

1 PRG005 Neprocedurální programování

1.1 přednáška úterý 15:40 M1 - cvičení začínají druhý týden výuky

Základní kurs neprocedurálního programování pro studenty informatiky.

- Vyhláška o rozdělení studentů do skupin na seminářích verze 15.11.
- Vyhláška o přihlašování na zkoušky - přihlašovat se lze až když již skutečně máte zápočet
- Obsah a cíle předmětu
- Co bylo na přednášce
- Příklady z Prologu
- Podmínky pro získání zápočtu
- Požadavky ke zkoušce
- Ukázková zadání písemky ve formátu rtf: první a druhá část
- Literatura
 - Základní literatura
 - Doplnková literatura
- Překladač SWI

1.2 Obsah a cíle předmětu

Úkolem předmětu je seznámit studenty s ideami neprocedurálního programování a vybavit je základními programovacími technikami logického a funkcionálního programování. Přednáška má tři hlavní části:

- Hlavní část přednášky je věnována **programování v jazyce Prolog** .
V Prologu se také vytvářejí a ladí zápočtové úlohy.
- V druhé (velmi krátké) části přednášky se seznámíme se **základy funkcionálního programování na bázi jazyka LISP** .
V seminářích (jednom nebo dvou) se přesvědčíte, že řadu obrátů, které jste si osvojili v Prologu, velmi snadno přenesete do programování v LISPu. Cílem této části je připravit půdu pro třetí část přednášky, LISP se u zkoušky nevyžaduje.
- Poslední část přednášky je věnována **výkladu funkcionálního programování na bázi jazyka Haskell** .
Nepůjdeme do teoretických detailů jazyka, ale soustředíme se spíše na praktické programování v něm.

Zpět začátek

1.3 Podmínky pro získání zápočtu

Student může získat zápočet za splnění dvou požadavků :

- Aktivní účast na seminářích a průběžné znalosti během semestru.
- Vytvoření a odladění zápočtového programu v Prologu.
K zápočtovému programu je potřeba vytvořit smysluplnou dokumentaci. Součástí předávaného programu musí být i zkušební ladící data. Jejich volba je součástí toho, co se na programu hodnotí.

Získání zápočtu je podmínkou přípuštění ke zkoušce.

Podmínky pro udělení zápočtu upřesňuje a kontroluje vedoucísemináře, na který student chodí. Při přebírání zápočtového programu sezpravidla vhodným způsobem přesvědčí zda student program vytvořil sám(např. může požadovat provedení drobných změn v programuv reálnémčase).

Pokud student nemůže ze závažných důvodů navštěvovat seminář se skupinou, kam podle rozvrhu patří, musí se nejpozději do dvou týdnů po zahájení výuky přihlásit do jiné skupiny. O přijetí či nepřijetí rozhoduje vedoucí semináře, kam chce student přejít. Kriteiem je udržení přibližně stejně velkých skupin.

Studenti, kteří se nepřihlásí (resp. nebudou přijati) na žádný zeseminářů do 30.10.2004 zařadí do skupin přednášející a pokud nebudou mít pro opoždění skutečně vážný důvod, dostanou jednu úlohu na zápočetnavíc.

Pokud jste se zúčastnili některého semináře se skupinou, do které nebudete chodit, dejte to vedoucímu cvičení na vědomí.

V souboru jsou uvedeny aktuální počty studentů ve skupinách k 10.listopadu. Podle počtu studentů byl každému ze seminářů přidělen statut. Jejich význam je následující:

- uzavřeno - nelze již přijímat nové studenty
- otevřeno - přijímání dalších studentů není žádoucí
- volné - přijímání dalších studentů je žádoucí
- velmi volné - přijímání dalších studentů je velmi žádoucí
- přeplněno - je nutné snížit počet studentů na nejvýše 24 resp. 25

Pokud student (kombinovaného) studia nemůže ze závažných důvodů semináře navštěvovat, dohodne si začátkem semestru náhradní podmínky udělení zápočtu s přednášejícím.

Zpět začátek

1.4 Literatura:

1.4.1 Základní literatura:

1. **Rudolf Kryl: Úvod do programovacího jazyka Prolog** učební text - ke stažení z WWW stránek přednášejícího aktuální verze 3.03 ve formátu pdf nebo zip. *Text byl původně napsán pro čtenáře s o poznání menší programátorskou erudicí, než má většina z Vás. Proto by mělo jít o "nenáročné čtení před usnutím". Text pokrývá přibližně první polovinu látky z Prologu. Obsahuje řadu jednoduchých programů.* Budu vděčný za

připomínky a opravy chyb - pošlete je na můj mail a v subjectu uveďte název textu a číslo verze, ke které se připomínky vztahují.

2. **Přehled předdefinovaných forem ve vykládaném dialektu LISPu** Ke stažení ve formátu pdf
3. **Paul Hudak, Joseph H. Fasel: A Gentle Introduction to Haskell** ke stažení z www.haskell.org *Poměrně čtivý základní text o programovacím jazyku Haskell. Pokrývá vše z jazyka, co probereme. Některé partie vynecháme, půjdeme však podstatně dál v programátorské obtížnosti úloh.* Na dané adrese můžete získat úplný přehled o problematice jazyka Haskell.

Zpět začátek

1.4.2 Doplnková literatura (není nutná)

1. **P.Jirků a kol. : Programování v jazyku Prolog** SNTL 1991 *Pěkná knížka v češtině. Neobsahuje mnoho těžších příkladů, věnuje se však dost podrobně souvislostem s logikou.*
2. **Bratko I.: PROLOG Programming for Artificial Intelligence** Addison-Wesley, Reading, Massachusetts, 1986 ISBN 0-201-14224-4 *Stále velmi pěkná knížka. První díl obsahuje výklad Prologu (naš přístup k výkladu je do značné míry poplatný přístupu této knihy), druhý obsahuje výklad některých základních partií umělé inteligence pomocí programů v Prologu.*
3. **Harold Abelson, Gerald Jay Sussman, Julie Sussman : Structure and Interpretation of Computer Programs** Mc Graw-Hill Book Company 1985 ISBN 0-07-000-422-6 *Výklad principů programování na bázi jazyka XSCHEME (dialekt jazyka LISP). Kniha byla používána v základním kursu programování na MIT. My použijeme tohoto jazyka i příkladů z této knihy k velmi zběžnému seznámení s principy LISPu.*
4. **Ivan Kalaš : Iné programovanie - stretnutie s jazykom Lisp** Alfa 1992 *Krásná kniha s množstvím originálních příkladů. Užívaný dialekt LISPu se poněkud liší od toho, se kterým budeme pracovat na přednášce my.*

Zpět začátek

1.5 Překladač prologu SWI

K ladění zápočtových programů a experimentům v Prologu doporučuji překladač SWI Prolog. Jde o volně stažitelný výborný překladač, který je k dispozici pro všechny platformy. Obecně lze experimenty s jazykem vřele doporučit, možná však je dobré počkat až z přednášky a seminářů ovládnete základy jazyka. Na třetí nebo na čtvrté přednášce si o práci s tímto překladačem řekneme více.

Zpět začátek

1.6 Informace o zkoušce

Nutnou podmínkou přípuštění ke zkoušce je předchozí získání zápočtu.

Ke zkoušce se přihlašujete ve studijním informačním systému.

Přihlašujte se až když již zápočet skutečně máte, jinak byste blokovali kapacitu termínů studentům, kteří na rozdíl od Vás zkoušku skládat mohou. Jde tedy o elementární slušnost k ostatním. Jen velmi nerad bych takové rozumné a ohleduplné chování vynucoval administrativními metodami (které již z definice musí být slepě tvrdé, musel bych např. zakázat odhlašování od zkoušky). Uchýlil bych se k nim jen kdybyste mě k tomu donutili. Studenty, kteří se na termíny již přihlásili, aniž měli zápočet, jsem odhlásil. Nehodlám však nadále takové nedostatky funkční gramotnosti studentů napravit (*Funkční gramotnost = schopnost porozumět krátkému textu v příslušném přirozeném jazyce a schopnost zachovat se adekvátně tomuto porozumění*).

Není třeba být nervozní, **termínů bude** - zvláště ve zkouškovém období - **dost**. Pokud by již vypsané termíny nestačili, nějaké doplním. U některých termínů půjde zvednout i kapacita - to se však netýká předtermínu. Pochopitelně přijdete-li na zkoušku a bude volná kapacita, budete moci zkoušku skládat i když jste nebyli přihlášení.

I když to nebudu nijak administrativně vynucovat, je rozumné naplánovat si práci tak, abyste mohli zkoušku (jednou - většinou to snad bude stačit) absolvovat do konce zkouškového období zimního semestru. Během semestru budou termíny také, ale pochopitelně vzhledem k obsazenosti poslucháren v podstatně menším počtu.

Zpět začátek

1.7 Požadavky ke zkoušce

Zkouška má **písemnou** a **ústní část**, větší váhu má část písemná.

1.7.1 Písemná část se skládá také ze dvou částí

- **první písemná práce na 90 minut**
obsahuje 4-5 jednoduchých příkladů, v nichž má student prokázat, že ovládl základní techniky neprocedurálního programování (u každého příkladu je předepsané, zda má být řešen v Prologu nebo Haskellu, případně i jak).
- **druhá písemná práce na 120 minut**
obsahuje dvě středně obtížné úlohy, v nichž má student prokázat, že umí programovat jak Prologu, tak v Haskellu. Zpravidla si student může vybrat, který problém bude řešit v Prologu a který v Haskellu. Pokud nebude řečeno jinak, předpokládá se vyřešení algoritmického jádra problému bez technických podrobností jako je např. vstup a výstup.

1.7.2 Ústní část zkoušky se skládá

- ze společné (zkoušející a zkoušený) **opravy vytvořených programů** a diskuse nad nimi
- ze **zodpovězení otázek** z některého z dále uvedených témat

Témata pro ústní zkoušku :

- PROLOG

1. Tvar programu v Prologu a jeho interpretace
2. Deklarativní a operační sémantika programu v Prologu
3. Operátor řezu
4. Negace
5. Práce se seznamy
6. Reprezentace datových struktur (např. grafy, stromy, rozdílové seznamy)
7. Edinbugrský model vstupu a výstupu.
8. Definování a použití operátorů
9. Predikáty pro řízení databáze (assert,...)
10. Predikáty grupování termů (bagof, setof) a jejich užití
11. Efektivita programů v Prologu

- HASKELL

1. Typy v Haskellu, typová specifikace funkce
2. Základní způsoby definování výrazů,
3. Sémantika "mečování" parametrů, as patterns (@s), žolíky (-), lazy-parametry (~x),
4. Lazy vyhodnocování, "nekonečné" termy.
5. Lambda abstrakce.
6. Používání jmen funkcí jako operátorů a naopak, specializace operátorů { (např. (x+)} .Definování priority a asociativity infixových operátorů.
7. Třídy, podtřídy, instance.
8. Pole v Haskellu.

Zpět začátek