МИНИСТЕРСТВО ОБРАЗОВАНИЯ

Белорусский государственный университет

информатики и радиоэлектроники

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Лабораторная работа № 3 по дисциплине

ЭКСПЕРТНЫЕ СИСТЕМЫ

на тему

«РАЗВИТЫЕ ТИПЫ ДАННЫХ В ПРОГРАММАХ НА ПРОЛОГЕ.   
БАЗЫ ДАННЫХ В ПРОГРАММАХ НА ПРОЛОГЕ»

Вариант 4

Выполнили:

ст. гр.620603

Гуща А. Д.

Колесников В. Г.

Проверил:

Тиханович Т. В.

Минск 2020

# Цель работы

Изучение развитых типов данных в программах на Прологе, необходимых для программной реализации экспертных систем. Изучение возможностей построения и обработки баз данных на Прологе.

# Условие

Создать базу данных компании, владеющей несколькими речными судами. Суда могут быть грузовыми, пассажирскими или грузо-пассажирскими. База данных должна содержать следующую информацию:

- о судах, принадлежащих компании. Каждый факт содержит название судна, год постройки, вместимость (для грузовых – количество тонн груза, для пассажирских - количество мест, для грузо-пассажирских - и то, и другое), фамилию капитана. Примеры:

sudno("Днепр",1985,g(800),"Семенов")

sudno("Звезда",1990,p(100),"Степанов")

sudno("Витебск",1983,gp(300,80),"Гришин")

Указание. Здесь g, p, gp - функторы, входящие в альтернативный домен (обозначают соответственно грузовые, пассажирские, грузо-пассажирские суда и их вместимость).

- о заказах на перевозку грузов. Каждый факт содержит шифр заказа, вид груза, стоимость перевозки, название судна, пункт отправления и прибытия. Примеры:

zakaz("П20","зерно",1200,"Днепр","Могилев","Быхов")

Разработать программу, выполняющую следующие операции: загрузка базы данных; просмотр списка заказов на экране; добавление нового судна; сохранение; выход.

# Теоретическая часть

Функторы представляют собой составные объекты программы на Прологе. Они позволяют структурировать данные для удобства их представ­ления и обработки. Функторы объявляются в разделе domains следую­щим образом:

domains

тип = имя\_функтора (типы аргументов)

Одна из наиболее важных задач, для которых применяются функции в программах на Прологе – объявление аргументов предикатов, которые в разных случаях могут принимать значения разных типов. В этом случае необходимо объявить набор функторов, называемых *альтернативным доменом*:

domains

альтернативный\_домен = функтор\_1 (типы аргументов);

функтор\_2 (типы аргументов);

.....

функтор\_n (типы аргументов)

Если затем указать "альтернативный домен" в качестве типа ар­гумента при объявлении предиката (в разделе predicates), то в качестве аргумента предиката можно будет указывать любой из функторов "функтор\_1", ..., "функтор\_n".

В программах на Прологе имеется возможность хранить предикаты-факты, обра­батываемые программой, отдельно от самой программы. Файл, содержа­щий предикаты-факты, называется базой данных.

Факты, хранящиеся в файле базы данных, объявляются в раз­деле global facts (или global database). Нестандартные типы данных, используемые в базе данных, должны быть объявлены в разделе global domains.

В одной базе данных могут храниться факты, имеющие разные имена, разное количество аргументов и т.д. Все они должны быть объявлены в разделе global facts.

Информация в базу данных может вводиться обычным текстовым редактором (например, редактором системы Visual Prolog). В каждой строке базы данных может содержаться *только один факт*. В базе данных не должно быть пустых строк, *в том числе в конце базы данных (после последнего факта)*. Факты в базе данных должны быть набраны *без пробелов*. Точка в конце фактов *не ставится*.

Загрузка базы данных во время работы программы выполняется стандартным предикатом consult (имя\_файла). Имя файла базы данных может задаваться константой (в этом случае оно должно быть заключено в кавычки) или переменной. Если используется переменная, то ей должно быть присвоено имя файла базы данных. Для указания имени файла в этом случае может использоваться предикат readln или операция присваивания.

Если имя файла базы данных *указывается в тексте программы* (например, задается в предикате consult или присваивается переменной), то в нем необходимо использовать двойные наклонные черты (\\). Если имя файла *вводится с клавиатуры* с помощью оператора readln, то используются одиночные наклонные черты (\).

**4. Выполнение работы**

База данных:

*ship("Aristotle",1994,p(100),"Valera")*

*ship("Sherlock",1987,cp(300,80),"Lord")*

*ship("Royal",1981,c(800),"Boroda")*

*ship("Kawa",1998,p(100),"Kafka")*

*ship("Velours",1989,cp(300,80),"Prince")*

*ship("Ikarus",1997,c(100),"Vanya")*

*ship("Bugai",1870,cp(200,12),"Ritmo")*

*ship("Qw",2000,c(300),"As")*

*ship("Qw",2000,c(200),"As")*

*ship("e",2,p(4),"q")*

*order("n1","table",100,"Aristotle","Mogilev","Brest")*

*order("n2","book",200,"Aristotle","Vitebsk","Gomel")*

*order("n3","chair",300,"Sherlock","Borisov","Minsk")*

*order("n4","phone",400,"Sherlock","Minsk","Grodno")*

*order("n5","air",500,"Royal","Mogilev","Grodno")*

*order("n6","water",600,"Royal","Minsk","Brest")*

*order("n7","phone",700,"Kawa","Brest","Gomel")*

*order("n8","table",800,"Kawa","Grodno","Mogilev")*

*order("n9","chair",900,"Velours","Gomel","Vitebsk")*

*order("n10","pen",1000,"Velours","Brest","Minsk")*

Код программы:

*include "laaba3.inc"*

*global domains*

*size = c (integer);*

*p (integer);*

*cp (integer, integer)*

*global facts*

*ship (string, integer, size, string)*

*order (string, string, integer, string, string, string)*

*predicates*

*nondeterm load*

*nondeterm show\_ships*

*nondeterm show\_orders*

*nondeterm add\_ship*

*nondeterm add\_ship\_case(string, string, integer, string)*

*nondeterm menu*

*nondeterm menu\_case(integer)*

*nondeterm task*

*nondeterm saver*

*goal*

*task.*

*clauses*

*task:-*

*load, nl,*

*menu.*

*task:-*

*write("error"), readchar(\_).*

*show\_ships:-*

*ship (Name, Year, c(Capacity), Cap),*

*write (Name, " ", Year, " Capacity:", Capacity, " ", Cap),*

*nl, fail.*

*show\_ships:-*

*ship (Name, Year, p(Seats), Cap),*

*write (Name, " ", Year, " Seats:", Seats, " ", Cap),*

*nl, fail.*

*show\_ships:-*

*ship (Name, Year, cp(Capacity, Seats), Cap),*

*write (Name, " ", Year, " Capacity:", Capacity, " Seats:", Seats, " ", Cap),*

*nl, fail.*

*show\_ships.*

*show\_orders:-*

*order(Key, Product, Coast, Name, Start, Finish),*

*write(Key, " ", Product, " ", Coast, " ", Name, " ", Start, " ", Finish),*

*nl, fail.*

*show\_orders.*

*add\_ship:-*

*write("enter name: "), readln(Name),*

*write("enter year: "), readint(Year),*

*write("enter captain: "), readln(Captain),*

*write("choise ship type(c, p, cp): "), readln(Type),*

*add\_ship\_case(Type, Name, Year, Captain).*

*add\_ship:-*

*write("Fail!").*

*add\_ship\_case("c", Name, Year, Captain):-*

*write("enter capacity: "), readint(Capacity),*

*assert(ship(Name, Year, c(Capacity), Captain)).*

*add\_ship\_case("p", Name, Year, Captain):-*

*write("enter seats amount: "), readint(Seats),*

*assert(ship(Name, Year, p(Seats), Captain)).*

*add\_ship\_case("cp", Name, Year, Captain):-*

*write("enter capacity: "), readint(Capacity),*

*write("enter seats amount: "), readint(Seats),*

*assert(ship(Name, Year, cp(Capacity, Seats), Captain)).*

*add\_ship\_case(\_,\_,\_,\_):- fail.*

*load:-*

*retractall(\_),*

*consult("D:\\prolog\\BIN\\WIN\\32\\lab3.dba"),*

*write("DB loaded."), nl, !.*

*load:-*

*write("DB not loaded."), nl, readchar(\_).*

*saver:- save("D:\\prolog\\BIN\\WIN\\32\\lab3.dba"), !,*

*write("Ship added."), nl.*

*menu:-*

*write("1: Show ships."), nl,*

*write("2: Show orders."), nl,*

*write("3: Add ship."), nl,*

*write("4: Save."), nl,*

*write("5: Exit."), nl,*

*readint(Case),*

*menu\_case(Case).*

*menu\_case(1):-*

*nl, show\_ships, nl,*

*menu.*

*menu\_case(2):-*

*nl, show\_orders, nl,*

*menu.*

*menu\_case(3):-*

*nl, add\_ship, nl,*

*menu.*

*menu\_case(4):-*

*nl, saver, nl,*

*menu.*

*menu\_case(5).*

*menu\_case(\_):-*

*write("Fail!"), nl, nl,*

*menu.*

**5. Вывод**

В ходе лабораторной работы были изучены возможности построения и обработки баз данных на Прологе. Была создана база данных по условию, также были описаны такие действия как загрузка, чтение, изменение и сохранение БД.