>0</barva>
<praprcd>0</praprcd>
<luxpat>1/0</luxpat>
<poznamka>Připravena do chovu</poznamka>
<foto>2</foto>
<oceneni>
  <rCACIB>CZ</rCACIB>
  <CAC>CZ</CAC>
  <rCAC>CZ</rCAC>
</oceneni>
<tituly>
  <titul typ="Junior šampion">Česká republika</titul>
  <titul typ="Čekatel klub. šampiona">Česká
    republika</titul>
  <titul typ="Klubový vítěz" />
</tituly>
</zaznam>
</root>
```

# Obsah CD

- Zdrojové soubory programu
- Spustitelný program
- Programová dokumentace programu
- Zdrojové soubory textové zprávy
- Výsledná textová zpráva

