# **Логические задачи на языке программирования Prolog**

# **Задание 1.**

1. Ввести предложенный текст программы, реализовать ее и записать на диск.

 predicates

 hello.

 goal

 hello.

 clauses

 hello:-

 makewindow(1,7,7,"Моя первая программа",4,56,14,22),

 nl, write("Введите ваше имя,\n","затем нажмите Enter."),

 cursor(4,6),

 readln(Name),nl,

 write("Добро пожаловать\n в PDC Prolog,\n","",Name,"!").

Результат: Добро пожаловать в PDC Prolog, Vladimir!

1. Умышленно ввести опечатки в текст программы и ознакомиться с сообщениями об этих опечатках, исправить их.

 predicates

 hello.

 goal

 hello.

 clauses

 hello:-

 makewindow(1,7,7,"Моя первая программа",4,56,14,22)

 nl, write("Введите ваше имя,n","затем нажмите Enter."),

 cursor(4,6),

 readln(Name)nl,

 write("Добро пожаловать\n в PDC Prolog,\n",",Name,!").

Результат: 423 Syntax eror.

# **Задание 2.**

Реализовать программу с вводом перечисленных ниже целей и проверить получаемые результаты. Цели внешние.

predicates

 book(symbol,symbol,symbol,integer).

clauses

 book("Фигурнов В.Э.","IBM PC для пользователя","ФиС",1988).

 book("Петухов О.А.","Проектирование ОРБД","Судостроение",1984).

 book("Петухов О.А.","Объектно-реляционные модели данных","СЗПИ",1987).

 book("Петухов О.А.","Моделирование СМО","СЗПИ",1989).

 book("Петухов О.А.","PDC Prolog","СЗПИ",2000).

 book("Анкудинов Г.И.","Теория автоматов","СЗПИ",1997).

 book("Николаев В.И.","Дискретные структуры","СЗПИ",1999).

Результат: X= Proektirovanie OR BD, Y=SUDOSTROENIE, Z=1984

 X=Object models of DATA, Y=SZPI, Z=1987

 X=Modelirovanie CMO, Y=SZPI, Z=1989

 X=PDC Prolog, Y=SZPI, Z=2000

 4 Solutions

 X= Proektirovanie OR BD

 X=Object models of DATA

 X=Modelirovanie CMO

 X=PDC Prolog

 4 Solutions

 E=Petuchov O.A., X=Object models of DATA, Z=1987

 E=Petuchov O.A., X=Modelirovanie CMO, Z=1989

 E=Petuchov O.A., X= PDC Prolog, Z=2000

 E=Ankudinov G.I., X=Automat theory, Z=1997

 E=Nikolaev V.I., X=Diskretniye struktury, Z=1999

 5 Solutions

# **Задание 3.**

Реализовать программу с вводом всех перечисленных внешних целей и проверить получаемые результаты:

domains

 name=symbol

 year\_in, year\_out = integer

predicates

 parents(name,name)

 woman(name)

 man(name)

 offspring(name,name)

 father(name,name)

 mother(name,name)

 parent\_parents(name,name)

 brother(name,name)

 grandfather(name,name)

 grandmother(name,name)

 emperor(name,year\_in,year\_out)

 emperor\_was(name,integer)

clauses

 parents("Петр III","Павел I").

 parents("Екатерина II","Павел I").

 parents("Павел I","Александр I").

 parents("Павел I","Николай I").

 parents("Николай I","Александр II").

 parents("Александр II","Александр III").

 parents("Александр III","Николай II").

 woman("Екатерина II").

 man("Петр III").

 man("Павел I").

 man("Алексндр I").

 man("Николай I").

 man("Александр II").

 man("Александр III").

 man("Николай II").

 offspring(Y,X):-parents(X,Y).

 father(X,Y):-parents(X,Y),man(X).

 mother(X,Y):-parents(X,Y),woman(X).

 parent\_parents(X,Z):-parents(X,Y),parents(Y,Z).

 brother(X,Y):-parents(Z,X),parents(Z,Y),man(X),X<>Y.

 grandfather(X,Y):-father(X,Z),father(Z,Y).

 grandmother(X,Y):-mother(X,Z),father(Z,Y).

 emperor("Петр III",1761,1762).

 emperor("Екатерина II",1762,1796).

 emperor("Павел I",1796,1801).

 emperor("Александр I",1801,1825).

 emperor("Николай I",1825,1855).

 emperor("Александр II",1855,1881).

 emperor("Александр III",1881,1894).

 emperor("Николай II",1894,1917).

 emperor\_was(X,Y):-emperor(X,A,B),Y>=A,Y<=B.

Результат: Y=aleksandr I X=petr III, A=1761, B=1762 X=pavel I

 1 Solution X=ekaterina II, A=1762, B=1769 1 Solution

 X=pavel I, A=1796, B=1801

 X=aleksandr I, A=1801, B=1825

 X=nikolay I, A=1825, B=1855

 X=aleksandr II, A=1855, B=1881

 X=aleksandr III, A=1881, B=1894

 X=nikolay II, A=1894, B=1917

 8 Solutions

# **Задание 4.**

1. Реализовать приведенную программу:

domains

 name=symbol

predicates

 star(name)

 planet(name)

 revolve(name,name)

 satellite(name,name).

goal

 satellite(X,"Марс"),

 write(X," спутник Марса."),

 nl.

clauses

 star("Солнце").

 planet("Земля").

 planet("Марс").

 revolve("Земля","Солнце").

 revolve("Марс","Солнце").

 revolve("Луна","Земля").

 revolve("Фобос","Марс").

 revolve("Деймос","Марс").

 satellite(X,Y):-planet(Y),revolve(X,Y).

Результат: Фобос спутник Марса.

1. Написать и реализовать программу установления родственных связей: Василий имеет дочь Ольгу, у которой два сына Михаил и Максим. Использовать внешние и внутренние цели.

domains

 name=symbol

predicates

 men(name)

 mama(name)

 sons(name,name)

 doughter(name,name)

 deda(name,name)

 brother(name,name).

goal

 doughter(Z,Y),

 write(Z,Y),

 nl.

clauses

 men(“Vaciliy”).

 men(“Michail”).

 men(“Maxim”).

 mama(“Olga”).

 sons(“Michail”,”Olga”).

 sons(“Maxim”,”Olga”).

 doughter(“Olga”,”Vasiliy”).

 deda(X,Y) :--men(X),men(Y),sons(X,Y),doughter(Z,Y).

 brother(X,Y) :--men(X),men(Y), sons(X,Z),sons(Y,Z),X<>Y.

Результат: Olga, Vaciliy

# **Задание 5.**

1. Реализовать следующую программу: Получить псевдослучайные вещественные числа в диапазоне от 0 до 1.

goal

 random(X),

 Z=X,

 write("Псевдослучайное вещественное число от 0 до 1 : ",Z),

 nl.

Результат: 0,64823988962

1. Получить случайные целые числа в диапазоне от 0 до 10 и от 10 до 35.

goal

 random(10,X),

 Z=X,

 write("Псевдослучайное вещественное число от 0 до 10 : ",Z),

 nl.

Результат: 5

goal

 random(25,X),

 Z=X++10,

 write("Псевдослучайное вещественное число от 10 до 35:,Z),

 nl.

Результат: 21

# **Задание 6.**

1. Реализовать приведенную пргограмму для вычисления

goal

 write("X = "),

 readint(X),

 nl,

 write("K = "),

 readreal(K),

 nl,

 Z=exp(sin(X))+sqrt(K+X\*X),

 write("Z = ",Z).

Результат: X=5, K=16, Z=6,7864292326

1. В режиме калькулятора вычислить X = (2 + 5) \* 3,4, т.е. вводя значения 2, 5, 3.4 с клавиатуры.

goal

 write("X = "),

 readint(X),

 nl,

 write("K = "),

 readint(K),

 nl,

 write(“N= “),

 readreal(N),

 nl

 Z=(X+K)\*N,

 write("Z = ",Z).

Результат: 23,8

# **Задание 7.**

1. Реализовать приведенную программу с внешней связью, выполнив все четыре арифметические операции.

predicates

 operation(symbol,real,real)

clauses

 operation("+",X,Y):-Z=X+Y,

 write(X,"+",Y,"=",Z),

 nl.

 operation("-",X,Y):-Z=X-Y,

 write(X,"-",Y,"=",Z),

 nl.

 operation("\*",X,Y):-Z=X\*Y,

 write(X,"\*",Y,"=",Z),

 nl.

 operation("/",X,Y):-Z=X/Y,

 write(X,"/",Y,"=",Z),

 nl.

Результат: Z=8+2 Z=8-2 Z=8\*2 Z=8/2

 Z=10 Z=6 Z=16 Z=4

 1 Solution 1 Solution 1 Solution 1 Solution

2. Реализовать эту же программу с внутренней целью

predicates

 operation(symbol,real,real)

Goal

 write(“Vvedite chisla”),

 nl,

 readreal(X),

 nl,

 readreal(Y),

 nl,

 operation(“+”,X,Y),

 operation(“-“,X,Y),

 operation(“\*”,X,Y),

 operation(“/ “,X,Y).

clauses

 operation("+",X,Y):-Z=X+Y,

 write(X,"+",Y,"=",Z),

 nl.

 operation("-",X,Y):-Z=X-Y,

 write(X,"-",Y,"=",Z),

 nl.

 operation("\*",X,Y):-Z=X\*Y,

 write(X,"\*",Y,"=",Z),

 nl.

 operation("/",X,Y):-Z=X/Y,

 write(X,"/",Y,"=",Z),

 nl.

Результат: Vvedite chisla

 2

 4

 2+4=6

 2-4=-2

 2\*4=8

 2/4=0,5

# **Задание 9.**

Реализовать программу задания 4 с новой целью, использующей встроенный предикат *fail*, и проанализировать полученный результат.

domains

 name=symbol

predicates

 men(name)

 mama(name)

 sons(name,name)

 doughter(name,name)

 deda(name,name)

 brother(name,name).

goal

 deda(X,”Vaciliy”),

 write(X,”Vaciliy”),

 nl.

clauses

 men(“Vaciliy”).

 men(“Michail”).

 men(“Maxim”).

 mama(“Olga”).

 sons(“Michail”,”Olga”).

 sons(“Maxim”,”Olga”).

 doughter(“Olga”,”Vasiliy”).

 deda(X,Y) :--men(X),men(Y),sons(X,Y),doughter(Z,Y),nl,

 write(“ “,X),nl, fail.

 brother(X,Y) :--men(X),men(Y), sons(X,Z),sons(Y,Z),X<>Y.

Результат: Michail

 Maxim

# **Задание 10.**

Реализовать программы вычисления суммы следующих рядов:

1. 1 + 2 + 3 + ... + 9 + 10
2. 2 + 4 + 6 + ... + 14 + 16
3. 10 + 9 + 8 + ... + 2 + 1
4. 1 + 3 + 5 + ... + 13 + 15

domains

 number,sum=integer

 predicates

 sum(number,sum)

goal

 write(“Сумма ряда : “),

 sum(1,sum), write(sum).

clauses

 sum(11,0).

 sum(Number, Sum) :--

 New\_number=Number+1,

 sum(New\_number,Partial\_sum),

 Sum=Number+Partial\_sum.

Результат: Сумма ряда : 55

domains

 number,sum=integer

 predicates

 sum(number,sum)

goal

 write(“Сумма ряда : “),

 sum(2,sum), write(sum).

clauses

 sum(18,0).

 sum(Number, Sum) :--

 New\_number=Number+2,

 sum(New\_number,Partial\_sum),

 Sum=Number+Partial\_sum.

Результат: Сумма ряда : 72

domains

 number,sum=integer

 predicates

 sum(number,sum)

goal

 write(“Сумма ряда : “),

 sum(9,sum), write(sum).

clauses

 sum(0,11).

 sum(Number, Sum) :--

 New\_number=Number-1,

 sum(New\_number,Partial\_sum),

 Sum=Number+Partial\_sum

Результат: Сумма ряда : 55

domains

 number,sum=integer

 predicates

 sum(number,sum)

goal

 write(“Сумма ряда : “),

 sum(1,sum), write(sum).

clauses

 sum(17,0).

 sum(Number, Sum) :--

 New\_number=Number+2,

 sum(New\_number,Partial\_sum),

 Sum=Number+Partial\_sum.

Результат: Сумма ряда : 64

# **Задание 12.**

Написать программу, создающую список городов. Выполнить программу с различными внутренними и внешними целями.

domains

 town\_list=town\*

 town=symbol

predicates

 towns(town\_list)

goal

 towns([A,B,C,D,E]),

 write(A,”,”,B,”,”,C,”,”,D,”,”,E).

clauses

 towns([ “Kazan”,”Nignekamsk”,”Elabuga”,”Bugulma”,”Almetevsk” ]).

# Результат: Kazan, Nignekamsk, Elabuga, Bugulma, Almetevsk