**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики

и вычислительной техники

Кафедра ИБ

**ИСЧИСЛЕНИЕ ПРЕДИКАТОВ ПЕРВОГО ПОРЯДКА**

Вариант 4

отчет

по лабораторной работе №7

по дисциплине «ЛиФП»

Выполнил:

студент группы БИс-31

Богданов В.М.

Проверил:

Смирнов В.И.

г. Йошкар-Ола

2016

**Теоретическая часть**

*Цель работы:* изучить возможности определения объектного мира спомощью выражений исчисления предикатов с последующим переводомв синтаксические структуры языка Prolog.

*Краткие теоретические сведения:*

Часть логики, которую называют исчислением высказываний предоставляет возможность выразить многие факты и рассуждения, которыми пользуются в обыденной жизни. Исчисление высказываний основано на утверждениях, которые имеют только два значения – true (истина) и false (ложь).

Исчисление предикатов является обобщением исчисления высказываний, позволяющим использовать параметры (называемые также аргументами или переменными) в высказываниях.

Объекты данных в языке Prolog называются термами. Терм может быть константой, переменной или составным термом (структурой). Константами являются целые и действительные числа.

К константам относятся также атомы. Атом - есть любая последовательность символов, заключенная в одинарные кавычки. Кавычки опускаются, если и без них атом можно отличить от символов, используемых для обозначения переменных.

Если переменная используется только один раз, необязательно называть ее. Она может быть записана как анонимная переменная, состоящая из одного символа подчеркивания "\_". Переменные, подобно атомам, являются элементарными объектами языка Prolog. Завершает список синтаксических единиц сложный терм, или структура. Все, что не может быть отнесено к переменной или константе, называется сложным термом. Следовательно, сложный терм состоит из констант и переменных.

Логические константы в языке Prolog используются для создания новых формул из старых.

Структура состоит из атома, называемого главным функтором, и последовательности термов, называемых компонентами структуры. Компоненты разделяются запятыми и заключаются в круглые скобки.

Программировать на Prolog - значит описывать некий мир. Программа на этом языке состоит из множества фраз, задающих взаимосвязь между термами. Каждый терм обозначает ту или иную сущность, принадлежащую миру. Один из способов описания - это задание фактов. Факт - это утверждение о том, что соблюдается некоторое конкретное отношение. Факт предполагается (задается), безусловно, верным.

**Практическая часть**

**Задание 1.**

Представить утверждение в форме факта языка Prolog. Ввести факт в БД и создать запрос.

*Утверждение:* John travels to London by train

*Описание БД:* 'John travels to London by train'.

*Результат:*?-‘John travels to London by train’.

Yes

?- 'John travels to London by car'.

ERROR: Undefined procedure: 'John travels to London by car'/0

**Задание 2.**

Представить утверждение в виде формулы (предиката), доступной для реализации на языке Prolog. Ввести утверждение в БД и создать запрос.

*Утверждение:* Bill only eats chocolate, bananas or cheese.

*Описание БД:* eat(bill, chocolate).

eat(bill, bananas).

eat(bill, cheese).

*Результат:* ?- eat(bill, chocolate).

Yes

?- eat(bill, bananas).

Yes

?- eat(bill, sugar).

No

**Задание 3.**

Представить приведенные ниже утверждения в виде одноцелевых правил. Ввести правила в БД и создать запрос.

*Утверждение:* If someone needs a bike then they may borrow jane’s.

*Описание БД:* need(\_,bike).

lend(jane,\_,\_x):-

need(\_,\_x).

*Результат*:?- lend(jane,nick,bike).

Yes

?- lend(jane,chris,car).

No

**Задание 4.**

Представить приведенные ниже утверждения в виде правил с двумя подцелями. Ввести правила в БД и создать запрос.

*Утверждение:* You are liable to be fined if your car is untaxed.

*Описание БД:* have(i,car).

is1(car,untaxed).

pay(\_x,\_y):-

have(i,\_x),

is1(car,\_y).

*Результат:*?- pay(car,untaxed).

Yes

?- pay(car,taxed).

No

?- pay(bike,untaxed).

No

?- pay(bike,taxed).

No

**Задание 5.**

Представить приведенное ниже утверждение в виде нескольких правил. Ввести правила в БД и создать запрос.

*Утверждение:* You are british if you are welsh, english, scottish or northernirish.

*Описание БД:*

welsh(welsh).

scottish(scottish).

english(english).

northernirish(northernirish).

british(\_x):- welsh(\_x).

british(\_x):- scottish(\_x).

british(\_x):- english(\_x).

british(\_x):- northernirish(\_x).

*Результат:*

?- british(english).

Yes

?- british(spanish).

No

**Задание 6.**

Представить выражение в форме, доступной для реализации на языке Prolog. Ввести выражение в БД и создать запрос.

*Утверждение:* Jane is tall.

*Описание БД:* ‘Jane is tall’.

*Результат:*?- ‘Jane is tall’.

Yes

*Вывод*: в ходе работы были изучены методы и задачи логического программирования; основы исчисления предикатов первого порядка и составление выражений; типы данных языка Prolog.