|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**Лабораторная работа №11**

**по курсу “Функциональное и логическое программирование”**

# по теме “ Среда Visual Prolog 5.2”

|  |  |
| --- | --- |
| Студент: | Уласик Е.А. |
| Группа: | ИУ7-61 |
| Преподаватель: | Толпинская Н.Б. |

*2020 г.*

**Цель работы:** познакомиться со средой Visual Prolog, познакомиться со структурой программы: способом запуска и формой вывода результатов.

**Задачи работы**: изучить принципы работы в среде Visual Prolog, возможность получения однократного и многократного результата, изучить базовые конструкции языка Prolog, структуру программы Prolog, форму ввода исходных данных и вывода результатов работы программы.

Запустить среду Visual Prolog5.2. Настроить утилиту TestGoal (способ настройки см. в дополнительных материалах к лаб. раб.). Запустить тестовую программу, проанализировать реакцию системы и множество ответов. Разработать свою программу - «Телефонный справочник». Протестировать работу программы.

Теоретическая часть

1. Что собой представляется программа на prolog?

Программа на Prolog представляет собой: базу знаний и вопрос. С помощью подбора ответов на запросы он (Prolog, программа) извлекает хранящуюся (известную в программе) информацию. База знаний содержит истинностные знания, используя которые программа выдает ответ на запрос. Одной из особенностей Prolog является то, что при поиске ответов на вопрос, он рассматривает альтернативные варианты и находит все возможные решения (методом проб и ошибок) — множества значений переменных, при которых на поставленный вопрос можно ответить –«да». Prolog включает в себя механизм вывода, который основан на сопоставлении образцов термов, например терма вопроса и терма факта. Цель работы программы – определить является ли вопрос логическим следствием программы или нет, что выполняется с применения правил вывода. Правила вывода – это утверждения о взаимосвязи между допущениями и заключениями, которые с позиции исчисления предикатов верны всегда.

Существенным недостатком в использовании этих правил вывода является необходимость «угадать» подстановку или пример терма. Кроме этого, переменные в факте и в вопросе могут стоять на одной позиции. Поэтому для выполнения логического вывода используется **механизм (алгоритм) унификации**, встроенный в систему.

**Унификация –** операция, которая позволяет формализовать процесс логического вывода (наряду с правилом резолюции). С практической точки зрения - это основной вычислительный шаг, с помощью которого происходит:

* Двунаправленная передача параметров процедурам,
* Неразрушающее присваивание,
* Проверка условий (доказательство).

В процессе работы система выполняет большое число унификаций. Процесс унификации запускается автоматически, но пользователь имеет право запустить его принудительно с помощью утверждения (немного нарушает форму записей): T1 = T2 . Унификация – попытка "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией (неудачей). В последнем случае включается механизм отката к предыдущему шагу.

1. Структура программы на prolog?

Программа может состоять только из фактов. Программа может состоять из фактов и правил – знаний. Одно знание может быть записано с помощью нескольких предложений.

С помощью термов и более сложных конструкций языка Prolog – **фактов** и **правил** «описываются» знания о предметной области, т.е. **база знаний**. Используя базу знаний, система Prolog будет делать логические выводы, отвечая на наши **вопросы**. Таким образом, **программа на Prolog представляет собой базу знаний и вопрос**.

**База знаний состоит из** предложений - CLAUSES (отдельных знаний или утверждений**): фактов и правил**. Каждое предложение заканчивается точкой. Предложения бывают двух видов: факты и правила. Предложение более общего вида – **правило** имеет вид:

A :- B1,... , Bn.

**A** называется **заголовком правила**, а B1,..., Bn – **телом правила**.

**Факт** – это частный случай правила. Факт – это предложение, в котором отсутствует тело (т.е. тело пустое).

Причем, A, B1,... , Bn  – это термы; символ "**:-**" это специальный символ-разделитель.

**Заголовок содержит** отдельное **знание** о предметной области (составной терм), а тело содержит условия истинности этого знания. Правило называют условной истиной, а факт, не содержащий тела – безусловной истиной.

Заголовок, как составной терм f(t1, t2, …,tm) , содержит **знание о том**, что между аргументами: t1, t2, …,tm существует **отношение** (взаимосвязь, взаимозависимость). А **имя** этого **отношения** – это **f**.

Если это факты (или правила) программы, то это записывается в разделе - CLAUSES:

**База знаний состоит из** предложений - CLAUSES (отдельных знаний или утверждений**): фактов и правил**. Каждое предложение заканчивается точкой. Предложения бывают двух видов: факты и правила. Предложение более общего вида – **правило** имеет вид:

A :- B1,... , Bn.

**A** называется **заголовком правила**, а B1,..., Bn – **телом правила**.

**Основным элементом языка** является **терм. Терм – это:**

1. **Константа:**

* Число (целое, вещественное),
* Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со строчной буквы: aA, ab\_2), используется для обозначения конкретного объекта предметной области или для обозначения конкретного отношения,
* Строка: последовательность символов, заключенных в кавычки,

1. **Переменная:**

* Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания ( X, A21, \_X),
* Анонимная - обозначается символом подчеркивания ( \_ ),

1. **Составной терм:**

* Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется: **f(t1, t2, …,tm)**, где f - функтор (функциональный символ) , t1, t2, …,tm – термы, в том числе и составные (их называют аргументами. Аргументом или параметром составного терма может быть константа, переменная или составной объект. Число аргументов предиката называется его **арностью** или **местностью**. Составные термы с одинаковыми функторами, но разной арности, обозначают разные отношения.

3. Как формируется результат работы программы?

Программа на Prolog может содержать вопрос в программе (так называемая внутренняя цель GOAL). Если программа содержит внутреннюю цель, то после запуска программы на выполнение система проверяет достижимость заданной цели, исходя из базы знаний.

Ответ на поставленный вопрос система дает в логической форме – «Да» или «нет». Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». Вариантов ответить «Да» на поставленный вопрос может быть несколько. Система может (в нашем случае обучения - должна) быть настроена в режим получения всех возможных вариантов ответа «Да» на поставленный вопрос.

Практическая часть

*Тестовая задача:*

include "test.inc"

predicates

likes(symbol, symbol)

clauses

likes(ellen, tennis).

likes(john, football).

likes(tom,baseball).

likes(eric, swimming).

likes(mark, tennis).

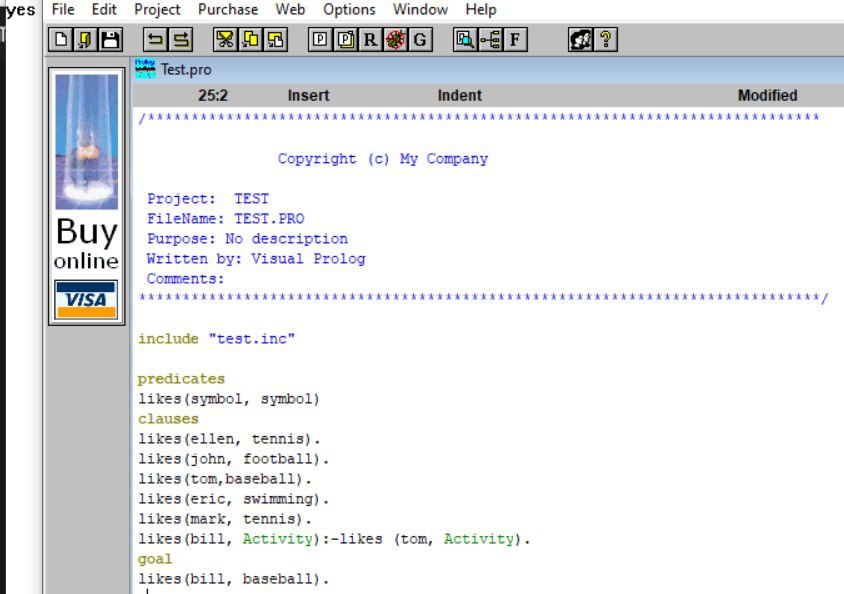
ikes(bill, Activity):-likes (tom, Activity).

goal

likes(bill, baseball).

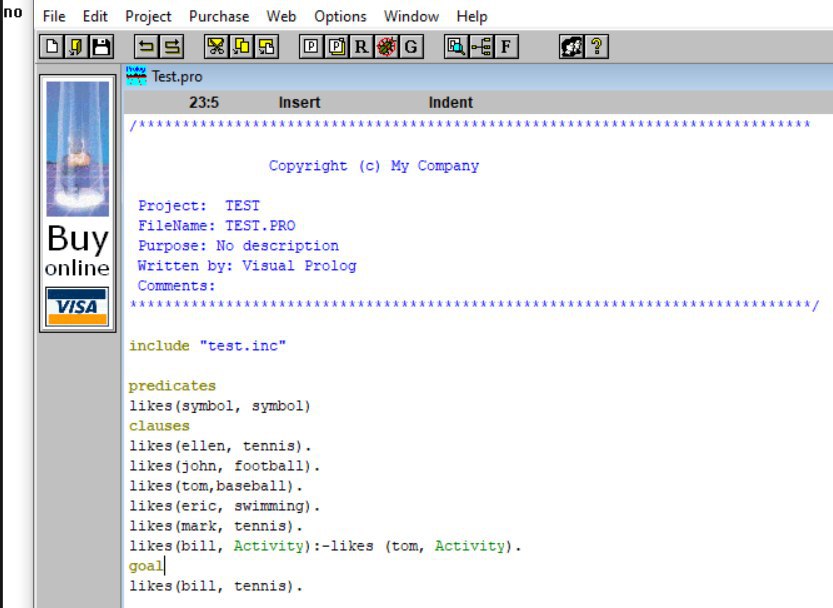
Листинг 1. Тестовая программа

Результат работы программы:



Картинка 1. Результат работы программы

Если изменил goal на likes(bill, baseball). , то выводится “no”:



Картинка 2. Результат работы программы

*Телефонный справочник:*

Описание переменных:

* Surname – фамилия человека;
* Phone – номер телефона;
* Street – улица;
* House – дом;

include "lab\_1.inc"

predicates

person(string, string, string, integer).

getInfo(string, string, string, integer).

clauses

person("Ivanov", "+71221231242", "Pushkina", 12).

person("Petrov", "+73214237612", "Lermontova", 6).

person("Smirnov", "+74126384227", "Griboedova", 1).

person("Markov", "+76234363174", "Gogolya", 24).

person("Kornilov", "+76343126412", "Chehova", 16).

person("Krasnov", "+76125212353", "Tolstova", 2).

person("Denikin", "+76343246344", "Kuprina", 6).

person("Semyonov", "+76233246343", "Bunina", 12).

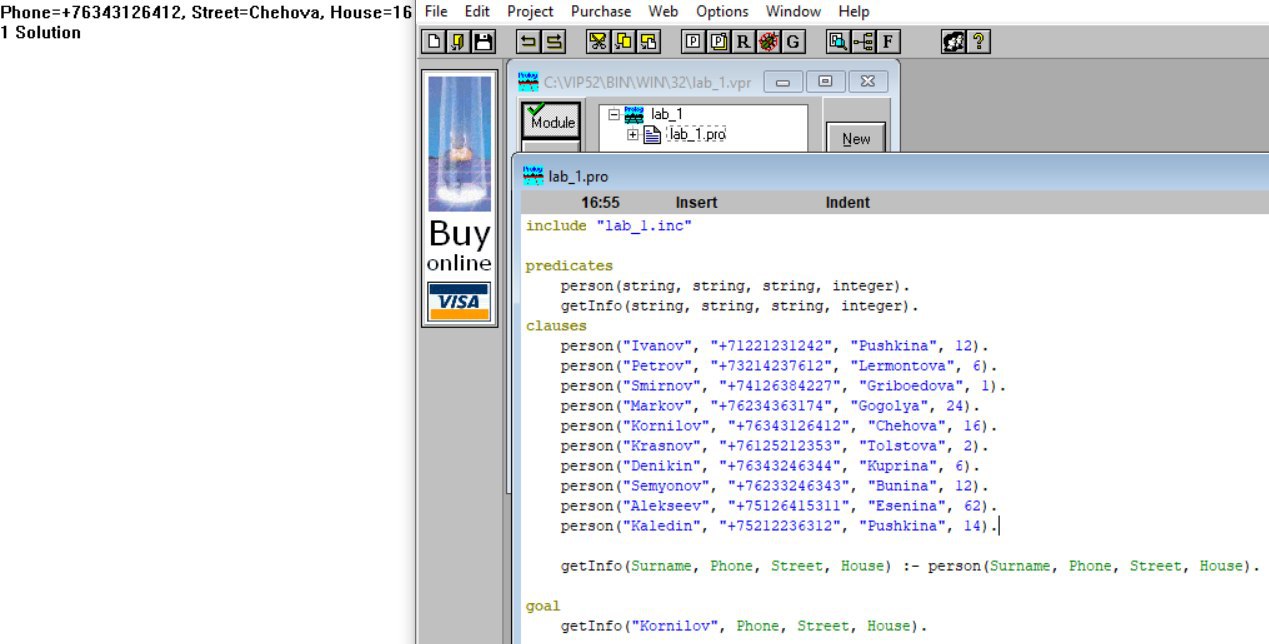
person("Alekseev", "+75126415311", "Esenina", 62).

person("Kaledin", "+75212236312", "Pushkina", 14).

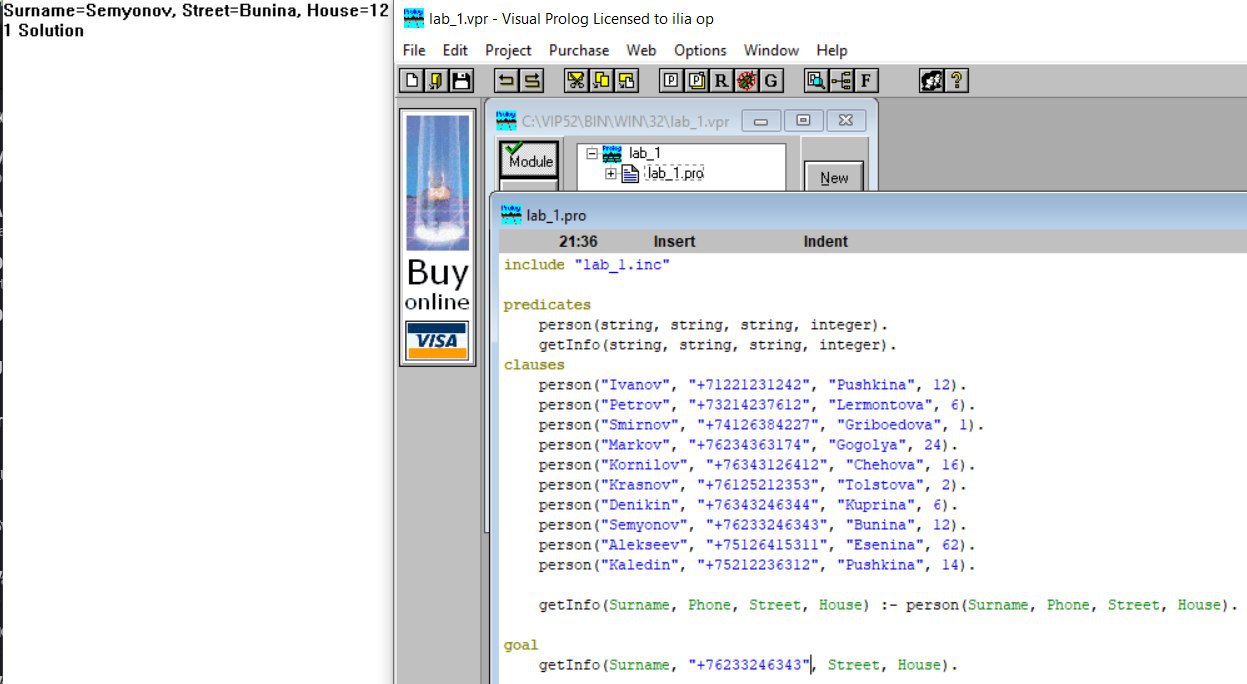
getInfo(Surname, Phone, Street, House) :- person(Surname, Phone, Street, House).

goal

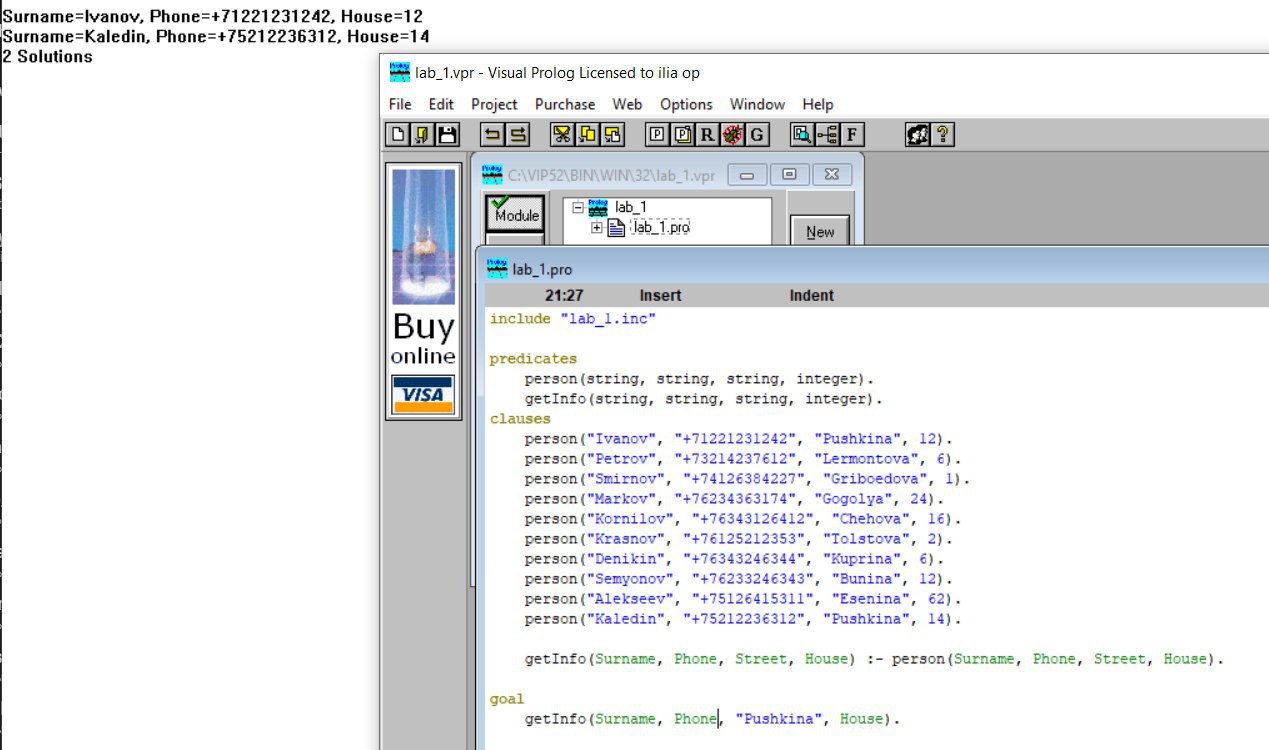
getInfo("Kaledin", Phone, Street, House).



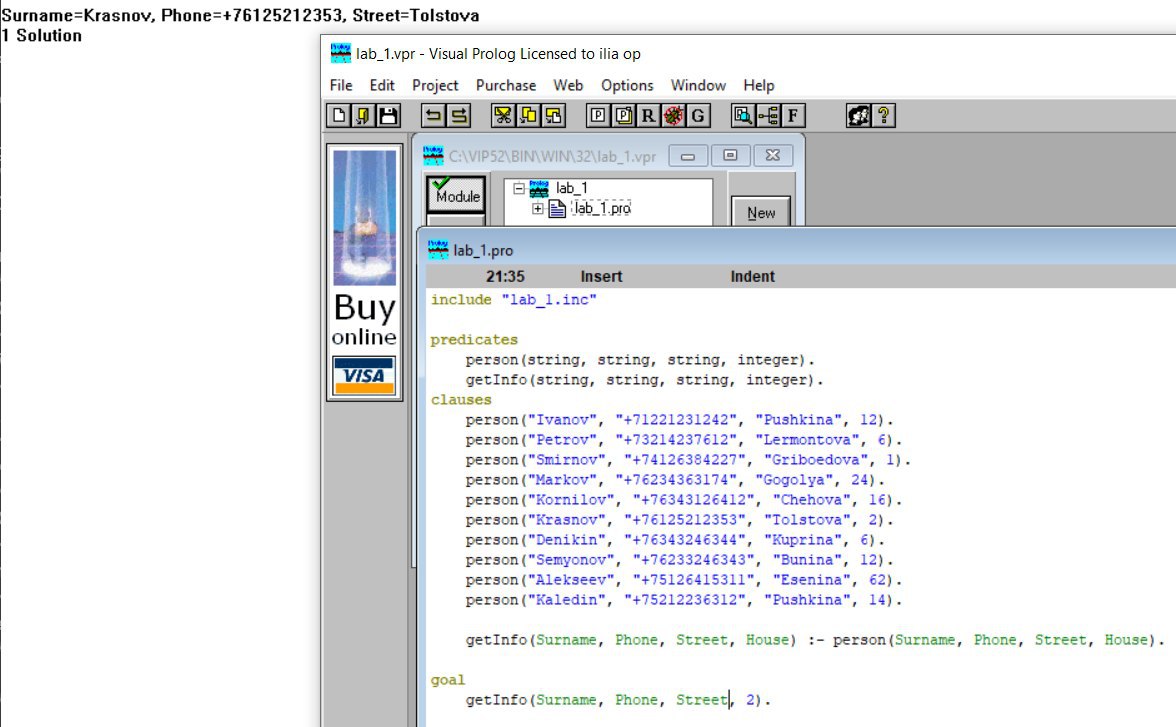
Картинка 3. Результат работы программы



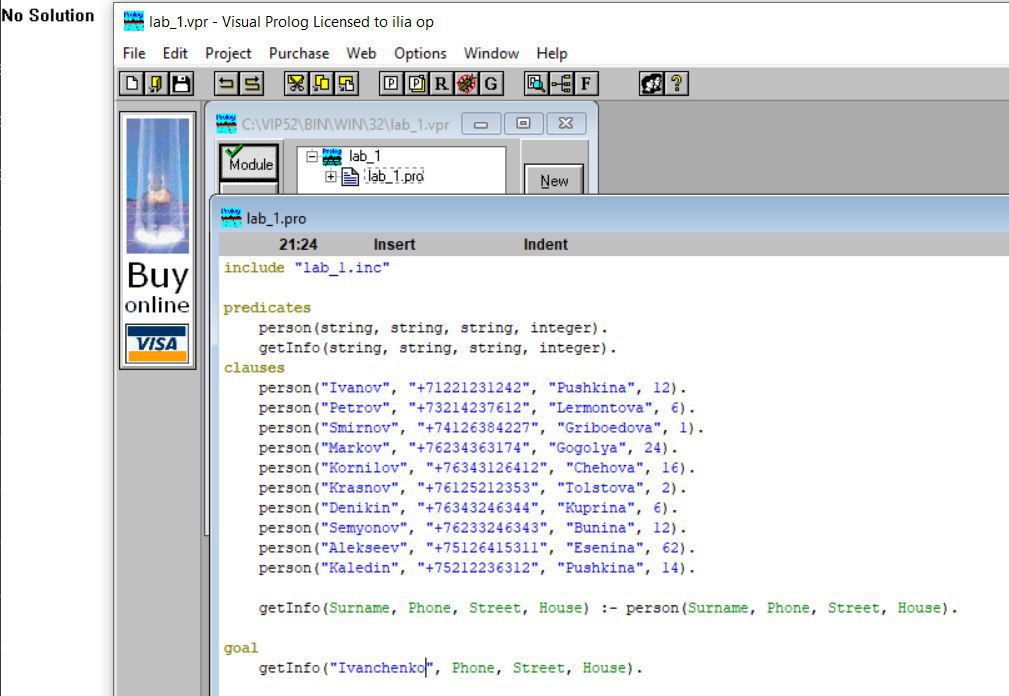
Картинка 4. Результат работы программы



Картинка 5. Результат работы программы



Картинка 6. Результат. работы программы



Картинка 7. Результат работы программы.

В ходе проведённого тестирования все ожидаемые результаты совпали с полученными.