

Лабораторна робота №1.2

Структура програми на Turbo-Prolog

Мета роботи: ознайомитись зі структурою Пролог-програм, вивчити основні прийоми відладки і виконання програм в середовищі TurboProlog.

Завдання роботи: набрати та запустити в середовищі Турбо-Прологу прості програми.

Короткі теоретичні відомості

Програма на Turbo Prolog складається з кількох розділів, які позначаються стандартними ключовими словами:

domains – оголошуються і описуються домени (тобто типи даних, визначені програмістом).

predicates – містить оголошення предикатів, введених програмістом.

clauses – містить базу знань про предметну область.

goal – розмішують внутрішню мету програми.

database – тут описують динамічні типи даних (бази даних).

Як правило, розділ *clauses* є основною, найбільш змістовною частиною програми на Пролозі. Він містить сукупність тверджень (висловлювань, речень, відношень; англ. – *clauses*), які за своєю суттю близькі до алгебри логіки та зчислення предикатів.

В розділі *clauses* можуть знаходитись твердження двох типів – *факти* і *правила*. Кожне речення (факт або правило) закінчується крапкою. В ході виконання або розгляду програми інтерпретатор Prolog виконує доведення цілей, які входять у ці твердження.

Факт – це одиночна ціль, яка вважається істинною за визначенням. Факти задаються у вигляді n-елементних структур, або кортежів об'єктів, які задовольняють відношенню згідно умови задачі.

Правило дозволяє вивести з фактів деякий інший факт.

Правило складається з голови і хвоста.

Голова правила істинна, якщо істинні всі цілі, вказані у хвості.

Процедура являє собою набір речень, що стосуються одного й того ж відношення (мають один головний функтор).

У системі Prolog для визначення відповіді на запит часто застосовується складний процес, який пов'язаний з виконанням логічного виведення, розглядом багатьох альтернатив і, можливо, перебору з поверненнями. Всі ці операції виконуються системою

Prolog автоматично, і приховані від користувача. Однак система передбачає можливість покрокового виконання програми (трасування). Для трасування слід в початок програми вставити директиву компілятора “trace” і запустити програму (Alt-R). Після цього покрокове проходження програми виконується за допомогою F10.

Порядок виконання роботи.

Завдання 1. Ознайомтеся з наведеними теоретичними відомостями, а також перегляньте додатки 1–4. Вони містять додаткову інформацію, яка може стати вам у нагоді при виконанні завдань роботи.

Завдання 2. Напишіть і виконайте програму, що відобразить вікно, в якому буде відбуватися введення і виведення інформації. Для входу у вікно редагування (натисніть Alt-E або виберіть у головному меню пункт Edit). Введіть наступний текст програми:

```
goal
    makewindow(1,7,7,"Зареєструйтеся",4,56,14,22), nl,
    write ("Будь ласка введіть своє\n ім'я, а потім
    натисніть\nEnter."),
    cursor(6,4),readln(Name), nl,
    write ("Ласкаво просимо до\n Турбо Прологу\n",Name), nl.
```

ПРИМІТКА. Якщо у вас Пролог некоректно відображає кириличні літери, пишіть латиницею:

```
goal
    makewindow(1,7,7,"Register please",4,56,14,22), nl,
    write ("Bud' laska vvedit' svoye\n im'ia, a potim
    natisnit'\nEnter."),
    cursor(6,4),readln(Name), nl,
    write ("Laskavo prosimo do\n Turbo Prologu\n",Name), nl.
```

Завдання 3. Як ви бачили, в попередній програмі (завдання 2) вся програма була записана у розділі goal. Змініть програму, перенісши змістовну частину до розділу clauses. Переконайтеся у працездатності отриманої програми.

Завдання 4. Виконайте трасування програми, отриманої в результаті виконання завдання 3.

Завдання 5. Спробуйте по одному змінити значення аргументів функції **makewindow**. Встановіть, яке значення має кожний з аргументів, і вкажіть це у звіті.

Завдання 6. Складіть програму для введення й виведення інформації про деяку людину, у результаті виконання якої в одному вікні вводиться прізвище, ім'я, по батькові людини, а також його адреса й телефон, а в іншому вікні уведена інформація виводиться у вигляді картки за таким зразком.

Ввод інформації
Введіть фамілію: Іванов
Введіть ім'я: Іван
Введіть отчество: Іванович
Адрес: Воинова, 8-12
Телефон: 23-45-76

Учетная запись N1
Іванов
Іван
Іванович
Воинова, 8-12
23-45-76

Примітка: Чомусь у більшості тих, хто це робить, виходить навпаки: перше вікно справа, а друге зліва. Це незручно для користувача, бо у нашій культурі розвиток подій прийнято зображати зліва направо. Тому зробіть так, щоб вікно для введення інформації знаходилося зліва, а друге вікно – справа.

Завдання 7. Зробіть так, щоб букви в другому вікні виводилися на кольоровому тлі і *загадково мерехтіли*.

Завдання 8. Розгляньте наступний приклад програми. Сформулюйте природною мовою (українською) записані у програмі відомості. Поставте до програми прості запити.

```
predicates  
    likes (symbol, symbol)
```

```

/* likes (person, activity)*/
clauses
  likes(ellen,tennis).
  likes(john,football) .
  likes(tom,baseball) .
  likes(eric,swimming).
  likes(mark,tennis) .
  likes (bill,X) if likes(tom,X) .

```

Завдання 9. Запишіть наступні факти й правила у формі, прийнятній для Turbo-Prolog :

```

ellen likes reading
john likes computers
eric likes swimming
david likes computers
mary likes X if john likes X
gina likes anything eric likes (Джина любить усе, що любшть Ерик)

```

Завдання 10. Складіть програму на Пролозі, що описує відомості про осіб предикатами типу : person("Ivan", 24, 1.73, 'm').

- ім'я (тип string: "Ivan"),
- вік (тип integer: 24),
- зріст (тип real: 1.73).
- стать (тип char, 'm' або 'f').

Задайте в такий спосіб дані про 5-6 людей.

Складіть запити, що виведуть:

10.1. імена всіх чоловіків: person(X,_,_, 'm'), write(X), nl, fail.

10.2. імена й стать людей, які мають зріст більше 1.50 (значення параметрів рекомендується задати так, щоб вивелося кілька записів): person(X, _, R, P), R>1.50, write(X, " ",P), nl, fail.

10.3. імена й зріст дівчат, молодших за 20 років.

10.4. Придумайте ще 2-3 власних запити і реалізуйте їх.

Завдання 11. Розгляньте приклад програми, наведений у Додатку 3. Побудуйте за таким же принципом власну базу знань (яка містить

факти і правила типу Якщо-То) про яких-небудь людей або об'єкти і зробіть до неї кілька запитів.

Завдання 12. Наведіть приклади для кожного з наступних типів мети (предметна область будь-яка): для кожного типу мети складіть по одній програмі мовою Пролог, або запишіть всі вказані види цілей до однієї програми:

- 12.1 Проста константна мета (ціль)
- 12.2 Проста мета зі змінними
- 12.3 Складена константна мета
- 12.4 Складена мета зі змінними
- 12.5 Зіставлення із реченням-правилом

Запишіть тексти програм до себе у зошит. По своєму зошиту (без комп'ютера) поясніть викладачу, як проходить пошук розв'язку в кожній з ваших програм. (Якщо важко це зробити одразу, спершу виконайте програми в Турбо-Пролозі і, використовуючи трасування, простежте за процедурою пошуку розв'язку.)

Запитання і завдання для самоперевірки

1. Перевірте, чи узяли Ви з собою зошит з конспектом лекцій?
(Без конспекта здати цю лабораторну буде важко ☺)
2. Як запустити програму, набрану в середовищі Турбо-Пролога?
3. Як виконати трасування програми?
4. Як визначається колір символів і колір фону в предикаті `makewindow`?
5. Яку інформацію містить розділ `clauses`?
6. Як задати питання системі Turbo-Prolog?
7. Як записати в Turbo-Prolog твердження «Маша вивчає математику»?
8. Як перевірити відповідність твердження до відомостей, записаних у Пролог-програмі? Наприклад, як перевірити істинність твердження «Маша вивчає математику»?
9. Як записуються змінні в Turbo-Prolog?
10. Яка різниця між фактами і правилами в мові Prolog?

11. Які розділи мають написані вами програми? Яке значення цих розділів?
12. Що таке ціль в Prolog-програмі? Які види цілей Ви знаєте?
13. Як виконується операція зіставлення з реченням-правилом?
14. Поясніть значення повідомлень CALL, RETURN, REDO, що виводяться при трасуванні.
15. Як реалізується дедукція (отримання логічного висновку) в Turbo-Prolog? Поясніть і наведіть приклади.
16. Що називається процедурою в Пролог-програмі? Чи є процедури у програмах, які Ви написали?

ДОДАТОК 1

Синтаксис предикатів *window

(тобто makewindow, removewindow, clearwindow та ін.)

Пролог-програма може забезпечити багатовіконний інтерфейс. Для цього у мові TurboProlog передбачені вбудовані предикати:

- створення вікна makewindow
- видалення вікна removewindow
- очищення вікна clearwindow
- перемикання активного вікна gotowindow, shiftwindow.

Синтаксис предиката makewindow

makewindow(№ вікна, кольори тексту і фону,
кольори рамки, заголовок рамки,
координати лівого верхнього кута вікна
(№ рядка 0..23, № стовпчика 0..79),
висота вікна, ширина вікна)

Коди кольорів наведені в таблиці.

Для одержання потрібної колірної гами коди необхідно додати, наприклад:

Синій символ - 1

На зеленому тлі - 32

Мерехтить (блимає) – 128

Загальний код дорівнює сумі кодів кольору тексту, кольору тла і спецефектів: $1+32+128=161$, от цю цифру й ставимо у функцію makewindow.

Кольори	Код кольорів Тексту / тла
Чорний – black	0/0
Синій – blue	1/16
Зелений – green	2/32
Блакитний – cyan	3/48
Червоний – red	4/64
Фіолетовий – magenta	5
Коричневий – brown	6

Ясно-сірий – lightgray	7
Темно-сірий – darkgray	8
Яскраво-синій – lightblue	9
Яскраво-зелений – lightgreen	10
Яскраво-блакитний – lightcyan	11
Рожевий – lightred	12
Малиновий – lightmagenta	13
Жовтий – yellow	14
Білий – white	15
мерехтіння	128
підкреслення ???	1 (???????)

Інші предикати *window

removewindow - видалення поточного вікна

removewindow(№ вікна, параметр 0 - не створювати вікон, що впливають за видаляємим вікном, що, 1 - створювати їх)

clearwindow - очищення поточного вікна

gotowindow(№ вікна) - перехід до вікна із заданим номером

shiftwindow(№ вікна) - перемикає або повертає номер активного вікна

ДОДАТОК 2

Приклад програми до завдання 3.

```
% trace
    predicates
        hello
    clauses
        hello:-
makewindow(1,7,7,"Зареєструйтеся",4,56,14,22), nl,
        write ("Будь ласка, введіть своє\n ім'я, а потім
натисніть\nввод."),
        cursor(6,4),readln(Name), nl,
        write ("Ласкаво просимо в\n Турбо Пролог\n",Name), nl.
    goal
        hello
```

Або латиницею:

```
trace
    predicates
        hello
    clauses
        hello:- makewindow(1,7,7,"Register
please",4,56,14,22), nl,
        write ("Pozhaluysta vvedite svoe\n imia, a
zatem nazhmite\nVvod."),
        cursor(6,4),readln(Name), nl,
        write ("Dobro pozhalovat' v\n Turbo
Prolog\n",Name), nl.
    goal
        hello
```

ДОДАТОК 3

Приклад програми до завдання 11.

(Автор програми – студент; я за достовірність описаних у ній подій не відповідаю 😊)

Словничок: spider – павук, cucumber – огірок, poisonous – отруйний.

predicates

```
human(symbol)
ate(symbol,symbol)
poisonous(symbol)
dead(symbol)
alive(symbol)
```

clauses

```
human(tom). human(olga).
human(kate). human(jack). human(nastya).
```

```
ate(tom,sandwich).
ate(olga,spider).
ate(kate,fish).
ate(kate,cucumber).
ate(jack,vorona).
ate(jack,myxomor).
ate(nastya,milk).
ate(nastya,chocolade).
```

```
poisonous(spider).
poisonous(poganka).
poisonous(myxomor).
```

```
dead(X) if human(X), ate(X,Y), poisonous(Y).
alive(X) if human(X), not(dead(X)).
```

```
/* dead(X) :- human(X),ate(X,Y),poisonous(Y). */
```

ДОДАТОК 4.

Редагування та запуск програм.Трасування.

Наведений далі уривок я узяв з якоїсь книжки. В тій книжці детально (часом аж занадто детально) описано, на які кнопки треба нажимати, щоб у вікні Прологу набрати програму з клавіатури. Я сподіваюся, що серед вас є люди, які здогадаються, що в нашому випадку цього можна і не робити.

Розглянемо наступну просту TURBO-PROLOG програму. Ми її використовуємо, щоб проілюструвати, як створити, відредагувати і виконати програму на TURBO-PROLOG.

```
goal
    makewindow(1,7,7,"Moia persha
programa",4,56,14,22), nl,
write ("Bud' laska vvedit' svoye\n im'ia, a potim
natisnit'\nEnter."),
cursor(6,4),readln(Name), nl,
write ("Laskavo prosimo do\n Turbo
Prologu\n",Name), nl.
```

Виберіть в меню редагування: або пересуваючи курсор за допомогою кнопок зі стрілками доти, поки не дійде до пункту меню EDIT (повинен стояти над словом EDIT) і потім натиснути на "Enter", або просто треба нажати на Alt-E. Курсор опиняється у вікні тексту програми.

Помітьте, що вікно редактора висвітлене й статус-рядок унизу відбиває нове значення функціональних клавіш.

Закінчити сеанс редагування можна натисканням клавіші Esc.

Щоб побачити, як виправити помилку, введіть перший рядок програми з помилкою - наприклад, **gool**. Але це помилкове написання. Спробуйте запустити програму (Alt-R). Оскільки **gool** - це синтаксична помилка, ви повинні побачити повідомлення про неї внизу вікна редактора. Курсор зупиниться над неправильною ділянкою тексту у вікні.

Помилку треба виправити. Щоб виправити помилку, розташуйте курсор над правильною буквою "o" і потім натисніть DEL. Уважно простежте, що відбудеться на дисплеї. Тепер натисніть A і подивіться знову.

Тепер введіть текст наведеної програми, натискаючи на "Enter" наприкінці кожного рядка.

Щоб побачити, що відбудеться, якщо ви зробили помилку при надрукуванні у вашому тексті, поверніться й вставте довільно помилку. Якщо вам не вдалося, поверніться до основного меню, натиснувши на штангу інтервалу або ESC (якщо ви заплуталися в системі, не хвилюйтеся, натисніть кілька разів ESC - і штанга інтервалу - без команди - завжди поверне вас до основного меню).

Зверніть увагу: Коли ви надрукуєте кінець рядка, що починається **makewindow** (довгий рядок), частина тексту переміститься вліво усередину вікна редактора. Щоб переконатися, що він не зник зовсім, натисніть "Enter" після того, як ввели цей рядок, потім натисніть на кнопку «стрілка вліво» або «стрілка вправо» і тримайте, поки не припинить рухатися курсор. Після того як ви побачите, що відбувається, пересуньте курсор назад і завершіть друкувати програму.

Як тільки ви переконаєтеся, що текст програми введений правильно, скористайтесь пунктом меню Run (натисніть Alt-R) для її запуску. Або натисніть ESC, щоб «відпустити» редактор, потім виберіть варіант RUN з меню.

```
trace
predicates
  hello
clauses
  hello:- makewindow(1,7,7,"Moia pervaiia programma",4,56,14,22),
nl,
        write ("Pozhaluysta vvedite svoe\n imia, a zatem
        nazhmite\nVvod."),
        cursor(6,4),readln(Name), nl,
        write ("Dobro pozhalovat' v\n Turbo Prolog\n",Name),
        nl.
goal
  hello
```

Щоб простежити за виконанням програми, ми додали директиву компілятора "trace" на початку програми. Тепер виберіть команду Run і подивіться, що відбувається на екрані.

У вікні редактора курсор висвічується наприкінці завдання “hello” і у вікні трасування початок виконання завдання показується як:

CALL: goal()

Можна тепер використати F10, щоб виконати завдання покроковим способом. Натисніть F10 один раз і вікно трасування тепер покаже:

CALL: hello()

реєструючи (для нашого відома), що зараз завдання повинно задовольнити предикат hello. Ще одне натискання пересуне курсор до визначення hello.

При цьому відбувається наступне: одержавши завдання задовольнити предикат hello, машина звертається до його опису й далі працює з ним, намагаючись з’ясувати, чи виконані (або чи можуть бути виконані) викладені там вимоги.

Натисніть ще раз F10 і помітьте у вікні трасування, що ми зараз викликали (CALL) предикат формування вікна *makewindow*. Ще одне натискання викликає виконання предиката формування вікна, що відображається у вікні трасування як:

RETURN: makewindow(1,7,7,”My first program”,4,56,10,22)

Помітьте, що при цьому праворуч нагорі екрана створюється вікно. *Зміст того, що відбувається, наступний: предикат makewindow після виконання повертає керування тому, хто його викликав, а крім того, повертає (return) повідомлення про те, що він виконаний успішно.*

Отже, перший пункт в описі предиката hello задоволений.

Тепер повторно натискайте F10 і спостерігайте за тим, що відбувається в кожному вікні, поки вікно трасування не покаже

CALL: readln(_)

Тепер курсор повинен блимати в тому місці у вікні виконання, де ми повинні надрукувати ім’я. Надрукуйте ім’я ТОМ, потім натисніть Enter. Вікно трасування тепер покаже

RETURN: readln("TOM")

Це нам теж зрозуміло: функція readln повертає введене значення.

Нарешті, натисніть повторно F10 поки вікно трасування не покаже

RETURN: hello()

RETURN: goal()

На цьому закінчилося виконання нашої програми. Натисніть F10 ще один останній раз і система дасть вам вказівку натиснути на штангу інтервалу, щоб повернутися до рівня основного меню.

Простеживши за тим, що відбувається, ви помітили, що на всіх кроках відбувається те ж саме. Кожен предикат викликається (що відбувається у вікні трасування словом CALL і його ім'ям), виконується й повертає (RETURN) повідомлення про успіх.

Уміння користуватися трасуванням обов'язково знадобиться нам під час відлагодження та дослідження програм на Turbo-Prolog.