**Программирование на языке Пролог**

SWI-Prolog является свободно распространяемой реализацией (диалектом) языка программирования Пролог, которая практически полностью соответствует стандарту ISO/IEC 13211-1 «Information technology - Programming languages - Prolog - Part 1: General core» («Информационные технологии. Языки программирования. Пролог. Часть 1. Общее ядро»).

SWI-Prolog позволяет разрабатывать приложения любой направленности, включая Web-приложения и параллельные вычисления, но основным направлением использования является разработка экспертных систем, программ обработки естественного языка, обучающих программ, интеллектуальных игр и т.п.

Программирование в SWI-Prolog возможно в разных вариантах:

- с помощью стандартной offline-среды программирования (<http://www.swi-prolog.org>);

- с помощью offline-среды программирования SWI-Prolog-Editor (<http://lakk.bildung.hessen.de/netzwerk/faecher/informatik/swiprolog/indexe.html>);

- с помощью стандартной online-среды программирования SWI-Prolog (<http://swish.swi-prolog.org>).

Выполнение лабораторных работ в SWI-Prolog рассматривается на примере online-среды.

**1. Освоение основ программирования на языке Пролог и online-среды программирования SWI-Prolog.**

Для начала работы необходимо открыть в браузере страницу online-среды программирования SWI-Prolog (<http://swish.swi-prolog.org>).

Для перехода в режим редактирования и исполнения программ необходимо нажать на кнопку «Program».

В левой панели осуществляется редактирование программы, содержащей факты и правила.

В правой нижней панели выполняется набор вопросов и запуск их на исполнение с помощью кнопки «Run!».

В правой верхней панели интерпретатор SWI-Prolog выдает ответы на вопросы. В случае если на вопрос может быть получено более одного ответа, с помощью кнопок «Next», «10», «100» и «1,000» можно вывести на панель дополнительные ответы.

Программа на Прологе состоит из предложений (утверждений). Каждое предложение заканчивается точкой.

Предложение имеет вид

***B :- A1, ... , An.***

***B*** называется заголовком или головой предложения, а ***A1, ..., An*** - телом. В предложении ключевой логической операцией является перевернутая импликация (***B :- A*** эквивалентно ***B ← A***, «B следует из A») или фразой Хорна. Символ «:-» означает «следует из», символ «,» – конъюнкция (логическое И, ∧).

При необходимости применения дизъюнкции (логическое ИЛИ, ∨) используется символ «;», действующий до следующей дизъюнкции, окончания правила или закрывающей круглой скобки. Например,

***(A1 ∧ ... ∧ An) ∨ (B1 ∧ ... ∧ Bm) → C*** эквивалентно ***C :- A1, ... , An; B1, ... , Bm.***

Предложения бывают трех видов: факты, правила, вопросы.

А) Факт – это предложение, у которого нет тела. В терминах логики предикатов, факт – это и есть предикат. Он фиксирует (определяет) некоторое отношение между объектами. Например, факт, что Наташа является мамой Даши, может быть записан в виде (в SWI-Prolog строки-константы записываются в одинарных кавычках):

***mother('Наташа', 'Даша').***

Факт представляет собой безусловно истинное утверждение. Если воспользоваться нормальной формой Бэкуса-Науэра, то предикат можно определить следующим образом:

<Предикат> ::= <Имя> | <Имя>(<аргумент> [,<аргумент>]\*),

т.е. предикат состоит либо только из имени, либо из имени и следующей за ним последовательности аргументов (термов), заключенной в скобки.

Аргументом (термом) предиката может быть константа, переменная или составной объект (список или функция). Число аргументов предиката называется арностью.

Б) Правило – предложение, истинность заголовка которого в виде предиката зависит от истинности одной или нескольких формул, указанных в теле. Обычно правило содержит несколько хвостовых целей, которые должны быть истинными для того, чтобы само правило было истинным. В нотации БНФ правило будет иметь вид:

<Правило> ::= <предикат> :- <предикат> [,<предикат>]\*.

Например. Известно, что бабушка человека - это мама его мамы или мама его папы. Соответствующие правила будут иметь вид:

***grandmother(X, Y):-***

***mother(X, Z), mother(Z, Y).***

***grandmother(X, Y):-***

***mother(X, Z), father(Z, Y).***

или

***grandmother(X, Y):-***

***mother(X, Z), mother(Z, Y);***

***mother(X, Z), father(Z, Y).***

или

***grandmother(X, Y):-***

***mother(X, Z), (mother(Z, Y); father(Z, Y)).***

В) Вопрос (запрос, цель) – предложение, состоящее только из тела. В нотации БНФ вопрос имеет вид:

<Вопрос> ::= <Предикат> [,<Предикат>]\*

Вопросы используют для выяснения выполнимости некоторого отношения между описанными в программе объектами. Автоматическая система логического вывода Пролога рассматривает вопрос как цель, к которой надо стремиться. Ответ на вопрос может оказаться положительным (true) или отрицательным (false), в зависимости от того, может ли быть достигнута соответствующая цель.

Программа может содержать вопрос в теле (внутренняя цель). Если программа содержит внутреннюю цель, то после запуска программы на выполнение система сразу проверяет достижимость заданной цели. Если внутренней цели в программе нет, то после запуска программы система выдает приглашение вводить вопросы в диалоговом режиме (внешняя цель). Программа, компилируемая в исполняемый файл, обязательно должна иметь внутреннюю цель.

Если цель достигнута, система отвечает «true», в противном случае «false». Следует отметить, что ответ «false» на вопрос не всегда означает, что он отрицательный. Система может дать такой ответ и в том случае, когда у нее просто недостаточно информации, позволяющей положительно ответить на вопрос. Т.е. Пролог основан на т.н. «модели закрытого мира», в которой все, что можно получить на основе описания модели является истиной, а остальное – ложью.

**Переменные**

Во всех предложениях можно использовать переменные. Считается, что переменные в теле одного правила неявно связаны квантором всеобщности. Имя переменной в Прологе может состоять из букв латинского алфавита, цифр, знаков подчеркивания и должно начинаться с прописной буквы или знака подчеркивания. При этом переменные в теле правила неявно связаны квантором всеобщности и эквивалентны объектам предметной области. Переменные могут быть свободными или связанными.

Свободная переменная – переменная, которая еще не получила значения. Она не равняется ни нулю, ни пробелу; у нее вообще нет никакого значения. Такие переменные еще называют неконкретизированными.

Переменная, которая получила какое-то значение, называется связанной. Такой переменной не может быть присвоено новое значение, т. е., по сути, переменная становится константой.

Областью действия переменной в Прологе является одно предложение. В разных предложениях может использоваться одно и то же имя переменной для обозначения разных объектов. Исключением из правила определения области действия является анонимная переменная, которая обозначается символом подчеркивания «\_». Анонимная переменная предписывает интерпретатору (компилятору) проигнорировать значение аргумента (терма). Если в правиле несколько анонимных переменных, то все они отличаются друг от друга, несмотря на то, что записаны с использованием одного и того же символа («\_»). Анонимные переменные могут записываться только в качестве терма предиката. Использовать их в выражениях (например, арифметических) нельзя. Пример использования анонимной переменной в вопросе «Есть ли у Даши мама?»:

***?- mother(\_, 'Даша').***

Ответом программы будет

***true.***

Разновидности предложений Пролога, записанные в виде фраз Хорна (B ← A), можно интерпретировать следующим образом:

- факт: B ← true;

- правило: B ← A;

- вопрос: true ← A.

Таким образом, программа на Прологе состоит из фактов и правил, выражающих некоторые знания о предметной области. Вопрос - это также некоторый предикат, истинность которого нас интересует. Если вопрос не содержит переменных, то вычисление его значения дает ответ «true» при его истинности, либо ответ «false» при его ложности. Если же в предикате вопроса есть переменные, то ищутся их значения (интерпретация), при которых этот предикат и все предикаты программы становятся истинными. В этом и состоит суть вычислений (логического вывода) программы на Прологе.

При поиске решения (доказательства цели) в Прологе используется метод перебора с возвратами (поиск в глубину). Пролог при доказательстве утверждения поочередно пытается установить истинность, входящих в него предикатов (утверждений). Если первый предикат истинен, то Пролог переходит ко второму. Если и он истинен, то переходит к третьему. Если второй предикат ложен, то Пролог пытается установить его истинность при других значениях, входящих в него переменных. Если этого не удается сделать, то он возвращается к первому предикату и пытается установить его истинность для новых значений переменных, а затем снова возвращается к доказательству второго предиката. Такая процедура повторяется до тех пор, пока не будет достигнута истинность последнего предиката. После доказательства истинности последнего предиката цели Пролог завершает работу. Процесс возврата в Прологе называется backtracking.

Пример программы:

Задайте в программе следующие факты (предикаты) и правила:

mother('Наташа', 'Даша').

mother('Даша', 'Маша').

mother('Наташа', 'Вася').

father('Вася', 'Маша').

grandmother(X, Y):-

mother(X, Z), (mother(Z, Y); father(Z, Y)).

grandfather(X, Y):-

father(X, Z), (mother(Z, Y); father(Z, Y)).

Сформируйте следующие вопросы:

Вопрос 1 – является ли Наташа мамой Даши?

Вопрос 2 – кто является мамой Даши?

Вопрос 3 – есть ли у Даши мама?

Вопрос 4 – найти всех мам и детей.

Вопрос 5 – для кого Наташа является бабушкой?

Некоторые операции и предикаты SWI-Prolog

|  |  |
| --- | --- |
| Операция / Предикат | Назначение |
| true | Истина |
| fail, false | Ложь |
| = | Для переменной, стоящей слева от операции: - свободной - присваивание без преобразования (вычисления) выражения справа от операции; - связанной - сравнение без преобразования (вычисления) выражения справа от операции. |
| <, =<, >=, > | Арифметические (только для чисел) операции сравнения |
| =:= | Арифметическое равенство |
| =\= | Арифметическое неравенство |
| is | Для переменной, стоящей слева от операции: - свободной - присваивание с преобразованием (вычислением) выражения справа от операции; - связанной - сравнение с преобразованием (вычислением) выражения справа от операции. |
| @<, @=<, @>=, @> | Операции сравнения для констант и переменных любого типа (чисел, строк, списков и т.д.) |
| == | Равенство констант и переменных любого типа |
| \== | Неравенство констант и переменных любого типа |
| not(A) | Отрицание логического выражения A |
| read(A) | Чтение значения с клавиатуры и присваивание его переменной А |
| write(A) | Печать А на экран с установкой курсора после последнего напечатанного символа |
| writeln(A) | Печать А на экран с переводом курсора в начало следующей строки |
| nl | Перевод курсора в начало следующей строки |
| repeat | Предикат, выдающий новое истинное значение при возврате. Передоказываемый предикат |
| ! | Предикат (cut, сократить), запрещающий возврат далее той точки, где он стоит |
| assert(A), assertz(A) | Динамическое добавление факта (правила) в конец списка подобных фактов (правил) базы знаний (программы) |
| asserta(A) | Динамическое добавление факта (правила) в начало списка подобных фактов (правил) базы знаний |
| retract(A) | Удаление первого факта (правила) базы знаний |
| retractall(A) | Удаление всех фактов (правил) базы знаний с именем А |

**2. Рекурсия**

Рекурсия – процесс повторения элементов самоподобным образом. Рекурсия (в программировании) – алгоритмический метод, заключающийся в возможности обращения правила (функции, процедуры) к самому себе один или более раз.

В виду отсутствия в Прологе операторов цикла, рекурсия является часто используемым приемом в программах на этом языке. Любая рекурсивная процедура в Прологе должна включать, как минимум, два правила:

1) нерекурсивное правило, определяющее его вид в момент прекращения рекурсии;

2) рекурсивное правило, где первая подцель вырабатывает новые значения аргументов, а вторая – вызов самого правила с новыми значениями аргументов.

Для примера базы знаний «Родственники» введем правило, описывающее отношение «предок» - «потомок» с помощью рекурсии.

% 1 параметр - имя отца или матери

% 2 параметр - имя сына или дочери

roditel('Вася', 'Вика').

roditel('Вася', 'Коля').

roditel('Коля', 'Юля').

roditel('Юля', 'Миша').

roditel('Юля', 'Маша').

roditel('Коля', 'Света').

roditel('Света', 'Рома').

roditel('Света', 'Леша').

% 1 параметр - предок

% 2 параметр - потомок

predok(A, B) :- roditel(A, B).

predok(A, B) :- roditel(A, C), predok(C, B).

Здесь первое (нерекурсивное) правило предиката predok предписывает Прологу завершить процедуру поиска предка (потомка), если первый терм является непосредственным родителем ребенка, указанного вторым термом. В противном случае срабатывает второе (рекурсивное) правило, предписывающее в процессе поиска связи между предполагаемыми предком и потомком добавить промежуточное поколение.

В Прологе отсутствуют конструкции циклов с параметром, пред- и постусловием, но с помощью соответствующих механизмов можно организовать разные типы циклов.

1 способ. Используя рекурсию.

2 способ. Используя предикат, который можно передоказать, и предикат fail. Например, выяснить всех правнуков «Коли».

?- roditel('Коля', A),

write('Правнуки ('), write(A), writeln('):'),

roditel(A, B),

writeln(B),

fail;

true.

3 способ. Используя предикат, который можно передоказать, и логическое утверждение. Например, выдать пары «предок» - «потомок» до тех пор, пока не встретится потомок с именем «Миша».

?- predok(A, B),

write(A), write(' является предком '), writeln(B),

B == 'Миша';

true.

Такая конструкция соответствует циклу с постусловием.

4 способ. Организация цикла со счетчиком, используя предикат repeat и динамическое добавление фактов в базу знаний (программу).

Следует отметить, что в некоторых реализациях языка Пролог отсутствует встроенный предикат repeat. Тогда данный предикат надо определить в программе следующим образом

repeat.

repeat:- repeat.

Его используют вместо передоказываемого предиката. Но если при передоказательстве обычных предикатов может наступить момент, что все факты исчерпаны (например, predok(A, ‘Миша’) ... fail), то при использовании предиката repeat такой момент не наступает никогда.

?- assert(count(1)),

repeat,

retract(count(X)),

X2 is X \* X,

write(X), write('^2 = '), writeln(X2),

X1 is X + 1,

(X1 > 10, !;

assert(count(X1)), fail).

Приведенная программа будет циклически выдавать на экран квадраты чисел от 1 до 10. В качестве счетчика используется предикат (факт) count, который динамически добавляется и удаляется из базы знаний.

Второй пример использования рекурсии – расчет факториала.

factorial(1, 1):- !. % факториал единицы равен единице (можно начать с 0)

factorial(N, F):-

N1 is N - 1,

factorial(N1, F1), % F1 равен факториалу числа на единицу меньшего N

F is F1 \* N. % факториал числа N равен произведению F1 на само число N

Здесь первое (нерекурсивное) правило предиката factorial определяет для числа «1» значение факториала, равное «1». В противном случае от числа указанного в качестве первого терма N необходимо отнять единицу [N1 is N – 1], найти значение факториала F1 для нового числа N1 [factorial(N1, F1)], определить факториал для исходного числа N [F is F1 \* N] и установить значение F в качестве второго терма.

При необходимости вычислений в программе следует учитывать разницу и особенности применения операций «=» и «is». Первая операция не приводит к вычислению (преобразованию) выражения, записанного справа от операции, а вторая – приводит. В случае использования свободной переменной операции приводят к присваиванию исходного (преобразованного) выражения, записанного справа от операции. В случае использования связанной переменной выполняется сравнение левой и правой частей.

**3. Списки**

В Прологе нет такой распространенной и часто используемой структуры хранения данных как массивы, но зато есть развитые возможности работы со списками. Список – упорядоченный набор элементов одного типа. В отличие от массивов, количество элементов которых строго фиксировано (в большинстве языков программирования), списки позволяют модифицировать, добавлять или удалять из него элементы.

Списки в Прологе заключаются в квадратные скобки, например [1, 2, 8, 123] или ['Пн', 'Вт', 'Четверг']. Список, не содержащий ни одного элемента «[]», называется пустым. Каждый непустой список состоит из двух частей: головы и хвоста. Головой является первый элемент списка, хвостом – все остальное. В программе голова отделяется от хвоста символом «|».

Часто используемыми операциями при работе со списками являются:

1. Проверка наличия элемента в списке.

% 1 параметр – элемент, наличие которого в списке требуется проверить

% 2 параметр - список

member(X, [X |\_]).

member(X, [\_|Tail]):- member(X, Tail).

2. Добавление элемента в список. Для данной операции не требуется отдельного правила, если элемент X добавляется в начало списка

NewList = [X| List].

3. Конкатенция двух списков.

% 1 параметр – первый список

% 2 параметр – второй список

% 3 параметр – результат объединения двух списков

concat([], L2, L2).

concat([X|L1], L2, [X|L3]):- concat(L1, L2, L3).

4. Удаление элемента из списка и задание обратного порядка следования элементов списка.

4.1 Удаление элемента из списка (результатом будет перевернутый список без элемента)

% 1 параметр – удаляемый элемент

% 2 параметр – исходный список

% 3 параметр – рабочий список

% 4 параметр – перевернутый список без элемента

delete(\_, [], L, L).

delete(X, [X|L], L1, L2):- delete(X, L, L1, L2).

delete(X, [Y|L], L1, L2):- X \== Y, delete(X, L, [Y|L1], L2).

4.2 Задание обратного порядка следования элементов списка

% 1 параметр – исходный список

% 2 параметр – рабочий список

% 3 параметр – перевернутый список

reverse([], Lr, Lr).

reverse([X|L], L1, Lr):- reverse(L, [X|L1], Lr).

5. Разделение списка на два.

% 1 параметр – элемент, задающий разбиение

% 2 параметр – исходный список

% 3 параметр – элементы, меньшие или равные 1 параметру

% 4 параметр – элементы, большие 1 параметра

split(\_, [], [], []).

split(Y, [X|L], [X|L1], L2):- X =@< Y, split(Y, L, L1, L2).

split(Y, [X|L], L1, [X|L2]):- X @> Y, split(Y, L, L1, L2).

**Лабораторная работа №1**

Задача:

- для заданной предметной области описать набор фактов, характеризующих информационные единицы данной предметной области в соответствии с приведенной в задании структурой (описать не менее 5 информационных единиц предметной области)

- разработать набор правил, позволяющих реализовать все типы запросов, приведенные в варианте задания.

- сформулировать на языке Пролог приведенные в задании запросы.

**Варианты индивидуальных заданий для реализации в среде**

**SWI-Prolog и Visual Prolog**

Вариант № 1. Предметная область – семья. Каждая семья может быть описана структурой из трех компонент: мужа, жены и детей. Каждый член семьи может быть описан структурой: имя, отчество, фамилия, год рождения, пол, ежемесячный доход. Для детей добавить поле «близнец».

Реализовать следующие типы запросов:

1. Проверить, существует ли в БД заданный человек (по ФИО);
2. Найти всех работающих детей;
3. Найти всех работающих мужей, чей доход больше чем у жены;
4. Найти всех людей, которые не работают и родились до указанного года;
5. Найти число семей, у которых есть близнецы.

Вариант № 2. Предметная область – библиотека. Каждая книга может быть описана структурой: название, автор, издание. Автор может быть описан структурой: имя, фамилия, год рождения. Издание может быть описано структурой: издательство, номер издания, год издания, количество страниц, цена, гонорар автора.

Реализовать следующие типы запросов:

1. Найти автора, у которого книга переиздавалась максимальное число раз;
2. Найти все книги, изданные более одного раза;
3. Найти все книги, изданные в одном издательстве в заданном году;
4. Найти все книги одного автора;
5. Найти все книги, цена которых превышает заданную сумму.

Вариант № 3. Предметная область – страны мира. Каждая страна может быть описана структурой: название, площадь, географическое положение, население. Географическое положение может быть описана структурой: часть света, материк, океаны, моря, горные хребты. Население может быть описано структурой: численность, государственный язык, национальный состав. Национальный состав может быть описан структурой: национальность, численность, процент от всего населения.

Реализовать следующие типы запросов:

1. Найти страну, у которой максимальная численность населения;
2. Найти все страны, находящиеся на указанном материке с населением больше заданной величины;
3. Найти все страны, у которых однородный национальный состав (численность основной национальности более 90%);
4. Найти все страны, имеющие выход к указанному морю;
5. Найти все страны с указанным государственным языком.

Вариант № 4. Предметная область – биржа труда. Каждая вакансия может быть описана структурой: название предприятия, должность, ежемесячный доход, требования к соискателю. Требования к соискателю могут быть описаны структурой: образование, возраст, пол, дополнительные условия. Образование может быть описано структурой: уровень образования (высшее техническое, высшее экономическое, среднее, среднее специальное и т.д.), специальность. Дополнительные условия могут быть описаны структурой: список иностранных языков, которыми должен владеть соискатель, умение работать на ПК, стаж работы по специальности.

Реализовать следующие типы запросов:

1. Найти все должности, для которых возраст соискателей, не превышает указанный;
2. Найти все вакансии для мужчин, с высшим техническим образованием и свободно владеющих более чем одним иностранным языком;
3. Найти все вакансии для соискателей с высшим гуманитарным образованием;
4. Найти все вакансии для женщин, умеющих работать на ПК, имеющим стаж работы более 5 лет;
5. Найти все предприятия, которые берут на работу женщин.

Вариант № 5. Предметная область – служба знакомств. Каждый клиент может быть описан структурой: фамилия, имя, отчество, возраст, национальность, образование, ежемесячный доход, дополнительные условия, требования к партнеру. Дополнительные условия могут быть описаны структурой: владение жилой площадью, наличие детей, отсутствие вредных привычек. Требования к партнеру могут быть описаны структурой: образование, возраст, национальность, ежемесячный доход, дополнительные условия.

Реализовать следующие типы запросов:

1. Найти всех клиентов, которым необходим партнер без вредных привычек;
2. Найти всех клиентов, с указанным уровнем образования, имеющих жилую площадь и не старше указанного возраста;
3. Найти всех клиентов с указанной национальностью;
4. Найти всех клиентов, которым необходим партнер, не старше указанного возраста и не имеющий детей;
5. Найти самого старого клиента службы знакомств.

Вариант № 6. Предметная область – спортивные соревнования. Каждое соревнование может быть описано структурой: вид соревнования, команды – участники. Вид соревнования может быть описан структурой: ранг соревнований (чемпионат Европы, чемпионат мира, Олимпийские игры), вид спорта, год проведения, страна проведения. Команды – участники могут быть описаны списком из следующих структур: название команды, страна, результаты соревнований. Результаты соревнований могут быть описаны списком структур: название команды – соперника, тип результата (выигрыш, проигрыш, ничья).

Реализовать следующие типы запросов:

1. Найти все команды указанного вида спорта, у которых не было ни одного проигрыша за указанный период времени;
2. Найти все страны, где проводились Олимпийские игры до указанного года;
3. Найти всех соперников указанной команды в соревнованиях заданного ранга;
4. Найти вид соревнования, в котором участвовало наибольшее число команд;
5. Найти все команды указанной страны.

Вариант № 7. Предметная область – видеотека. Каждая видеокассета может быть описана структурой: название фильма, год создания, киностудия, атрибуты фильма. Атрибуты фильма могут быть описаны структурой: автор сценария, режиссер, исполнители главных ролей, премии. Исполнители главных ролей могут быть описаны списком из следующих структур: фамилия, роль. Премии могут быть описаны списком из следующих структур: название фестиваля, год проведения.

Реализовать следующие типы запросов:

1. Подсчитать число фильмов указанного режиссера;
2. Найти все фильмы, получившие премии на определенном фестивале;
3. Найти всех режиссеров, фильмы которых создавались на одной киностудии;
4. Найти все роли, определенного актера, которые он сыграл в фильмах, которые получили какие-либо премии;
5. Найти всех сценаристов, в фильмах которых снимался определенный актер.

Вариант № 8. Предметная область – учебная группа факультета. Каждая учебная группа может быть описана структурой: название факультета, код специальности, номер группы, состав группы. Состав группы может быть описан списком структур, описывающих отдельного студента: фамилия, имя, отчество, пол, год рождения, обучение на военной кафедре, сводная ведомость. Сводная ведомость может быть описана списком из следующих структур: предмет, оценка.

Реализовать следующие типы запросов:

1. Подсчитать число групп на указанном факультете;
2. Найти всех студенток, обучающихся по указанной специальности;
3. Найти группу, которая сдала больше всего предметов сессию;
4. Найти всех студентов, имеющих задолженности;
5. Найти всех студентов, обучающихся на военной кафедре.

Вариант № 9. Предметная область – семья. Каждая семья может быть описана структурой из трех компонент: мужа, жены и детей. Каждый член семьи может быть описан структурой: имя, отчество, фамилия, год рождения, пол, ежемесячный доход. Для детей добавить поле «близнец».

Реализовать следующие типы запросов:

1. Найти всех близнецов;

1. Найти всех детей, родившихся в заданном году;
2. Найти всех работающих жен, чей доход больше заданной суммы;
3. Найти фамилии людей, у которых есть заданное число детей.
4. Найти всех людей, у которых есть только один ребенок.

Вариант № 10. Предметная область – библиотека. Каждая книга может быть описана структурой: название, автор, издание. Автор может быть описан структурой: имя, фамилия, год рождения. Издание может быть описано структурой: издательство, номер издания, год издания, количество страниц, цена, гонорар автора.

Реализовать следующие типы запросов:

1. Найти автора, у которого максимальный гонорар за издание книги;
2. Найти все книги, изданные только один раз;
3. Найти всех авторов, родившихся позже указанного года;
4. Найти все издательства, в которых была издана указанная книга;
5. Найти все книги, цена которых не изменялась в разных изданиях.

Вариант № 11. Предметная область – страны мира. Каждая страна может быть описана структурой: название, площадь, географическое положение, население. Географическое положение может быть описана структурой: часть света, материк, океаны, моря, горные хребты. Население может быть описано структурой: численность, государственный язык, национальный состав. Национальный состав может быть описан структурой: национальность, численность, процент от всего населения.

Реализовать следующие типы запросов:

1. Найти страну, которую омывает больше всего морей;
2. Найти все страны, на территории которых находится указанный горный хребет;
3. Найти все страны, у которых число национальностей превышает заданную величину;
4. Найти все горные хребты, находящиеся на территории указанной страны;
5. Найти все страны, у которых численность населения меньше заданной величины.

Вариант № 12. Предметная область – биржа труда. Каждая вакансия может быть описана структурой: название предприятия, должность, ежемесячный доход, требования к соискателю. Требования к соискателю могут быть описаны структурой: образование, возраст, пол, дополнительные условия. Образование может быть описано структурой: уровень образования (высшее техническое, высшее экономическое, среднее, среднее специальное и т.д.), специальность. Дополнительные условия могут быть описаны структурой: список иностранных языков, которыми должен владеть соискатель, умение работать на ПК, стаж работы по специальности.

Реализовать следующие типы запросов:

1. Найти все должности, для которых подходят соискатели со средним специальным образованием;
2. Найти предприятие, у которого больше всего вакансий;
3. Найти все вакансии для соискателей, указанной специальности;
4. Найти все вакансии для мужчин с ежемесячным доходом выше указанного значения;
5. Найти все должности, для которых не требуется знания иностранного языка.

Вариант № 13. Предметная область – служба знакомств. Каждый клиент может быть описан структурой: фамилия, имя, отчество, возраст, национальность, образование, ежемесячный доход, дополнительные условия, требования к партнеру. Дополнительные условия могут быть описаны структурой: владение жилой площадью, наличие детей, отсутствие вредных привычек. Требования к партнеру могут быть описаны структурой: образование, возраст, национальность, ежемесячный доход, дополнительные условия.

Реализовать следующие типы запросов:

1. Найти всех клиентов, для которых подходит партнер с высшим образованием и с указанным уровнем дохода;
2. Найти всех клиентов, у которых не указаны дополнительные условия;
3. Найти всех клиентов младше указанного возраста;
4. Найти клиента, которому требуется самый молодой партнер;
5. Найти всех клиентов, у которых нет детей.

Вариант № 14. Предметная область – спортивные соревнования. Каждое соревнование может быть описано структурой: вид соревнования, команды – участники. Вид соревнования может быть описан структурой: ранг соревнований (чемпионат Европы, чемпионат мира, Олимпийские игры), вид спорта, год проведения, страна проведения. Команды – участники могут быть описаны списком из следующих структур: название команды, страна, результаты соревнований. Результаты соревнований могут быть описаны списком структур: название команды – соперника, тип результата (выигрыш, проигрыш, ничья).

Реализовать следующие типы запросов:

1. Найти страну, где проводилось максимальное число соревнований за указанный период;
2. Найти все страны, где проводились чемпионаты мира по указанному виду спорта.
3. Найти всех соперников указанной команды в соревнованиях в заданном году;
4. Найти все команды, участвовавшие в указанных соревнованиях в заданной стране;
5. Найти все команды, у которых не было ничьих.

Вариант № 15. Предметная область – видеотека. Каждая видеокассета может быть описана структурой: название фильма, год создания, киностудия, атрибуты фильма. Атрибуты фильма могут быть описаны структурой: автор сценария, режиссер, исполнители главных ролей, премии. Исполнители главных ролей могут быть описаны списком из следующих структур: фамилия, роль. Премии могут быть описаны списком из следующих структур: название фестиваля, год проведения.

Реализовать следующие типы запросов:

1. Найти режиссеров и сценаристов, у которых все фильмы получили премии;
2. Найти сценаристов, которые работали с указанным режиссером более одного раза;
3. Найти все фильмы указанного сценариста после указанного года;
4. Найти все фильмы, получившие премии в указанном году;
5. Найти все киностудии, которые работали с указанным режиссером.

Вариант № 16. Предметная область – учебная группа факультета. Каждая учебная группа может быть описана структурой: название факультета, код специальности, номер группы, состав группы. Состав группы может быть описан списком структур, описывающих отдельного студента: фамилия, имя, отчество, пол, год рождения, обучение на военной кафедре, сводная ведомость. Сводная ведомость может быть описана списком из следующих структур: предмет, оценка.

Реализовать следующие типы запросов:

1. Подсчитать общее число студентов на указанном факультете;
2. Найти группу, у которой больше всего отличников;
3. Найти все предметы в группах указанной специальности, по которым сдавался зачет;
4. Найти всех студентов мужского пола, не обучающихся на военной кафедре.
5. Найти всех студентов, старше заданного года рождения;

Вариант № 17. Предметная область – семья. Каждая семья может быть описана структурой из трех компонент: мужа, жены и детей. Каждый член семьи может быть описан структурой: имя, отчество, фамилия, год рождения, пол, ежемесячный доход. Для детей добавить поле «близнец».

Реализовать следующие типы запросов:

1. Найти всех людей, чей доход меньше заданного;
2. Найти всех детей, младше заданного возраста;
3. Найти всех неработающих жен, которые родились позже заданного года;
4. Найти всех детей, у которых разница в возрасте родителей превышает заданную величину;
5. Подсчитать количество семей, у которых нет близнецов.

Вариант № 18. Предметная область – библиотека. Каждая книга может быть описана структурой: название, автор, издание. Автор может быть описан структурой: имя, фамилия, год рождения. Издание может быть описано структурой: издательство, номер издания, год издания, количество страниц, цена, гонорар автора.

Реализовать следующие типы запросов:

1. Найти книгу, у которой минимальная цена;
2. Найти все книги, изданные в одном издательстве;
3. Найти всех авторов, книги которых издавались только один раз;
4. Найти все книги указанного автора, у которых число страниц больше заданной величины;
5. Найти все издательства, выпускавшие книги до заданного года.

Вариант № 19. Предметная область – страны мира. Каждая страна может быть описана структурой: название, площадь, географическое положение, население. Географическое положение может быть описана структурой: часть света, материк, океаны, моря, горные хребты. Население может быть описано структурой: численность, государственный язык, национальный состав. Национальный состав может быть описан структурой: национальность, численность, процент от всего населения.

Реализовать следующие типы запросов:

1. Найти страну, у которой максимальная плотность населения;
2. Найти все моря, которые омывают территорию указанной страны;
3. Найти страну, у которой численность ни одной из национальностей не превышает 50 %;
4. Найти все страны, имеющие выход к указанному океану;
5. Найти все страны, у которых название части света совпадает с названием материка.

Вариант № 20. Предметная область – биржа труда. Каждая вакансия может быть описана структурой: название предприятия, должность, ежемесячный доход, требования к соискателю. Требования к соискателю могут быть описаны структурой: образование, возраст, пол, дополнительные условия. Образование может быть описано структурой: уровень образования (высшее техническое, высшее экономическое, среднее, среднее специальное и т.д.), специальность. Дополнительные условия могут быть описаны структурой: список иностранных языков, которыми должен владеть соискатель, умение работать на ПК, стаж работы по специальности.

Реализовать следующие типы запросов:

1. Найти должность, у которой минимальный ежемесячный доход;
2. Найти все вакансии для мужчин, с указанным уровнем образования, владеющих хотя бы одним иностранным языком;
3. Найти все вакансии для женщин с указанным уровнем образованием;
4. Найти все вакансии для женщин, не старше указанного возраста;
5. Найти все должности, для которых требуется знание хотя бы одного иностранного языка.

Вариант № 21. Предметная область – служба знакомств. Каждый клиент может быть описан структурой: фамилия, имя, отчество, возраст, национальность, образование, ежемесячный доход, дополнительные условия, требования к партнеру. Дополнительные условия могут быть описаны структурой: владение жилой площадью, наличие детей, отсутствие вредных привычек. Требования к партнеру могут быть описаны структурой: образование, возраст, национальность, ежемесячный доход, дополнительные условия.

Реализовать следующие типы запросов:

1. Найти всех клиентов, для которых подходит партнер со средним образованием, владеющим жилой площадью и не имеющим детей;
2. Найти клиента, с максимальной разницей в возрасте между ним и партнером;
3. Найти всех клиентов указанной национальности, не старше указанного возраста;
4. Найти всех клиентов, которым необходим партнер без вредных привычек;
5. Найти всех клиентов, у которых нет детей, и которым подходит партнер, имеющий детей.

Вариант № 22. Предметная область – спортивные соревнования. Каждое соревнование может быть описано структурой: вид соревнования, команды – участники. Вид соревнования может быть описан структурой: ранг соревнований (чемпионат Европы, чемпионат мира, Олимпийские игры), вид спорта, год проведения, страна проведения. Команды – участники могут быть описаны списком из следующих структур: название команды, страна, результаты соревнований. Результаты соревнований могут быть описаны списком структур: название команды – соперника, тип результата (выигрыш, проигрыш, ничья).

Реализовать следующие типы запросов:

1. Найти команду, которая провела больше всего игр в указанных соревнованиях;
2. Найти все страны, где проводились чемпионаты Европы по определенному виду спорта до указанного года;
3. Найти все команды, которые участвовали в соревнованиях заданного ранга по определенному виду спорта;
4. Найти все команды, участвовавшие в соревнованиях в заданном году;
5. Найти все команды определенной страны, у которых не было выигрышей.

Вариант № 23. Предметная область – видеотека. Каждая видеокассета может быть описана структурой: название фильма, год создания, киностудия, атрибуты фильма. Атрибуты фильма могут быть описаны структурой: автор сценария, режиссер, исполнители главных ролей, премии. Исполнители главных ролей могут быть описаны списком из следующих структур: фамилия, роль. Премии могут быть описаны списком из следующих структур: название фестиваля, год проведения.

Реализовать следующие типы запросов:

1. Найти режиссера, чьи фильмы получили максимальное число премий;
2. Найти все роли указанного актера;
3. Найти все фильмы, снятые на одной киностудии, одним и тем же режиссером;
4. Найти актеров, снимавшихся на одной киностудии;
5. Найти всех актеров, снимавшихся в фильмах определенного сценариста.

Вариант № 24. Предметная область – учебная группа факультета. Каждая учебная группа может быть описана структурой: название факультета, код специальности, номер группы, состав группы. Состав группы может быть описан списком структур, описывающих отдельного студента: фамилия, имя, отчество, пол, год рождения, обучение на военной кафедре, сводная ведомость. Сводная ведомость может быть описана списком из следующих структур: предмет, оценка.

Реализовать следующие типы запросов:

1. Подсчитать средний балл сессии по факультету;
2. Найти группу, у которой меньше всего задолженностей;
3. Найти все предметы в указанной группе, по которым сдавался экзамен;
4. Найти всех студенток, обучающихся в указанной группе;
5. Найти всех студентов, не обучающихся на военной кафедре.