# Агрегаты фактов

**1. Списки.**

Если в пр-ме необходимо организовать с переменными или заранее неопределенным количеством объектов, то испол. списки.

СПИСОК — упорядоченная последовательность эл-в одного типа неопределенной длины, кот м. состоять из 0 и более эл-в.

Константы ,попавшие в список , записываются в [ ] и отделяются друг от друга запятыми.

Исходя из определения списка, два списка , сост. из одних и тех же

эл-в , но расположенных в разном порядке считаются разными. В разделе domains эл-ты списка обяз-но д.б. сопоставлены с определенным типом домена (Sp=integer\*).

Список в domains м.б. опосредствован через имя переменной, эл-ты кот. собираются в список.

domains

   k\_fms = string, kol = integer,

   Sp = kol\*

predicates

   fms ( k\_fms, kd, ...)

Списки состоят из заголовка ( начала списка) и захвата (окончания списка). К заголовку относится только 1-й эл-т списка, остальное — хвост.

Список с нулевым количеством эл-в определяется как пустой список, он не имеет ни заголовка ни хвоста.

Для работы со списками в языке имеется ряд встроенных предикатов.

MEMO (<имя эл-та>, <список>) — где <список> м.б. задан либо именем Sp, либо непосредственно константами, входящими в этот список.

Предикат определяет принадлежность эл-та к списку.

I — деление списка на голову и хвост [ Head I Tail] или [H I T]

H — или переменной для обозначения заголовка

T — — “ — хвоста

ПРИМЕР

domains

   list=char\*

predicates

   test (List)

clouses

   test ( [‘A’, ‘B’ , ‘C’, ‘D’] ).

goal

   test ([H/T]),

   nl,

   write (H)

APPEND ( [ эл-ты старого списка ],[ эл-ты нового списка ],< имя нового списка > )

— из 2-х списков, старого и нового, создают III список.

REVERS ( [ эл-ты стар. списка], < имя нового списка > )

— меняет последовательность эл-в на противоположную

FINDALL (< имя переменной >,< формат предиката >,< имя списка >)

ПРИМЕР

domains

   post = string

   Kol, Cena, Sum = integer

   Sp = integer\* ( Sp = Kol\*)

   Kod = string

predicates

   tmc ( P, Kol, Cena)

   sum ( Sp, Sum)

goal

   write ( “ Введите код”),

   readln ( Kod),

   nl,

   findall ( Kol, tmc ( Pos1, Kod, Kol, Cena), Sp),

   sum (Sp, Sum),

   write ( “Количество =” , Sum)

clauses

   tmc ( “ 001”, “001”, 45, 80)

   .

   .

   .

   sum ( [ ], 0).

   sum ( [ H/T ], Sum) if

               sum ( T, Sum1).

               Sum = H + Sum1.

В языке имеется возможность работы с динамической базой фактов, в которой м. объединяться как однородные, так и разнородные предикаты-факторы.

База фактов — нечто среднее м-у реляционной СУБД и массивом. В момент активизации все факты базы переносятся в ОЗУ с внешнего зап. устройства. Для работы с БД создается новый раздел программы, в кот. определяются прототипы предикатов-фактов, объедененных в базу.

Database

прототипы в этом разделе описываются по тем же правилам, что и в predicates.

Раздел database записывается перед разделом predicates и предикаты, кот. в нем описываются не могут описываться в разделе predicates, а ис-ся в разделах clouses и goal. Активизация базы происходит в области ОЗУ, кот. по умолчанию имеет тип домена dbasedom. Этот тип программист не указывает явно в программе в разделе domains, но м. его использовать в качестве аргументов встроенных предикатов языка, что позволяет сократить текст программы при работе с базой.

domains

   dbasedom = tms(Post, Kod, Kol, Cena)

   .

   .

   .

   database

   .

   .

               tmc (Post, Kod, Kol, Cena)

Возможно дополнение базы новыми фактами, удаление устаревших, корректировка отдельных фактов. Все операции в базе фактов производятся с помощью стандартных встроенных предикатов:

asserta

               ( < факт >) ( dbasedom) : (i)

assertz

— используется для добавления нового факта в базу. Факт д.б. обязательно определен и относится к области dbasedom. При использовании предиката asserta факт добавляется перед остальными фактами имеющегося предиката.

assertz — добавление после — “ —

retraсt(< факт>)( dbasedom) : (i) — удаление из БД первого факта, кот. сопоставляется с указанным в retraсt фактом. Возможно удаление группы. При этом в факте указывается общее для удаления фактов значение, а на месте всех остальных переменных записываются те переменные, кот. нет в этом предикате.

SAVE(< имя файла >) — сокращает все факты дин. базы из ОЗУ на магн. диск под именем

   string : (i)         заданным в предикате save/

CONSULT(< имя файла >) — добавляет в опер. дин. базу все факты из файла на диске с

   string : (i)                     указанным именем.

**2. Приемы работы с динамической базой фактов.**

1.Перезапись фактов из раздела clouses в базу на МД.

domains

   Post, Kod = string

   Kol, Cena = integer

database

   tmc ( Post, Kod, Kol, Cena)

predicates

   perezap

dauses

   tmc (“001”, “001”, 45,80).

   .

   .

   .

perzap if

save (“data”)

gocel

   perezap.

2. Создание базы в процессе диалога

domains

   Post, Kod = string

   Kol, Cena = integer

   Pc = integer                                         &&признак окончания ввода

database

   tmc ( Post, Kod, Kol, Cena)               &&база создается первоначально

predicates

   vvod

clouses

   vvod if

               write (“Введите код поставщика”),

               nl,

               readln (post),

               nl,

               write (“Введите код ТМЦ”),

               nl,

               readln (Kod),

               nl,                                                                                                                                            write (“Введите кол-во ТМЦ”),

               nl,

               readint (Kol),

               write (“Введите цену”),

               nl,

               readint (Cena),

               assertz (tmc ( Post, Kod, Kol, Cena)),

               write ( “ Введите признак продолжения s/0”),

               pr=0,

               vvod.

goal

   vvod,

   save (“data”)

3.Дополнение базы новыми фактами

.

.

.

goal

   consult(“data”),

   vvod,

   save(“data’).

4.Активизация фактов из файла на диск

domaines

   Post, Kod = string

   Kol, Cena = integer

   Sum, Sum1 = integer

   Sp = kol\*

database

   tmc (Post, Kod, Kol, Cena)

predicates

   sum( Sp, Sum)

clouses

   sum ( [ ], 0).

   sum ( [ H/t ], Sum) if

               sum(T,Sum1),

               Sum= H + Sum1.

goal

   consult (“data”),

   write (“Введите код ТМЦ”),

   nl,

   readln (Kod),

   sum(Sp, Sum),

   write (Sum).

5.Удаление конкретных фактов из БД

domains

   Post, Kod = string

   Kol, Cena = integer

database

   tmc (Post, Kod, Kol, Cena)

predicates

   udal

clouses

   udal if

   write (“Укажите код поставщика”),

   nl,

   readln (Post),

   write (“Укажите код ТМЦ”),

   nl,

   readln (Kod),

   retract (tmc (Post, Kod, Kol, Cena)),

goal

   consult (data),

   udal,

   save(“data’).

6.Удаление группы факторов

dauses

   udal if

   write (“Введите код поставщика”),

               nl,

               readln (post),

   retract(tmc (Post, Kod, Kol, Cena)),

   .

   .

   .

С помощью retract производится удаление дин. базы фактов из ОЗУ.

Корректировка содержимого факта.

Операции проихзводятся в 2-а этапа:

1-- с помощью retract осущ. удаление устаревшего факта из базы

2-- assertz — добавление нового факта в базу.

**3. Структуры.**

СТР-РА — набор объектов, логически связанных между собой в процессе решения задачи и объединенных под одним именем. Стр-ра в П. используется при создании сложной базы фактов и правил.

Если объекты стр-ры относятся к одному типу доменов. то стр-ра наз. однодоменной. Допускается исп-е доменов разного типа, в этом случае стр-ра наз. разнодоменной.

Исп-е стр-ры в программах позволяет упорядочить базу, разрешается обращение к стр-рам по имени.