|  |  |
| --- | --- |
| 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 №12**

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

# по теме “ Структура программы на Prolog”

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

*2020 г.*

**Цель работы:** познакомиться со структурой, принципами оформления и логикой выполнения программы на Prolog

**Задачи работы**: приобрести навыки декларативного описания предметной области с использованием фактов и правил.

Изучить способы использования фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, принцип унификации.

Составить программу – базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов (не менее 3-х). Описать порядок формирования вариантов ответа.

Исходную базу знаний сформировать с помощью только фактов.

\* Исходную базу знаний сформировать, используя правила.

\* Разработать свою базу знаний (содержание произвольно).

Теоретическая часть

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), (например: likes(judy, tennis) – знание о том, что judy любит tennis или еще, например: book( author(tolstoy, liev ), war\_and\_peace) и т.д. ). 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.

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**. **База знаний состоит из** предложений - CLAUSES (отдельных знаний или утверждений**): фактов и правил**. Каждое предложение заканчивается точкой.

Третьим специфическим видом предложений Prolog можно считать вопросы. Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель. Третьим специфическим видом предложений Prolog можно считать вопросы. Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Факты, правила, и вопросы могут содержать переменные. В процессе выполнения программы переменные могут связываться с различными объектами – **конкретизироваться**. Это относится только к именованным переменным. Поиск содержательного ответа на поставленный вопрос, с помощью имеющейся базы знаний, фактически заключается в поиске нужного знания, но какое знание понадобится – заранее неизвестно. Этот поиск осуществляется формально с помощью механизма **унификации**, встроенного в систему и не доступного программисту. Упрощенно, процесс унификации можно представить как формальный процесс сравнивания (сопоставления) терма вопроса с очередным термом знания. При этом, знания по умолчанию просматриваются сверху вниз, хотя такой порядок и не очевиден.

Программа на Prolog состоит из разделов. Каждый раздел начинается со своего заголовка. Структура программы:

* директивы компилятора — зарезервированные символьные константы
* CONSTANTS — раздел описания констант
* **DOMAINS** — раздел описания доменов
* DATABASE — раздел описания предикатов внутренней базы данных
* **PREDICATES** — раздел описания предикатов
* **CLAUSES** — раздел описания предложений базы знаний
* **GOAL** — раздел описания внутренней цели (вопроса).

В программе необязательно должны быть все разделы.

Практическая часть

1. База знаний студентов:

Назначение переменных:

* university – название университета;
* lastname – фамилия студента;
* name – имя студента;

include "lab\_2.inc"

predicates

student(symbol, symbol, symbol).

clauses

student("BMSTU", "Ivanov", "Dmitriy").

student("MSU", "Petrov", "Petr").

student("MAI", "Semenov", "Semen").

student("MEI", "Alexandrov", "Alexander").

student("BMSTU", "Maksimov", "Maksim").

student("BMSTU", "Petrov", "Ivan").

student("MSU","Ivanov", "Semen").

student("MIRAE", "Alexandrov", "Anatoliy").

student("MSU", "Petrov", "Dmitriy").

student("BMSTU", "Ivanchuk", "Kirill").

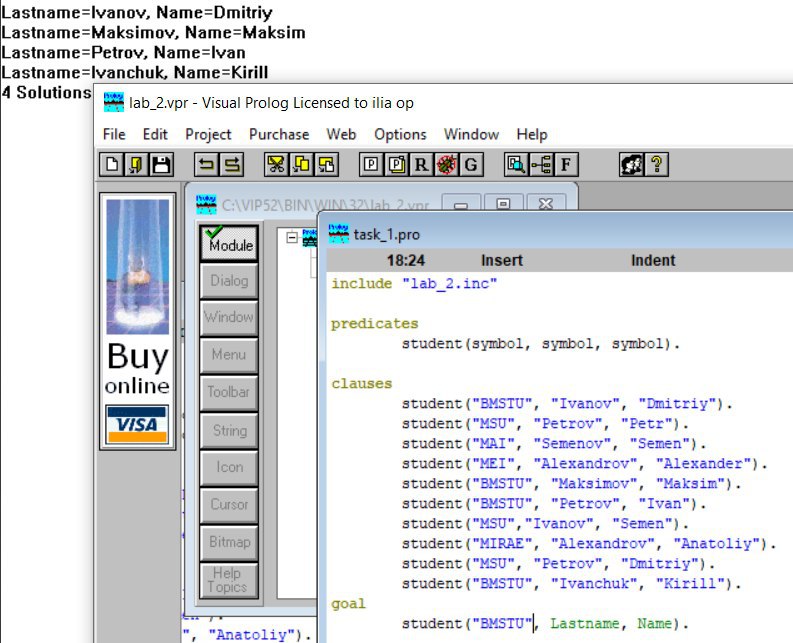
goal

student("MSU","Maksimov", "Maksim").

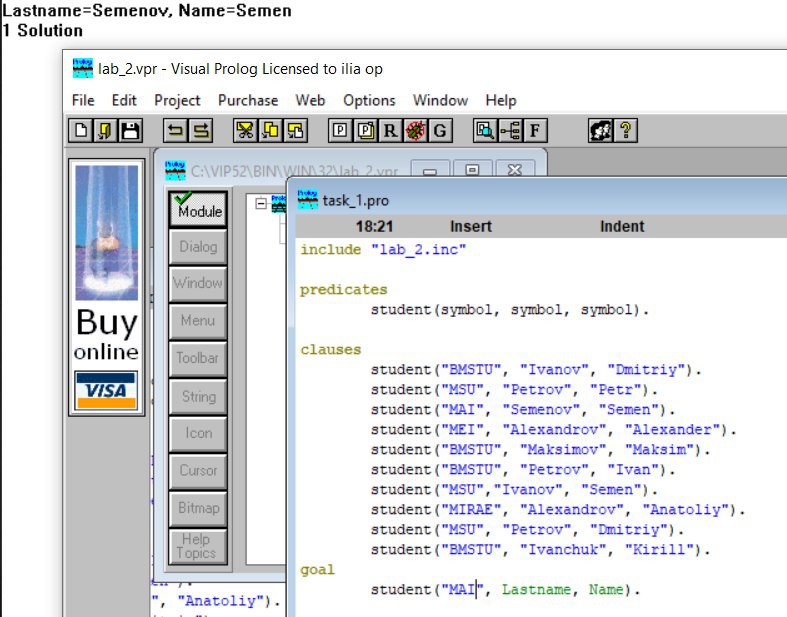
Листинг 1. База знаний студентов

Тестирование:

1. При вводе названия университета будут получены имена и фамилии студентов, учащихся в этом университете.

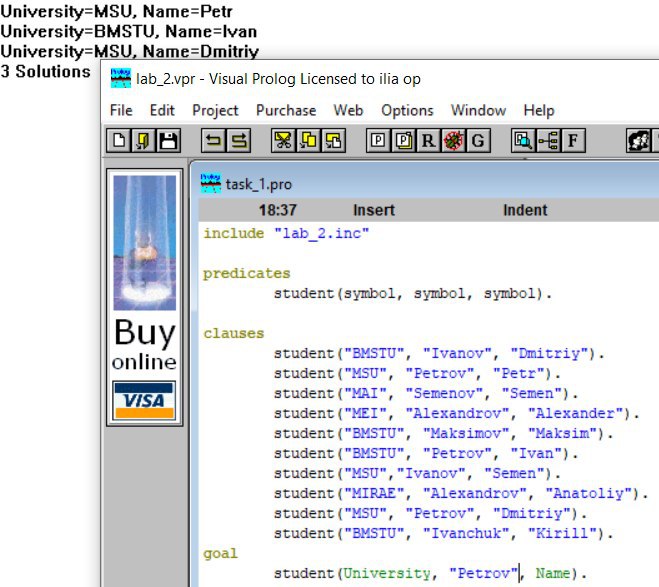


Проверка с несколькими результатами

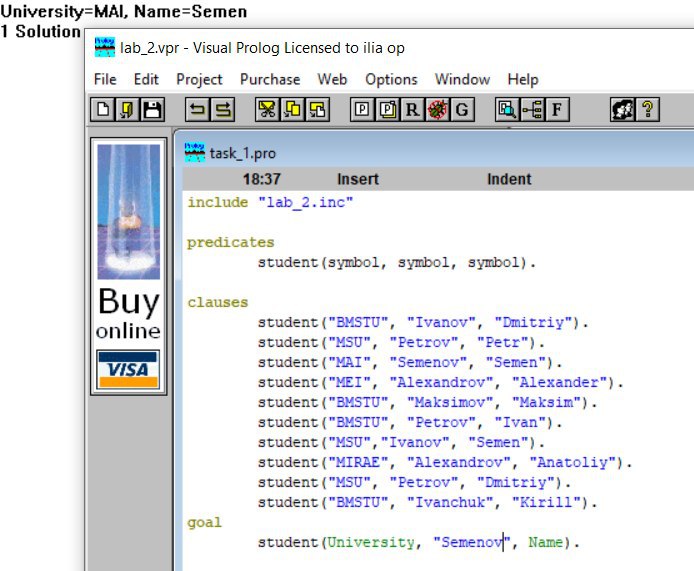


Проверка с одним результатом

1. При вводе фамилии студента будут получены имена студентов и названия университетов, в которых учатся студенты с введённой фамилий.

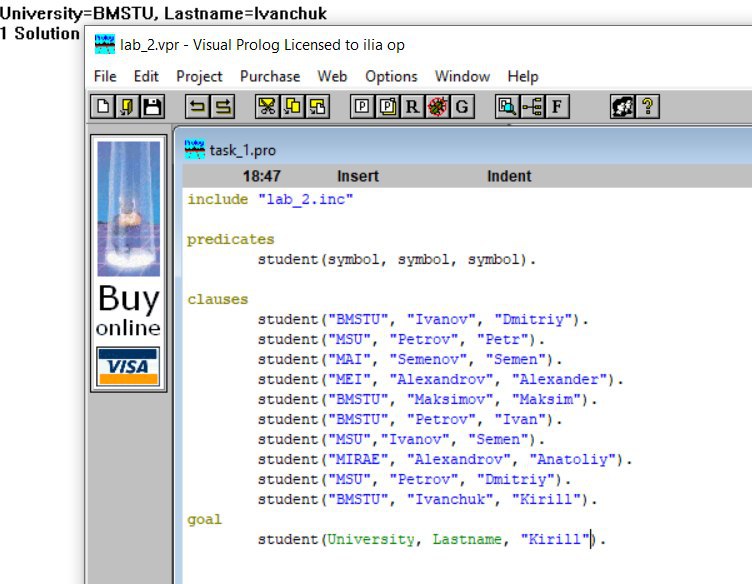


Проверка с несколькими решениями

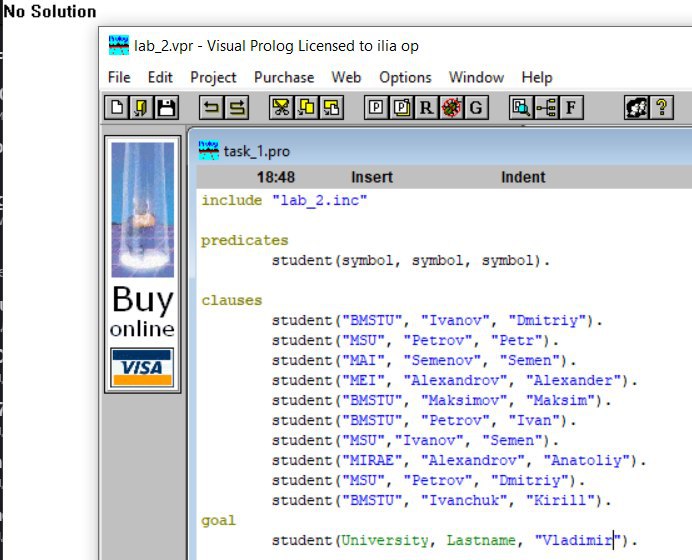


Проверка с одним решением

1. При вводе имени студента будет получены фамилии студентов и названия университетов, в которых учатся студенты с введённым именем.

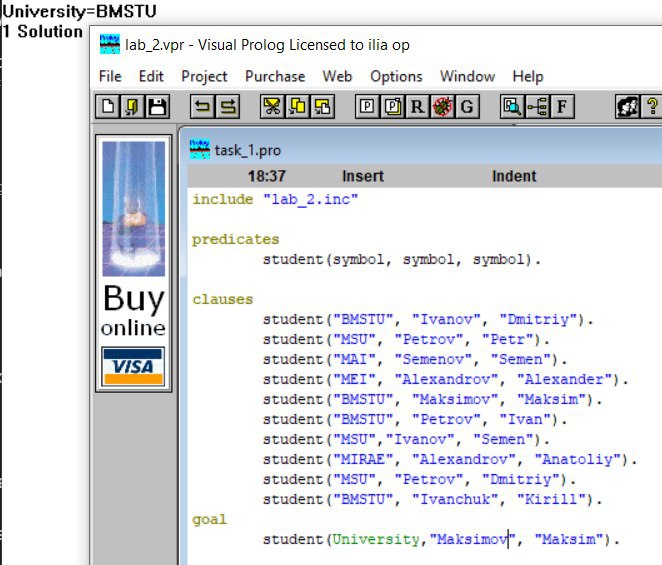


Проверка с одним решением



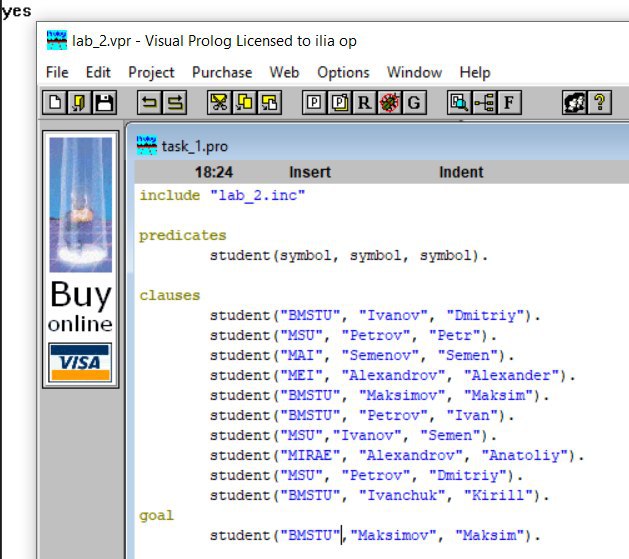
Проверка при отсутствии решения

1. При вводе имени и фамилии студента будут получены университеты, в которых учатся студенты с введёнными именем и фамилией;

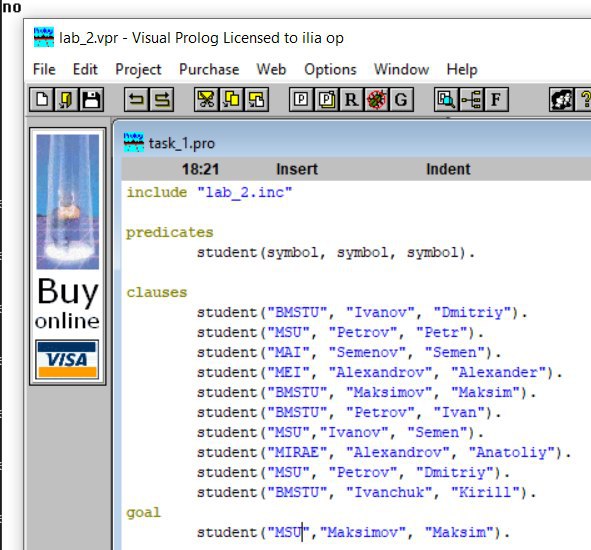


Проверка с одним решением

1. При вводе имени, фамилии и названия университета будет выведена информация о наличии такого студента в базе знаний:



Проверка с наличием студента



Проверка с отсутствием студента

2. База знаний сформированная с использованием правила:

Назначение переменных:

* university – название университета;
* lastname – фамилия студента;
* name – имя студента;

include "lab\_2.inc"

predicates

student(symbol, symbol, symbol).

getInfo(symbol, symbol, symbol).

clauses

student("BMSTU", "Ivanov", "Dmitriy").

student("MSU", "Petrov", "Petr").

student("MAI", "Semenov", "Semen").

student("MEI", "Alexandrov", "Alexander").

student("BMSTU", "Maksimov", "Maksim").

student("BMSTU", "Petrov", "Ivan").

student("MSU","Ivanov", "Semen").

student("MIRAE", "Alexandrov", "Anatoliy").

student("MSU", "Petrov", "Dmitriy").

student("BMSTU", "Ivanchuk", "Kirill").

getInfo(University, Lastname, Name) :- student(University, Lastname, Name).

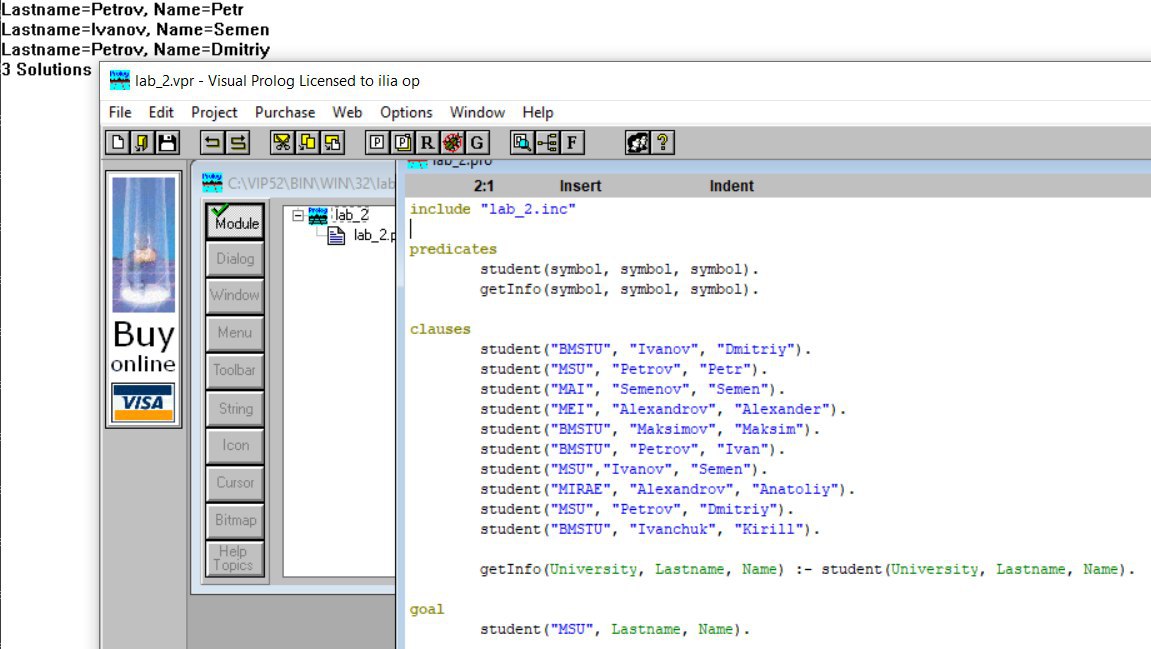
goal

student("MAI", "Semenov", "Ivan").

Листинг 2. База знаний сформированная с использованием правила

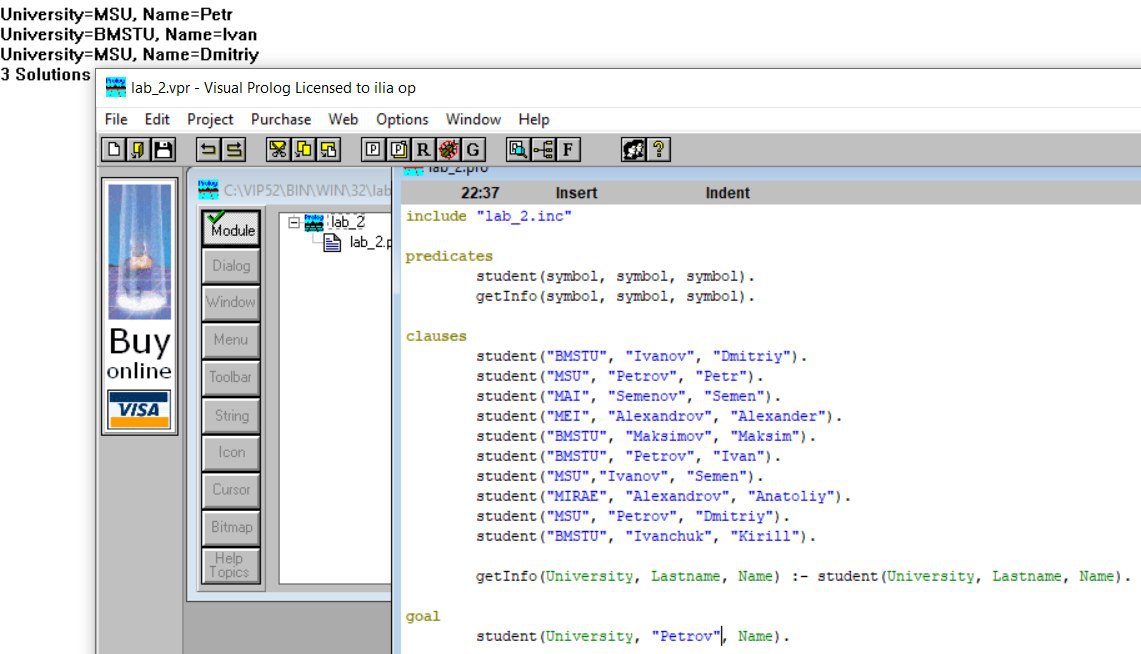
Тестирование:

1. При вводе названия университета будут получены имена и фамилии студентов, учащихся в этом университете.



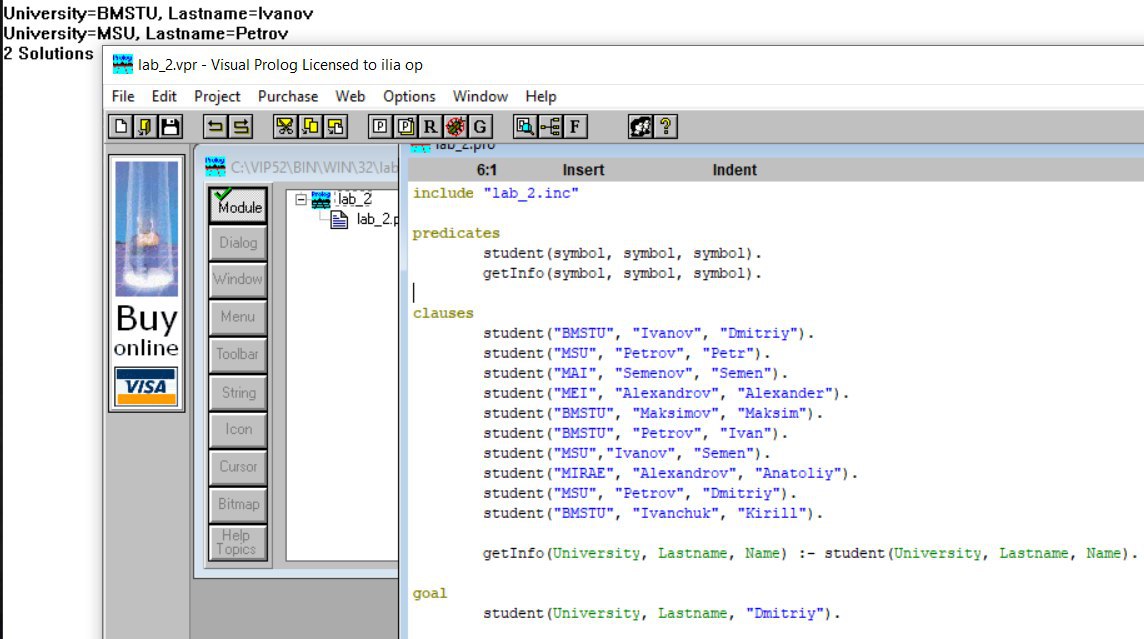
Проверка с несколькими решениями

1. При вводе фамилии студента будут получены имена студентов и названия университетов, в которых учатся студенты с введённой фамилий.

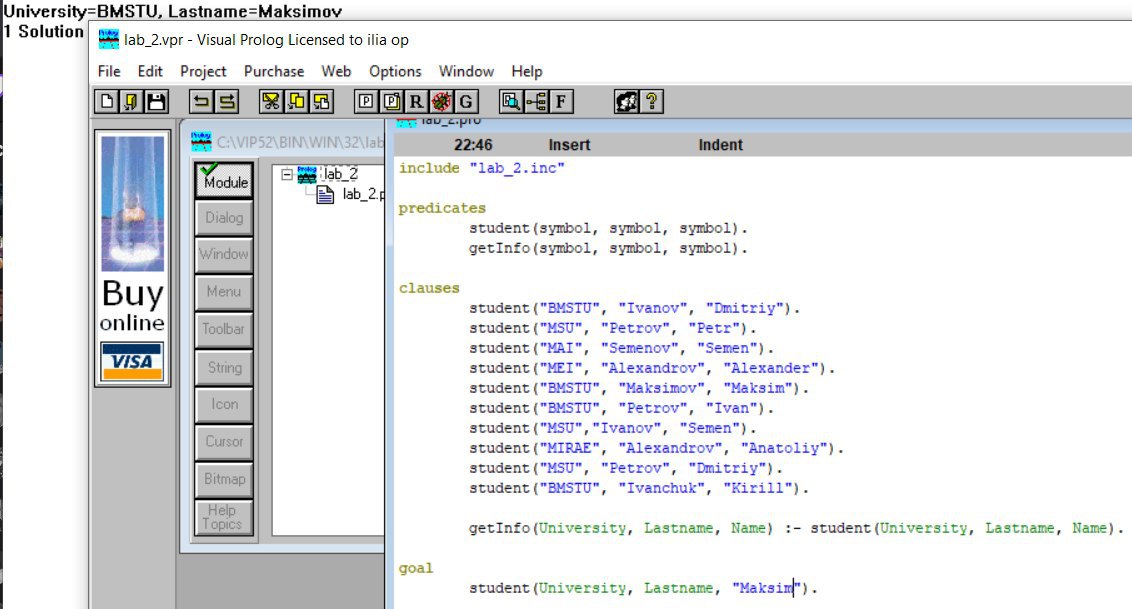


Проверка с несколькими решениями

1. При вводе имени студента будет получены фамилии студентов и названия университетов, в которых учатся студенты с введённым именем.

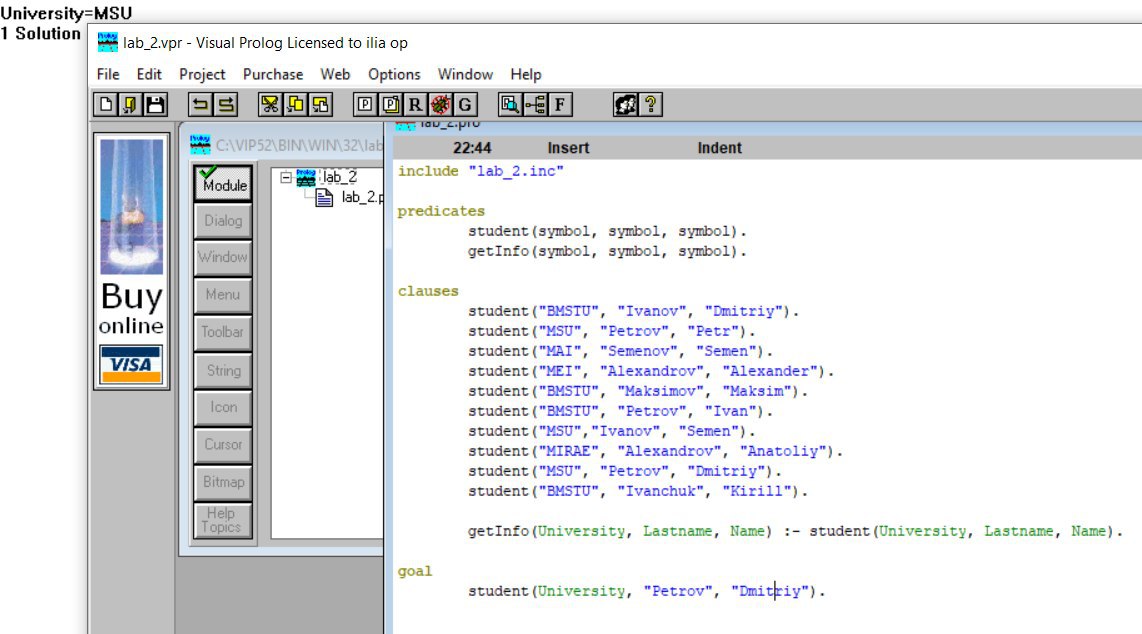


Проверка с двумя решениями;

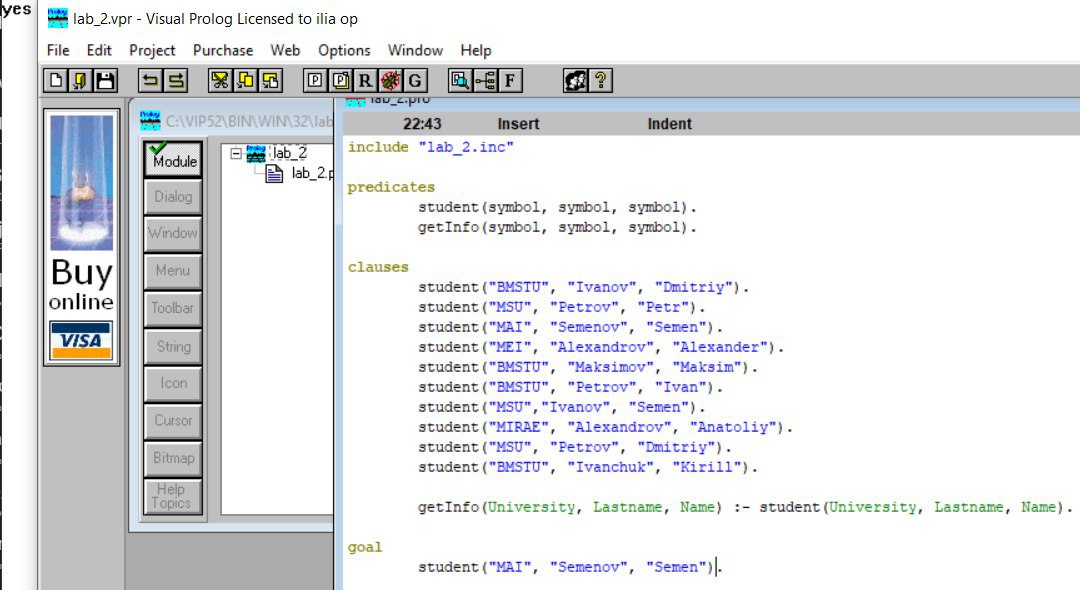


Проверка с одним решением;

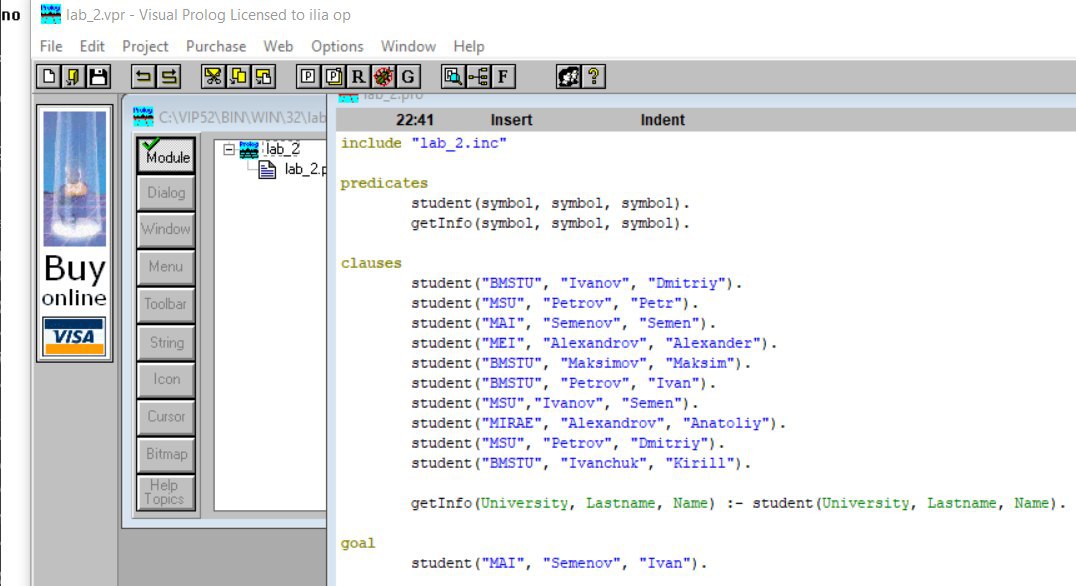
1. При вводе имени и фамилии студента будут получены университеты, в которых учатся студенты с введёнными именем и фамилией;



1. При вводе имени, фамилии и названия университета будет выведена информация о наличии такого студента в базе знаний:



Тест с существующим студентом в базе знаний



Тест с несуществующим студентом в базе знаний

3. База знаний на произвольную тему:

Назначение переменных:

* brand – марка автомобиля;
* model – модель автомобиля;
* color – цвет автомобиля;
* price – цена автомобиля;

domains

brand, model, color = string.

price = integer.

predicates

car(brand, model, color, price).

getCar(brand, model, color, price).

clauses

car("tayota", "corolla", "red", 300000).

car("kia", "rio", "blue", 250000).

car("hyundai", "accent", "black", 400000).

car("volkswagen", "polo", "silver", 410000).

car("ford", "focus", "yellow", 500000).

car("opel", "zafira", "black", 600000).

car("skoda", "octavia", "green", 610000).

car("ford", "mondeo", "yellow", 550000).

car("mercedes", "eklasse", "green", 800000).

car("audi", "a6", "black", 900000).

car("audi", "a8", "yellow", 1500000).

car("porsche", "panamera", "blue", 2000000).

getCar(Brand, Model, Color, Price) :- car(Brand, Model, Color, Price).

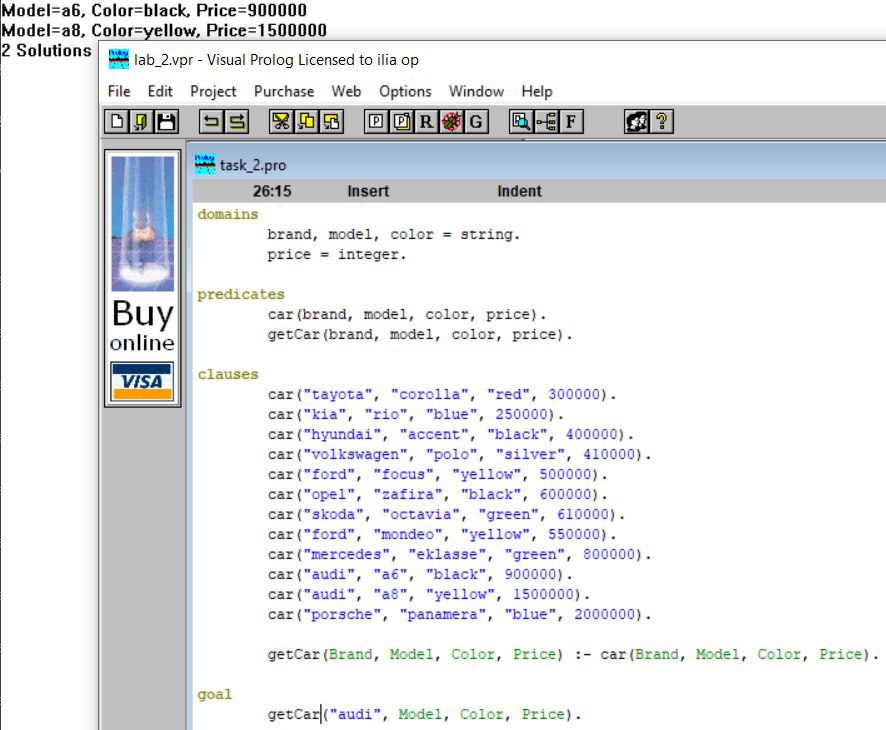
goal

getCar(Brand, Model, "yellow", 550000).

Листинг 3. База знаний на произвольную тему

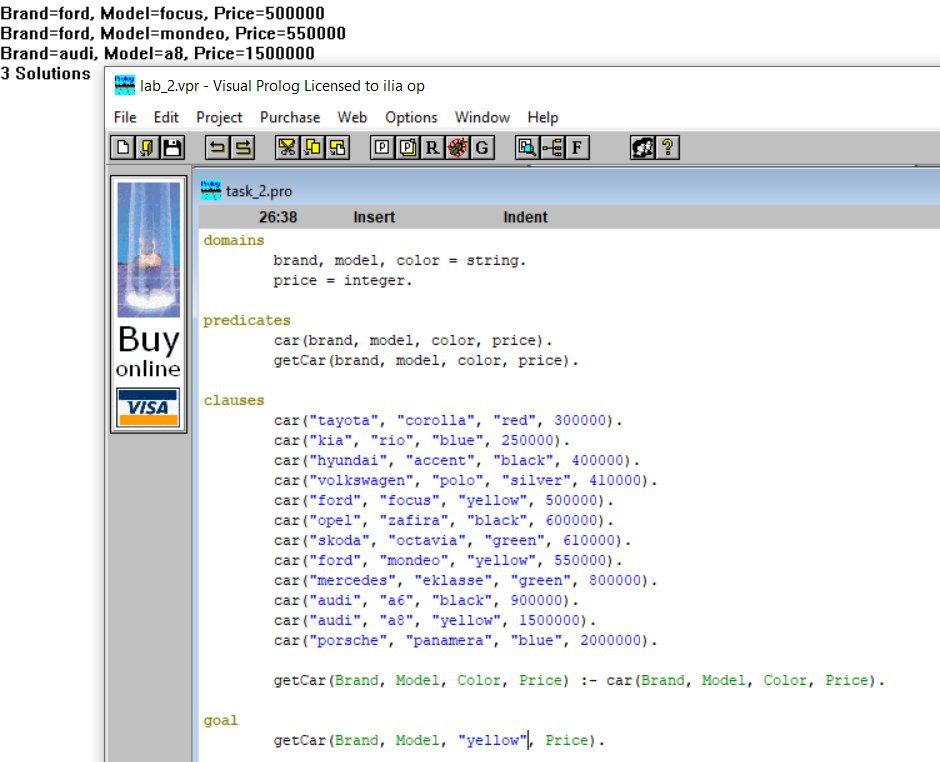
Тестирование:

1. Если ввести марку автомобиля, то будет получена информация об автомобилях этой марки (модель, цвет и цена):

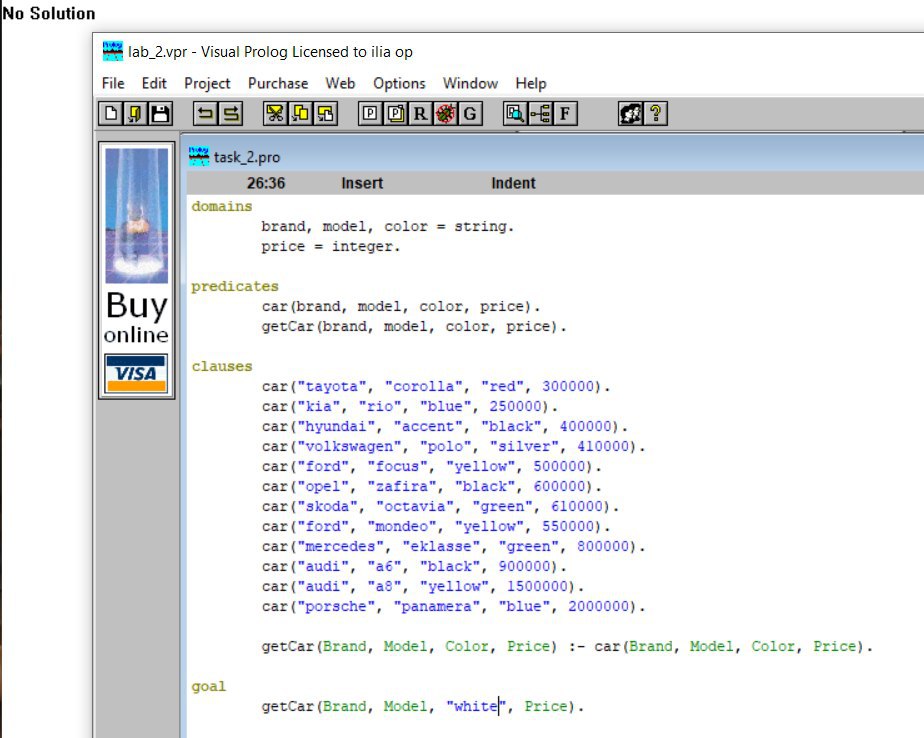


Проверка с несколькими решениями

1. Если ввести цвет автомобиля, то будет получена информация об автомобилях этого цвета (марка, модель и цена):

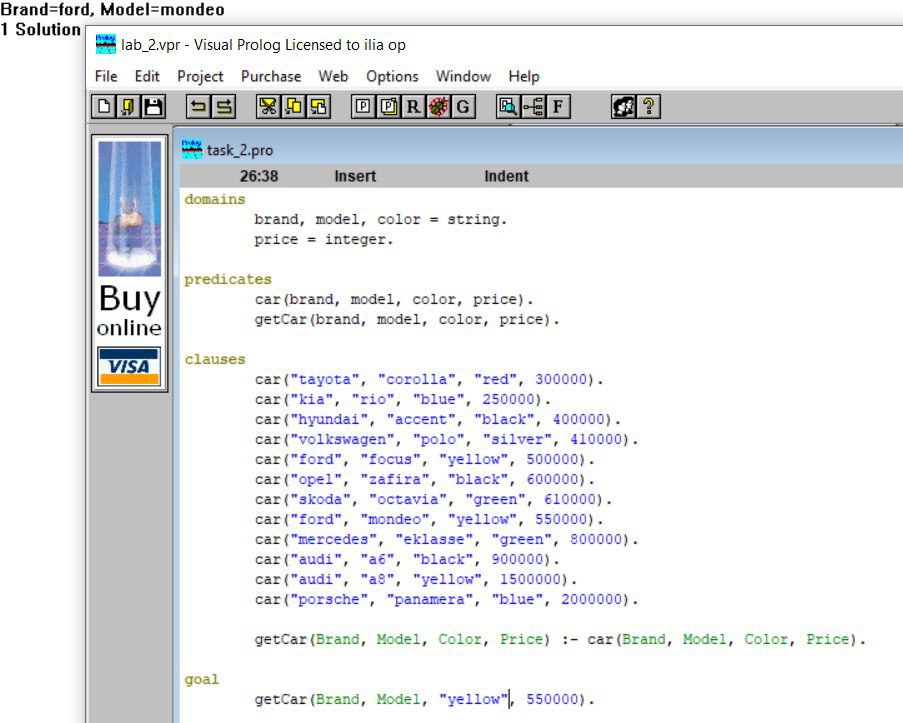


Проверка с несколькими решениями



Проверка с отсутствием решения

1. Если ввести цвет и цену автомобиля, то будет получена информация об автомобилях этого цвета и цены (марка и модель):



Проверка с одним решением