Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Институт «Электронных и информационных систем»

Кафедра «Информационных технология и систем»

Лабораторная работа №9

**РЕШЕНИЕ ЛОГИЧЕСКИХ ЗАДАЧ МЕТОДОМ «ОБРАЗУЙ И ПРОВЕРЬ»**

по дисциплине:

«Функциональное и логическое программирование»

**Отчёт**

Принял преподаватель:

\_\_\_\_\_\_\_\_\_\_ / Михайлов Д.В

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Выполнил студент группы 8091:

\_\_\_\_\_\_\_\_\_\_ / Лехновский А. Д.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**Великий Новгород**

**2021**

1. **Цель и задачи данной лабораторной работы**

Целью лабораторной работы является изучение приемов использования метода «Образуй и проверь» при решении задач на Прологе.

1. **Задание на лабораторную работу**

**Задача:**

Три купчихи – Олимпиада, Сосипатра и Поликсена – пили чай.

* Если бы Олимпиада выпила на 5 чашек больше, то она выпила бы столько, сколько две другие вместе.
* Если бы Сосипатра выпила бы на 9 чашек больше, то она выпила бы столько, сколько две другие вместе.

Требуется написать программу, которая определяла бы, сколько каждая из трех купчих, выпила чашек и кого из них какое отчество, если известно, что:

* Уваровна пила чай вприкуску;
* количество чашек чая, выпитых Титовной, кратно трем;
* Карповна выпила 11 чашек.

1. **Решение поставленной задачи**

Для выполнения поставленной задачи был использован Visual Prolog 5.2.

«Образуй и проверь» - общий прием проектирования алгоритмов и программ. Суть его состоит в том, что один процесс или программа генерирует множество предполагаемых решений задачи, а другой процесс или программа проверяет эти предполагаемые решения и пытается найти те из них, которые действительно являются решениями задачи.

В реализованной программе в качестве генератора предполагаемых решений используются правила generate. Эти правила, основываются на 4 условиях, три из которых:

1. Уваровна пила чай вприкуску;
2. количество чашек чая, выпитых Титовной, кратно трем;
3. Карповна выпила 11 чашек.

Четвёртое условие было введено, чтобы ограничить количество сгенерированных решений, которые точно не подойдут для решения задачи. Так как из начальных условий, нам неясно сколько максимум чашек могло быть выпито одной из купчих. Давайте получим это условие, которое покажет нам максимум выпитых кружек одной из купчих:

Если Олимпиада -> Карповна -> 11 чашек, тогда

Олимпиада + 5 = Сосипатра + Поликсена; (Max = 11 + 5 = 16 чашек).

Если Сосипатра -> Карповна -> 11 чашек, тогда

Сосипатра + 9 = Олимпиада + Поликсена; (Max = 11 + 9 = 20 чашек).

Если Поликсена -> Карповна -> 11 чашек, тогда

В этом случае, данная система не имеет решений.

Таким образом, получается наше четвертое условие:

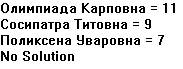
4) Максимальное количество чашек, которых могло быть выпито одной из купчих ≤ 20.

Правило test проверяет допустимость решения. Здесь вступают в силу условия “если бы” и ограничения на то, что у каждой купчихи разные отчества.

Правило get\_answer представляет собой конъюнкцию двух целей, одна из которых действует как генератор предполагаемых решений (generate), а вторая проверяет допустимость полученных решений (test). Если проверка test завершается отказом, то производится возвращение к цели generate, с помощью которой генерируется следующий элемент. Процесс продолжается итерационно до тех пор, пока при успешной проверке не будет найдено решение с требуемыми свойствами или генератор не исчерпает все альтернативные решения.

Код программы для решения задачи приведен в *Приложении 1*.

Пример работы программы:

**

*Рисунок №1. Результат работы программы для при вызове get\_answer(Middle\_name1,Cup1,Middle\_name2,Cup2,Middle\_name3,Cup3).*

1. **Вывод**

В данной лабораторной работе, мною были изучены приемы использования метода «Образуй и проверь», а также решена поставленная задача с помощью этого метода.

**Приложение 1**

Код программы:

PREDICATES

nondeterm generate(integer,integer,string)

nondeterm get\_answer(string,integer,string,integer,string,integer)

nondeterm test(string,integer,string,integer,string,integer)

CLAUSES

%Для правила generate - генерация возможных решений

%Первый аргумент - просто счётчик

%Второй аргумент - количество выпитых чашек

%Третий аргумент - отчество человека, походящее по условиям

generate(J,J,"Уваровна"). % В остальных случаях Уваровна

generate(J,J,"Титовна"):- %Количество чашек,выпитых Титовной,кратно трём

J mod 3=0.

generate(11,11,"Карповна"). %Карповна выпила 11 чашек

generate(I,J,B):- %Исходя из условий задачи маловерятно, что кто-то из купчих выпил больше 20 чашек

I<20, U=I+1, generate(U,J,B).

test(B,A,D,C,F,E):-

%Проверка на то, что у всех трёх купчих будут разные отчества

B<>D,

F<>B,

F<>D,

%Условия "Если бы", накладываемые на перечень возможных решений

A+5=C+E, % Если бы Олимпиада выпила на 5 чашек больше, то она выпила бы столько, сколько две другие вместе.

C+9=A+E. % Если бы Сосипатра выпила бы на 9 чашек больше, то она выпила бы столько, сколько две другие вместе.

get\_answer(B,A,D,C,F,E):-

%Генерируем возможные количества, выпитых чашек и Отчества для каждой купчихи

generate(1,A,B),

generate(1,C,D),

generate(1,E,F),

%Проверка допустимости решений

test(B,A,D,C,F,E),

writef("Олимпиада % = % \nСосипатра % = % \nПоликсена % = % \n",B,A,D,C,F,E),

fail.

GOAL

get\_answer(Middle\_name1,Cup1,Middle\_name2,Cup2,Middle\_name3,Cup3).