******

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

**Ознакомление с программированием на языке ПРОЛОГ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ**

**Направление 230100.65**

**«Вычислительные машины, комплексы, системы и сети»,**

**4 курс, дневное отделение**

**Киров, 2014**

УДК ***Цель выполнения лабораторной работы***

В результате выполнения лабораторной работы необходимо:

1. Ознакомиться с принципами логического программирования на языке Турбо-Пролог.

2. Научиться составлять программы на языке Турбо-Пролог.

1. ***Задания на лабораторную работу:***

*Вариант1.* Составить текст программы на языке ПРОЛОГ поиска студента в статической базе по фамилии и возрасту (до 21 года) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фамилии, имени, отчества – string, тип возраста -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант2.* Составить текст программы на языке ПРОЛОГ поиска студента в статической базе по фамилии и возрасту (до 21 года) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фамилии, имени, отчества –symbol, тип возраста -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 3.* Составить текст программы на языке ПРОЛОГ поиска фруктов в статической базе по весу(до 100 грамм), цвету и вкусу без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фруктов – string, тип данных веса-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 4.* Составить текст программы на языке ПРОЛОГ поиска фруктов в статической базе по весу(более 4 кг), цвету и вкусу без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фруктов – symbol, тип данных веса-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант5.* Составить текст программы на языке ПРОЛОГ поиска студента в статической базе по фамилии и возрасту (от 19 до 21 года) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фамилии, имени, отчества – string, тип возраста -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант6.* Составить текст программы на языке ПРОЛОГ поиска студента в статической базе по фамилии и возрасту (от 19 до 21 года) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фамилии, имени, отчества –symbol, тип возраста -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 7.* Составить текст программы на языке ПРОЛОГ поиска фруктов в статической базе по весу(от 100 до 500 грамм), цвету и вкусу без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фруктов – string, тип данных веса-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 8.* Составить текст программы на языке ПРОЛОГ поиска фруктов в статической базе по весу(от 3 до 9 кг), цвету и вкусу без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных фруктов – symbol, тип данных веса-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 9.* Составить текст программы на языке ПРОЛОГ поиска болезни в статической базе по симптому и температуре (от 26 до 38) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных болезни – string, тип данных температуры-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 10.* Составить текст программы на языке ПРОЛОГ поиска болезни в статической базе по симптому и температуре (от 26 до 38) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно Тип данных болезни – symbol, тип данных температуры -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 11.* Составить текст программы на языке ПРОЛОГ поиска марки автомобиля в статической базе по цвету и мощности (более 90 л.с.) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных болезни – string, тип данных температуры-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 12.* Составить текст программы на языке ПРОЛОГ поиска марки автомобиля в статической базе по цвету и мощности (более 90 л.с.) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно Тип данных болезни – symbol, тип данных температуры -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 13.* Составить текст программы на языке ПРОЛОГ поиска болезни в статической базе по симптому и температуре (от 90 до 150 л.с.) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных болезни – string, тип данных температуры-integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 14.* Составить текст программы на языке ПРОЛОГ поиска марки автомобиля в статической базе по цвету и мощности (от 90 до 150 л.с.) без использования списков. Размер базы 7 записей. Результаты поиска вывести в отдельное окно Тип данных болезни – symbol, тип данных температуры -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 15.* Задача “Обезьяна и бананы”.Обезьяна находится в запертой комнате. В комнате на полу лежат 3 коробки, а связка бананов свисает с потолка так, что достать ее обезьяна не может.

Решение задачи – это последовательность действий, которые позволять обезьяне взять бананы. Последовательность действий обезьяны может быть организовано следующим образом: обезьяна строит из коробок лесенку под бананами, обезьяна залезает на коробку и хватает бананы.

Программа должна предусматривать последовательность установки 3-х коробок в виде лесенки. Эту последовательность вывести в отдельное окно.

*Вариант 16.* Написать программу для продажи туристических туров по цене от 10000 до 30000. Должна быть представлена следующая информация: название тура, страна, продолжительность, цена. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных название тура, страна – string, тип данных продолжительность и цена- -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 17.* Написать программу для продажи туристических туров от 10000 до 30000. Должна быть представлена следующая информация: название тура, страна, продолжительность, цена. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных название тура, страна – symbol, тип данных продолжительность и цена- -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 18.* Написать программу для продажи туристических туров от 10000 до 30000 и продолжительности не более 7 дней. Должна быть представлена следующая информация: название тура, страна, продолжительность, цена. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных название тура, страна – string, тип данных продолжительность и цена- -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 19.* Написать программу для продажи туристических туров от 10000 до 30000 и продолжительности от 7 до 10 дней. Должна быть представлена следующая информация: название тура, страна, продолжительность, цена. Размер базы 7 записей. Результаты поиска вывести в отдельное окно. Тип данных название тура, страна – symbol, тип данных продолжительность и цена- -integer. Выполнить запуск программы с внутренней и внешней целью и в режиме трассировки.

*Вариант 20.* Средствами языка Prolog создать программу, которая решает ребус:

1. УЗЕЛ + УЗЕЛ = ПЕТЛЯ

Например: 6425 + 6425 = 12850. Программа должна найти все решения.

*Вариант 21.* Задан неориентированный взвешенный граф, представленный на рисунке 1.



# Рисунок 1 – Неориентированный взвешенный граф

Разработать программу на языке Prolog, которая находит минимальное связывающее дерево на графе с помощью алгоритма Краскала.

В отдельное окно вывести минимальное связывающее дерево в виде перечисления дуг.

Алгоритм Краскала

1. Составить список ребер в порядке увеличения их весов.
2. Выбрать ребро с минимальным весом
3. Добавить к нему следующее ребро с минимальным весом и проверить: не приводит ли добавление этого ребра к образованию цикла. Если образуется цикл, ребро отбросить и выбрать следующее с минимальным весом.
4. Процесс завершить после выбора (n-1)го ребра.

Список ребер в порядке увеличения их весов представлен в таблице 1.

Таблица 1 – Список ребер.

|  |  |
| --- | --- |
| Вес ребра | Ребра |
| 1 | (1,4), (5,6), (5,8) |
| 2 | (2,3), (2,5), (6,7) |
| 3 | (1,2), (2,6), (3,6), (6,8) |
| 4 | (3,4) |
| 5 | (1,3), (4,6) |
| 6 | (4,7) |
| 7 | (7,8) |

Минимальное покрывающее дерево для данного графа, построенное с помощью алгоритма Краскала, представлено на рисунке 2.



# Рисунок 2 – Минимальное покрывающее дерево

1. ***Пример разработки программы***

*ВАРИАНТ №1*

Средствами языка Prolog создать программу, которая решает логическую задачу поиска расстановки 8 ферзей на шахматной доске таким образом, чтобы ни один из ферзей не находился под боем. Расположение ферзей представить в виде списка, например F[[1,4],[2,2],[3,7],[4,3],[5,6],[6,8],[7,5],[8,1]].

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *8* |  |  |  |  |  | *F* |  |  |
| *7* |  |  | *F* |  |  |  |  |  |
| *6* |  |  |  |  | *F* |  |  |  |
| *5* |  |  |  |  |  |  | *F* |  |
| *4* | *F* |  |  |  |  |  |  |  |
| *3* |  |  |  | *F* |  |  |  |  |
| *2* |  | *F* |  |  |  |  |  |  |
| *1* |  |  |  |  |  |  |  | *F* |
|  | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* |

***Текст программы:***

domains

X,Y=integer

Predicates *//раздел служит для описания предикатов, используемых в программе*

win

position

f(X,Y)

goal *//в разделе goal формулируется назначение создаваемой программы. Составными частями при этом могут являться некоторые подцели, из которых формируется единая цель программы;*

win(),

position.

Clauses *// раздел clauses заносятся факты и правила. Факт представляется именем предиката, за которым следуют аргументы, заключенные в круглые скобки. Заканчивается запись факта точкой. Все имена предикатов и константы должны начинаться со строчной буквы. Правило состоит из заголовка правила и тела правила. Заголовок представляет собой предикат, тело состоит из термов, которые могут быть связаны между собой словами or или and. Между телом и заголовком стоит слово if. Каждое правило должно заканчиваться точкой. Имена переменных должны начинаться с прописной буквы и могут содержать только буквы, цифры или знак подчеркивания.*

f(1,1). f(1,2). f(1,3). f(1,4). f(1,5). f(1,6). f(1,7). f(1,8).

f(2,1). f(2,2). f(2,3). f(2,4). f(2,5). f(2,6). f(2,7). f(2,8).

f(3,1). f(3,2). f(3,3). f(3,4). f(3,5). f(3,6). f(3,7). f(3,8).

f(4,1). f(4,2). f(4,3). f(4,4). f(4,5). f(4,6). f(4,7). f(4,8).

f(5,1). f(5,2). f(5,3). f(5,4). f(5,5). f(5,6). f(5,7). f(5,8).

f(6,1). f(6,2). f(6,3). f(6,4). f(6,5). f(6,6). f(6,7). f(6,8).

f(7,1). f(7,2). f(7,3). f(7,4). f(7,5). f(7,6). f(7,7). f(7,8).

f(8,1). f(8,2). f(8,3). f(8,4). f(8,5). f(8,6). f(8,7). f(8,8).

win:-

makewindow(1,7,8,"Dimon",0,0,25,75). *// создание окна(задание цвета, размеров, расположения*)

position:-

f(X0,Y0),

X1=X0+1,

f(X1,Y1),

Y1<>Y0,

Y1-Y0<>X1-X0,

Y1-Y0<>X0-X1,

X2=X1+1,

f(X2,Y2),

Y2<>Y1, Y2<>Y0,

Y2-Y0<>X2-X0, Y2-Y1<>X2-X1,

Y2-Y0<>X0-X2, Y2-Y1<>X1-X2,

X3=X2+1,

f(X3,Y3),

Y3<>Y0, Y3<>Y1, Y3<>Y2,

Y3-Y0<>X3-X0, Y3-Y1<>X3-X1, Y3-Y2<>X3-X2,

Y3-Y0<>X0-X3, Y3-Y1<>X1-X3, Y3-Y2<>X2-X3,

X4=X3+1,

f(X4,Y4),

Y4<>Y0, Y4<>Y1, Y4<>Y2, Y4<>Y3,

Y4-Y0<>X4-X0, Y4-Y1<>X4-X1, Y4-Y2<>X4-X2, Y4-Y3<>X4-X3,

Y4-Y0<>X0-X4, Y4-Y1<>X1-X4, Y4-Y2<>X2-X4, Y4-Y3<>X3-X4,

X5=X4+1,

f(X5,Y5),

Y5<>Y0, Y5<>Y1, Y5<>Y2, Y5<>Y3, Y5<>Y4,

Y5-Y0<>X5-X0, Y5-Y1<>X5-X1, Y5-Y2<>X5-X2, Y5-Y3<>X5-X3, Y5-Y4<>X5-X4,

Y5-Y0<>X0-X5, Y5-Y1<>X1-X5, Y5-Y2<>X2-X5, Y5-Y3<>X3-X5, Y5-Y4<>X4-X5,

write(X0,',',Y0,' '),

write(X1,',',Y1,' '),

write(X2,',',Y2,' '),

write(X3,',',Y3,' '),

write(X4,',',Y4,' '),

write(X5,',',Y5,' '), nl.

1. **Требования по выполнению лабораторной работы и формированию отчёта**

Практическая часть лабораторной работы предусматривает получение навыков программирования логических задач в программе ПРОЛОГ. Перед выполнением работы студенту необходимо получить у преподавателя вариант задания.

В отчет по лабораторной работе №1 включить следующие результаты:

1. Вариант задания.
2. Текст программы на языке ПРОЛОГ с комментариями по каждому разделу, каждой цели и подцели, а также механизма отката.
3. Описать механизм работы программы, выполнив её трассировку.
4. Защитить лабораторную работу и сформулировать выводы.

***Текст отчёта формируется в программе Mirosoft Word 97-2003 и записывается на носитель преподавателя*.**