

**ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA
PROVOZNĚ EKONOMICKÁ FAKULTA
KATEDRA INFORMAČNÍHO INŽENÝRSTVÍ**



Teorie ICT

Zápočtová práce

Jídelní lístek

Zdeněk Styblík, 1. ročník, INFONK

Obsah

Obsah.....	2
Popis zvoleného problému.....	3
Teorie.....	3
Analýza problému.....	3
Implementace.....	3
Vysvětlení programu.....	7
Očekávaný vstup.....	7
Metody.....	8
Ukázka trasování.....	8
Použité zdroje.....	10

Popis zvoleného problému

Máme nabídku předkrmů, polévek, hlavních jídel a moučníků. Každý pokrm má cenu a kalorickou hodnotu. Úkolem je sestavit nabídku menu podle různých kritérií, např. minimální kalorická hodnota, minimální cena, minimální kalorická hodnota při nepřekročení zadané ceny atd.

Teorie

Prolog byl vyvinut v sedmdesátých letech minulého století Alainem Colmerauerem ve Francii a Robertem Kowalskim ve Velké Británii. Prolog je zkratkou od "programování v logice" a na rozdíl jiných programovacích jazyků je deklarativní. To znamená, že programátor pouze stanoví cíl a jeho dosažení je ponecháno na Prologu. Naopak v procedurálním programovacím jazyce musí programátor detailně specifikovat co a jak se má provést.

Programování:

- deklarací faktů popisujících explicitní vztahy mezi objekty a vlastnostmi, které mohou mít
- definicí pravidel implicitních vztahů mezi objekty a pravidly definující implicitní vlastnosti objektů
- a potom na základě kladení dotazů ohledně těchto vztahů

Datové typy v Prologu jsou:

- term
 - struktura
 - jednoduchý term
 - proměnná
 - konstanta
 - číslo
 - atom

Analýza problému

Data bude nejlepší(nejsnazší) nadeklarovat jako fakta(term). Pro sestavení menu a výpočty bude nejvhodnější napsat metody, které nám jednak zabrání duplicitě kódu a zároveň ušetří spoustu psaní, resp. kopírování stejných řádek kódu/výroků.

Implementace

```
% fakta - predkrmy
predkrm(kruti_rolka, 70, 595).
predkrm(carpaccio_de_ternera, 60, 665).
predkrm(krevetovy_koktejl, 100, 591).
predkrm(morsky_koktejl, 90, 328).
predkrm(zadny, 0, 0).
```

```
% fakta - polevky
```

polevka(borsc, 20, 185).
 polevka(bramboracka_s_houbami, 22, 750).
 polevka(cibulova_se_syrem, 25, 3155).
 polevka(cesnekova, 24, 500).
 polevka(cinska_instantni_slepici, 15, 915).
 polevka(cockova, 25, 637).
 polevka(brokolicova, 20, 247).
 polevka(fazolova, 20, 775).
 polevka(gaspacho, 25, 350).
 polevka(gulasova, 23, 440).
 polevka(hrachova, 22, 135).
 polevka(kapustova, 21, 652).
 polevka(zadny, 0, 0).

% fakta - hlavni jidla

hjidlo(sladke_knedliky, 99, 1066).
 hjidlo(bavorske_vdolky, 90, 2160).
 hjidlo(bolonske_lasagne, 115, 975).
 hjidlo(bramboraky_4ks, 88, 2922).
 hjidlo(svickova_na_smetane_knedlik, 100, 1860).
 hjidlo(gulas_s_hovezim_masem_a_fazolemi, 95, 295).
 hjidlo(hovezi_gulas_s_knedliky, 85, 967).
 hjidlo(steak_z_tunaka, 120, 2347).
 hjidlo(kureci_rizek_bramborova_kase, 90, 798).
 hjidlo(hovezi_rostena, 100, 772).
 hjidlo(sekana_pecene, 93, 2020).
 hjidlo(veprova_pecene, 85, 1177).
 hjidlo(veprove_rizoto_se_zeleninou_syrem, 85, 816).
 hjidlo(zadny, 0, 0).

% fakta - mouchniky

mouchnik(vetrik, 30, 552).
 mouchnik(marokanka, 15, 215).
 mouchnik(kremrole_se_slehackou, 20, 756).
 mouchnik(bananek_v_cokolade, 12, 793).
 mouchnik(cokoladova_rolada, 40, 1472).
 mouchnik(indian, 23, 688).
 mouchnik(konakova_spicka, 25, 837).
 mouchnik(zadny, 0, 0).

% menu limitovane cenou shora. hlavni jidlo musi byt vzdy a bezpodminecne
 % zahrnuto v menu.

% ocekavane parametry jsou C ~ Cena.

```

menu_cena_max(C) :- predkrm(PR, C_PR, _),
    polevka(PO, C_PO, _),
    hjidlo(HJ, C_HJ, _),
    mouchnik(MO, C_MO, _),
    C_HJ > 0, (C_HJ + C_PR + C_PO + C_MO) =< C,
    write('# Nazev, cena'), nl,
    write('Predkrm: '), write(PR), write(', '), write(C_PR), nl,

```

```

write('Polevka: '), write(PO), write(', '), write(C_PO), nl,
write('Hlavni jidlo: '), write(HJ), write(', '), write(C_HJ), nl,
write('Moucnik: '), write(MO), write(', '), write(C_MO), nl,
write('---'), nl.

```

```

% menu limitovane cenou zdola. hlavni jidlo musi byt vzdy a bezpodminecne
% zahrnuto v menu.

```

```

% ocekavane parametry jsou C ~ Cena.

```

```

menu_cena_min(C) :- predkrm(PR, C_PR, _),
    polevka(PO, C_PO, _),
    hjidlo(HJ, C_HJ, _),
    moucnik(MO, C_MO, _),
    C_HJ > 0, (C_HJ + C_PR + C_PO + C_MO) >= C,
    write('# Nazev, cena'), nl,
    write('Predkrm: '), write(PR), write(', '), write(C_PR), nl,
    write('Polevka: '), write(PO), write(', '), write(C_PO), nl,
    write('Hlavni jidlo: '), write(HJ), write(', '), write(C_HJ), nl,
    write('Moucnik: '), write(MO), write(', '), write(C_MO), nl,
    write('---'), nl.

```

```

% menu limitovane pouze cenou shora, tzn. je mozne vybrat cokoliv.

```

```

% ocekavane parametry jsou C ~ Cena.

```

```

menu_free_cena_max(C) :- predkrm(PR, C_PR, _),
    polevka(PO, C_PO, _),
    hjidlo(HJ, C_HJ, _),
    moucnik(MO, C_MO, _),
    (C_PR + C_PO + C_HJ + C_MO) =< C,
    write('# Nazev, cena'), nl,
    write('Predkrm: '), write(PR), write(', '), write(C_PR), nl,
    write('Polevka: '), write(PO), write(', '), write(C_PO), nl,
    write('Hlavni jidlo: '), write(HJ), write(', '), write(C_HJ), nl,
    write('Moucnik: '), write(MO), write(', '), write(C_MO), nl,
    write('---'), nl.

```

```

% menu limitovane poctem kalorií shora. hlavni jidlo musi byt vzdy a

```

```

% bezpodminecne zahrnuto v menu.

```

```

% ocekavane parametry jsou K ~ Kalorie.

```

```

menu_kalorie_max(K) :- predkrm(PR, _, K_PR),
    polevka(PO, _, K_PO),
    hjidlo(HJ, _, K_HJ),
    moucnik(MO, _, K_MO),
    K_HJ > 0, (K_HJ + K_PR + K_PO + K_MO) =< K,
    write('# Nazev, kcal'), nl,
    write('Predkrm: '), write(PR), write(', '), write(K_PR), nl,
    write('Polevka: '), write(PO), write(', '), write(K_PO), nl,
    write('Hlavni jidlo: '), write(HJ), write(', '), write(K_HJ), nl,
    write('Moucnik: '), write(MO), write(', '), write(K_MO), nl,
    write('---'), nl.

```

```

% menu limitovane poctem kalorií zdola. hlavni jidlo musi byt vzdy a

```

```

% bezpodminecne zahrnuto v menu.
% ocekavane parametry jsou K ~ Kalorie.
menu_kalorie_min(K) :- predkrm(PR, _, K_PR),
    polevka(PO, _, K_PO),
    hjidlo(HJ, _, K_HJ),
    moucnik(MO, _, K_MO),
    K_HJ > 0, (K_HJ + K_PR + K_PO + K_MO) >= K,
    write('# Nazev, kcal'), nl,
    write('Predkrm: '), write(PR), write(', '), write(K_PR), nl,
    write('Polevka: '), write(PO), write(', '), write(K_PO), nl,
    write('Hlavni jidlo: '), write(HJ), write(', '), write(K_HJ), nl,
    write('Moucnik: '), write(MO), write(', '), write(K_MO), nl,
    write('---'), nl.

% menu limitovane poctem kalorií a cenou shora.
% ocekavane parametry jsou K ~ Kalorie, C ~ Cena.
menu_kalorie_max_cena_max(K, C) :- predkrm(PR, C_PR, K_PR),
    polevka(PO, C_PO, K_PO),
    hjidlo(HJ, C_HJ, K_HJ),
    moucnik(MO, C_MO, K_MO),
    K_HJ > 0,
    (K_HJ + K_PR + K_PO + K_MO) =< K,
    (C_PR + C_PO + C_HJ + C_MO) =< C,
    write('# Nazev, cena, kcal'), nl,
    write('Predkrm: '), write(PR), write(', '), write(C_PR),
        write(', '), write(K_PR), nl,
    write('Polevka: '), write(PO), write(', '), write(C_PO),
        write(', '), write(K_PO), nl,
    write('Hlavni jidlo: '), write(HJ), write(', '), write(C_HJ),
        write(', '), write(K_HJ), nl,
    write('Moucnik: '), write(MO), write(', '), write(C_MO),
        write(', '), write(K_MO), nl,
    write('---'), nl.

% menu limitovane poctem kalorií zdola a cenou shora.
% ocekavane parametry jsou K ~ Kalorie, C ~ Cena.
menu_kalorie_min_cena_max(K, C) :- predkrm(PR, C_PR, K_PR),
    polevka(PO, C_PO, K_PO),
    hjidlo(HJ, C_HJ, K_HJ),
    moucnik(MO, C_MO, K_MO),
    K_HJ > 0,
    (K_HJ + K_PR + K_PO + K_MO) >= K,
    (C_PR + C_PO + C_HJ + C_MO) =< C,
    write('# Nazev, cena, kcal'), nl,
    write('Predkrm: '), write(PR), write(', '), write(C_PR),
        write(', '), write(K_PR), nl,
    write('Polevka: '), write(PO), write(', '), write(C_PO),
        write(', '), write(K_PO), nl,
    write('Hlavni jidlo: '), write(C_HJ), write(', '),
        write(HJ), write(', '), write(K_HJ), nl,

```

```

write('Moucnik: '), write(MO), write(' '),
    write(C_MO), write(' '), write(K_MO), nl,
write('---'), nl.

% vypise Hlavni Jidla
vypis_hlavni_jidla :- write('Seznam hlavnich jidel - nazev, cena, kcal:'), nl,
    hjidlo(HJ, C_HJ, K_HJ), write(HJ), write(' '), write(C_HJ),
    write(' '), write(K_HJ), nl, fail.

% vypise Moucniky
vypis_moucniky :- write('Seznam moucniku - nazev, cena, kcal:'), nl,
    moucnik(MO, C_MO, K_MO), write(MO), write(' '), write(C_MO),
    write(' '), write(K_MO), nl, fail.

% vypise Polevky
vypis_polevky :- write('Seznam polevek - nazev, cena, kcal:'), nl,
    polevka(PO, C_PO, K_PO), write(PO), write(' '), write(C_PO),
    write(' '), write(K_PO), nl, fail.

% vypise Predkrmy
vypis_predkrmy :- write('Seznam predkrmu - nazev, cena, kcal:'), nl,
    predkrm(PR, C_PR, K_PR), write(PR), write(' '), write(C_PR),
    write(' '), write(K_PR), nl, fail.

```

Vysvětlení programu

Očekávaný vstup

Očekává se, že uživatel bude volat požadované metody ručně. Očekávaným vstupem jsou tedy názvy předem deklarovaných metod a jejich parametry. Následuje krátká ukázka takového volání:

```

151 ?- consult('e:/xstyz101_jidelni_listek.pl').
% e:/xstyz101_jidelni_listek.pl compiled 0.01 sec, 1 clauses
% Pozn.: nacteme Prolog soubor, který deklaruje data a metody
true.

152 ?- vypis_predkrmy.
% Pozn.: vypise seznam predkrmu, resp. znamych faktu
Seznam predkrmu - nazev, cena, kcal:
kruti_rolka, 70, 595
carpaccio_de_ternera, 60, 665
krevetovy_koktejl, 100, 591
morsky_koktejl, 90, 328
zadny, 0, 0
false.

153 ?- menu_kalorie_max(2000).
% Pozn.: vypise mozne kombinace jidel, ktere maji soucet kalorií mensi nez 2000.
# Nazev, kcal
Predkrm: kruti_rolka, 595

```

Polevka: borsc, 185
Hlavní jídlo: sladke_knedliky, 1066
Moucník: zadny, 0

true .

Metody

- menu_cena_max(<CENA_MAX>)
- menu_cena_min(<CENA_MIN>)
- menu_free_cena_max(<CENA_MAX>)
- menu_kalorie_max(<KCAL_MAX>)
- menu_kalorie_min(<KCAL_MIN>)
- menu_kalorie_max_cena_max(<KCAL_MAX>, <CENA_MAX>)
- menu_kalorie_min_cena_max(<KCAL_MIN>, <CENA_MAX>)
- vypis_hlavni_jidla()
- vypis_moucniky()
- vypis_polevky()
- vypis_predkrmy()

Ukázka trasování

Ukázka trasování metody menu_free_cena_max() při zavolání s parametrem 210:

```
% zapne trasovani prikazu  
[trace] [2] 6 ?- trace.  
true.  
% zavoláme danou metodu a předáme jí parametr 210  
[trace] [2] 6 ?- menu_free_cena_max(210).  
Call: (18) menu_free_cena_max(210) ? creep  
Call: (19) predkrm(_G2088, _G2089, _G2090) ? creep  
Exit: (19) predkrm(kruti_rolka, 70, 595) ? creep  
Call: (19) polevka(_G2088, _G2089, _G2090) ? creep  
Exit: (19) polevka(borsc, 20, 185) ? creep  
Call: (19) hjidlo(_G2088, _G2089, _G2090) ? creep  
Exit: (19) hjidlo(sladke_knedliky, 99, 1066) ? creep  
Call: (19) moucnik(_G2088, _G2089, _G2090) ? creep  
Exit: (19) moucnik(vetrik, 30, 552) ? creep  
Call: (19) 70+20+99+30=<210 ? creep  
Fail: (19) 70+20+99+30=<210 ? creep  
Redo: (19) moucnik(_G2088, _G2089, _G2090) ? creep  
Exit: (19) moucnik(marokanka, 15, 215) ? creep  
% ověření podmínky dle zadaného parametru  
Call: (19) 70+20+99+15=<210 ? creep  
Exit: (19) 70+20+99+15=<210 ? creep
```

% podmínka je splněna, následuje vypsání výstupu

Call: (19) write('# Nazev, cena') ? creep

Nazev, cena

Exit: (19) write('# Nazev, cena') ? creep

Call: (19) write('Predkrm: ') ? creep

Predkrm:

Exit: (19) write('Predkrm: ') ? creep

Call: (19) write(kruti_rolka) ? creep

kruti_rolka

Exit: (19) write(kruti_rolka) ? creep

Call: (19) write(', ') ? creep

,

Exit: (19) write(', ') ? creep

Call: (19) write(70) ? creep

70

Exit: (19) write(70) ? creep

Call: (19) write('Polevka: ') ? creep

Polevka:

Exit: (19) write('Polevka: ') ? creep

Call: (19) write(borsc) ? creep

borsc

Exit: (19) write(borsc) ? creep

Call: (19) write(', ') ? creep

,

Exit: (19) write(', ') ? creep

Call: (19) write(20) ? creep

20

Exit: (19) write(20) ? creep

Call: (19) write('Hlavni jidlo: ') ? creep

Hlavni jidlo:

Exit: (19) write('Hlavni jidlo: ') ? creep

Call: (19) write(sladke_knedliky) ? creep

sladke_knedliky

Exit: (19) write(sladke_knedliky) ? creep

Call: (19) write(', ') ? creep

,

Exit: (19) write(', ') ? creep

Call: (19) write(99) ? creep

99

Exit: (19) write(99) ? creep

Call: (19) write('Moucnik: ') ? creep

Moucnik:

Exit: (19) write('Moucnik: ') ? creep

Call: (19) write(marokanka) ? creep

marokanka

Exit: (19) write(marokanka) ? creep

Call: (19) write(', ') ? creep

,

Exit: (19) write(', ') ? creep

Call: (19) write(15) ? creep

```

15
  Exit: (19) write(15) ? creep
  Call: (19) write(---) ? creep
---
  Exit: (19) write(---) ? creep
  Exit: (18) menu_free_cena_max(210) ? creep
true .

  Call: (7) fail ? creep
  Fail: (7) fail ? creep
% Prolog hledá další alternativní řešení
  Redo: (7) predkrm(_G513, _G514, _G515) ? creep
  Exit: (7) predkrm(carpaccio_de_ternera, 60, 665) ? creep
  Call: (7) write(carpaccio_de_ternera) ? creep
carpaccio_de_ternera
  Exit: (7) write(carpaccio_de_ternera) ? creep
  Call: (7) write(', ') ? creep
,
  Exit: (7) write(', ') ? creep
  Call: (7) write(60) ? creep
60
  Exit: (7) write(60) ? creep
  Call: (7) write(', ') ? creep
,
  Exit: (7) write(', ') ? creep
  Call: (7) write(665) ? creep
665
  Exit: (7) write(665) ? creep
  Call: (7) fail ? creep
  Fail: (7) fail ? creep
...
...
...
% Pokračuje dokud nejsou nalezena/vypsána všechna řešení

```

Použité zdroje

1. https://cs.wikipedia.org/wiki/Prolog_%28programovac%C3%AD_jazyk%29#Oper.C3.A1tory
2. <http://www.kaloricketabulky.cz>
3. <http://cs.union.edu/~striegnk/courses/esslli04prolog/index.php>
4. <http://www.swi-prolog.org/FAQ/>
5. http://www.csupomona.edu/~jrfisher/www/prolog_tutorial/contents.html
6. <http://jmvanel.free.fr/ai/prolog-getting-started.html>
7. <http://cs.union.edu/~striegnk/courses/esslli04prolog/practical.day2.php?s=practical.day2.node2>