

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
katedra informatiky a výpočetní techniky

ZÁPOČTOVÁ PRÁCE z UIR

Genealogický expertní systém

17.05.2014

Antonín Neumann
skupina Po 12:05 – 13:45

1 Zadání

Vytvořte genealogický expertní systém. Program bude umět odpovídat na otázky:

- Kdo je můj pokrevní příbuzný?
- Kolik má moje prateta sestřenic?
- A další podobné.

1.1 Formát vstupních dat

Vstupem bude informace databáze základních informací o osobě ve tvaru: *[id, jméno, matka, otec, pohlaví, partner]*.

Tento formát jsem si upravil do podle mého praktičtější podoby. Jako separátor byl použit znak pro středník, protože čárka se může za určitých okolností vyskytovat i uvnitř položky, například typu u příjmení.

Jméno bylo rozděleno na dvě hodnoty, křestní jméno a příjmení. Pokud některá reference na jinou osobu neexistuje, uvede se místo jejího *id* klíčové slovo *null*. Změnou prošlo i pořadí prvků, a to zejména z důvodu přehlednosti, nový vstupní řádek s údaji o osobě vypadá následovně:

[id; jmeno; prijmeni; pohlavi; id_matka; id_otec; id_partner].

Seznam osob je uveden v textovém souboru, který je zadán jako vstupní soubor při spuštění programu. Každá osoba je v tomto souboru uvedena na jedné řádce.

2 Analýza

Rodokmen, který potřebujeme z dat vytvořit, je nejvhodnější reprezentovat grafovou strukturou. Konkrétně se jedná o datovou strukturu strom, kde má každý uzel (osoba) *n* potomků a dva rodiče.

Každá osoba musí mít minimálně jednoho rodiče, matku, pokud tato položka není uvedena, použije se jako matka tzv. „první předek“. Tento první předek je uměle vytvořená osoba, která je na vrcholu rodinného stromu a má *id* nastaveno na hodnotu 0.

Dále jsou potřeba pravidla, která by umožnila odpovídat na konkrétní expertní otázky. Rozhodl jsem se založit vyhodnocování vztahů na predikátové logice. Definoval jsem základní tři predikáty:

- $P(X, A)$ – rodiče osoby X je množina osob A .
- $C(X, A)$ – dětmi osoby X je množina osob A .

- $N(X, A)$ – množina osob A neobsahuje množinu osob X .

Z těchto základních predikátů lze následně sestavit složitější formule. Například pro prarodiče je pravidlo $P(X, A) \wedge P(A, B)$. Toto pravidlo představuje rodiče svých rodičů, tedy mé prarodiče.

Pro seznam všech pravidel je použit textový soubor, který je opět zadán jako vstupní při spuštění programu. Každé pravidlo je zapsáno na samostatné řádce v následujícím formátu:

```
[id; nazev_vztahu; muzsky_tvar_vztahu; muzsky_tvar_vztahu_ve4pade; zensky_tvar_vztahu; zensky_tvar_vztahu_ve4pade; pravidlo].
```

3 Návrh řešení a Implementace

Pro tuto úlohu jsem zvolil programovací jazyk Java, protože s ním mám již zkušenosti a jedná se o objektově orientovaný jazyk, jehož vlastnosti jsou v programu využívány.

Program je rozdělen do několika datových tříd, dále obsahuje třídu *cli* použitou jako textové uživatelské rozhraní a třídu *Main*, která se stará o spuštění a inicializaci celého programu.

3.1 Hlavní datové třídy

Třída *Person* reprezentuje osobu a uchovává si informace této osobě, tzv. atributy, jako jméno, příjmení, pohlaví, a odkazy (též reference) na rodiče a případně i partnera.

Třída *Relation* reprezentuje vztahy pro expertní vyhodnocování dotazů. Atributy této třídy jsou názvy vztahů pro mužské a ženské osoby, a též pravidlo složené ze základních predikátů.

Třída *Lineage* uchovává model rodinného stromu, seznam vztahů a metody, pomocí kterých je nad tímto stromem možné vyhodnocovat požadované dotazy.

3.2 Podpůrné datové třídy

Tyto datové třídy rozšiřují atributy hlavních datových tříd. Jedná se o třídu *Gender* a *MFString*. *Gender* je výčetová třída poskytující atributy a metody pro práci s pohlavím. Druhá zmíněná *MFString* poskytuje rozdílný řetězec podle pohlaví, a též rozlišuje mezi 1. a 4. pádem podstatných jmen.

4 Obsluha programu

Program lze spustit z příkazového řádku pomocí příkazu

```
java genealogy.Main -d personData.txt -r relationData.txt
```

Pokud není některý ze vstupních souborů zadán, použijí se implicitní soubory *data.ini* a *rules.ini*, jsou-li dostupné.

Po spuštění programu a načtení potřebných dat můžeme program ovládat několika nastavenými příkazy:

- *add* – přidání nové osoby nebo vztahu.
- *print* – vytisknutí rodiného stromu nebo seznamu dostupných pravidel.
- *expert* – umožňuje pokládat programu otázky ohledně rodinných vztahů.
- *help* – zobrazí seznam příkazů s nápovědou.
- *exit* – ukončí aplikaci.

Každý příkaz, je-li zadán bez dalších parametrů, vypíše nápovědu pro jeho správné použití.

5 Závěr

Program splňuje všechny body zadání, umožňuje odpovídat na různé otázky a je plně funkční.

Aplikace disponuje pouze textovým uživatelským rozhraním tzv. CLI, které nenabízí tak vysoký komfort a tolik možností jaké by umožnilo grafické uživatelské rozhraní tzv. GUI. Nicméně celé CLI je od programové logiky odděleno, takže tvorba GUI je jen otázkou doplnění patřičné třídy starající se o vytvoření grafických prvků a jejich propojení s programovou logikou.