



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 предмету: «Функциональное и логическое  
программирование»*

Студент: Лаврова А. А.,  
Группа: ИУ7-65Б  
Преподаватель: Толпинская Н. Б.

Москва, 2020 г.

## Практическая часть

Листинг программы:

domains

```
name, university = symbol.
```

predicates

```
student(name, university).
```

```
one_university(name, university).
```

```
one_student(name, university).
```

clauses

```
student("Lavrova Anastasia", "BMSTU").
```

```
student("Latypov Timur", "SUSU").
```

```
student("Bukov Anton", "ITMO").
```

```
student("Kondrashova Olga", "BMSTU").
```

```
student("Ochneva Ekaterina", "CSU").
```

```
student("Ochneva Ekaterina", "ITMO").
```

```
one_university(Name, University) :- student(X, University),  
student(Name, University), X <> Name.
```

```
one_student(Name, University) :- student(Name, X), student(Name,  
University), X <> University.
```

goal

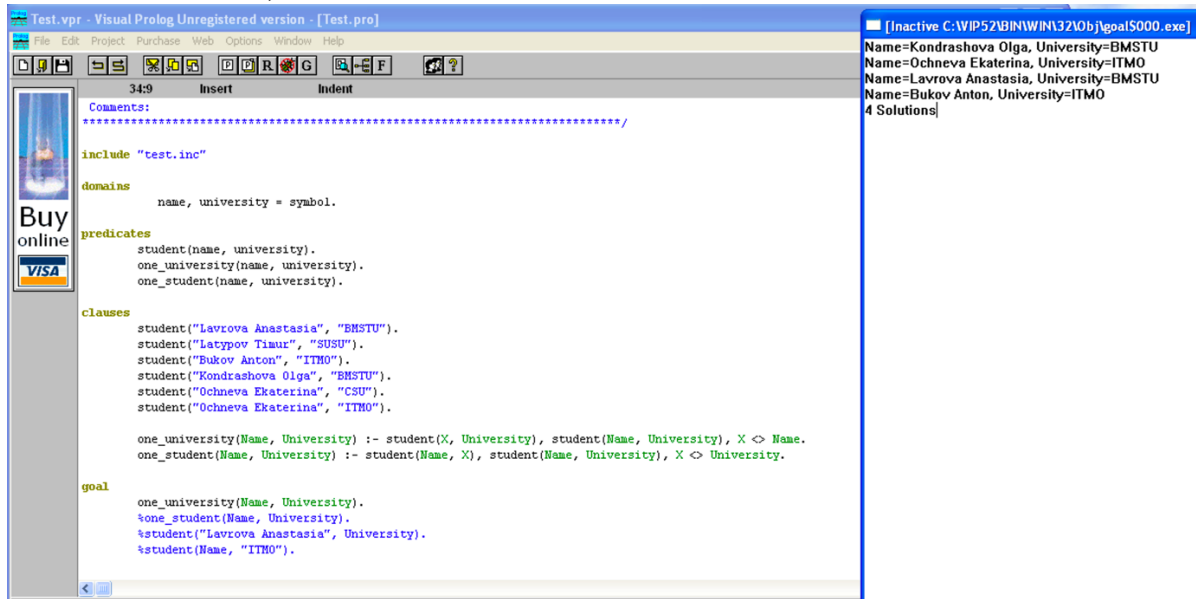
```
%one_university(Name, University).
```

```
%one_student(Name, University).
```

```
%student("Lavrova Anastasia", University).
```

```
student(Name, "ITMO").
```

Пример вывода первого вопроса (студенты и университеты, где учится более одного человека):



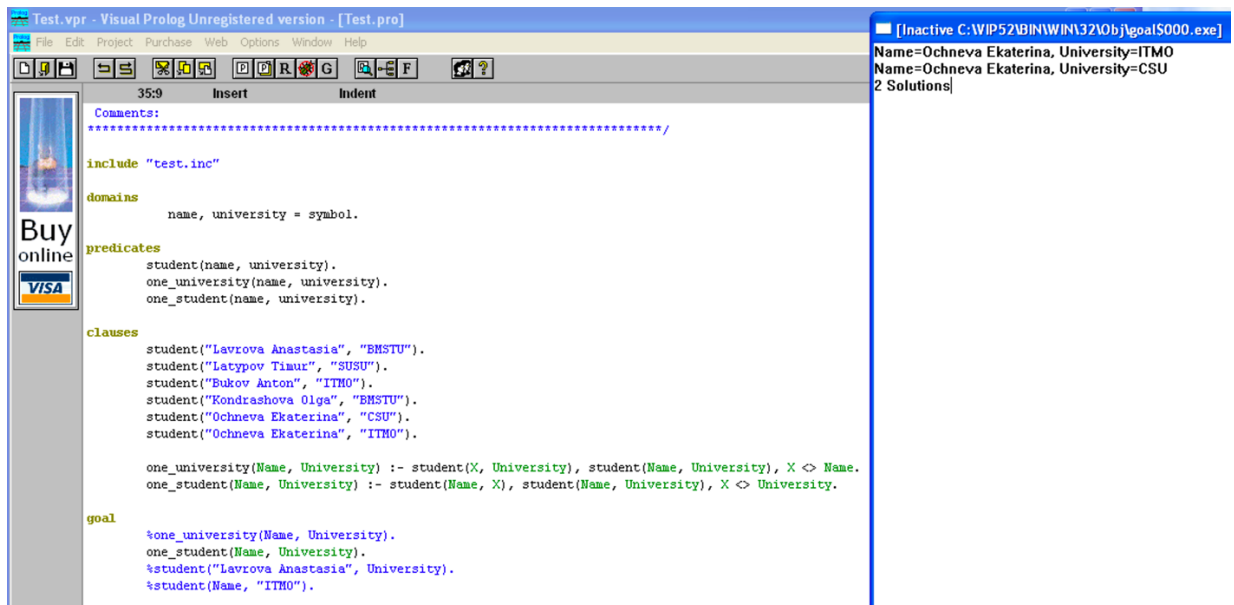
The screenshot shows the Visual Prolog IDE with a Prolog program in the main editor and its output in the right-hand pane. The program defines a set of students and universities and a goal to find universities where more than one student is enrolled. The output pane shows the results of the first query, listing three universities: BMSTU, ITMO, and CSU, each with the names of the students enrolled there.

```
Test.vpr - Visual Prolog Unregistered version - [Test.pro]
File Edit Project Purchase Web Options Window Help
34:9 Insert Indent
Comments:
*****/
include "test.inc"
domains
    name, university = symbol.
predicates
    student(name, university).
    one_university(name, university).
    one_student(name, university).
clauses
    student("Lavrova Anastasia", "BMSTU").
    student("Latypov Timur", "SUSU").
    student("Bukov Anton", "ITMO").
    student("Kondrashova Olga", "BMSTU").
    student("Ochneva Ekaterina", "CSU").
    student("Ochneva Ekaterina", "ITMO").

    one_university(Name, University) :- student(X, University), student(Name, University), X <> Name.
    one_student(Name, University) :- student(Name, X), student(Name, University), X <> University.
goal
    one_university(Name, University).
    %one_student(Name, University).
    %student("Lavrova Anastasia", University).
    %student(Name, "ITMO").
```

[Inactive C:\WP52\BIN\WIN32\Obj\goal\$000.exe]  
Name=Kondrashova Olga, University=BMSTU  
Name=Ochneva Ekaterina, University=ITMO  
Name=Lavrova Anastasia, University=BMSTU  
Name=Bukov Anton, University=ITMO  
4 Solutions

Пример вывода второго вопроса (человек, который учится в двух университетах одновременно):



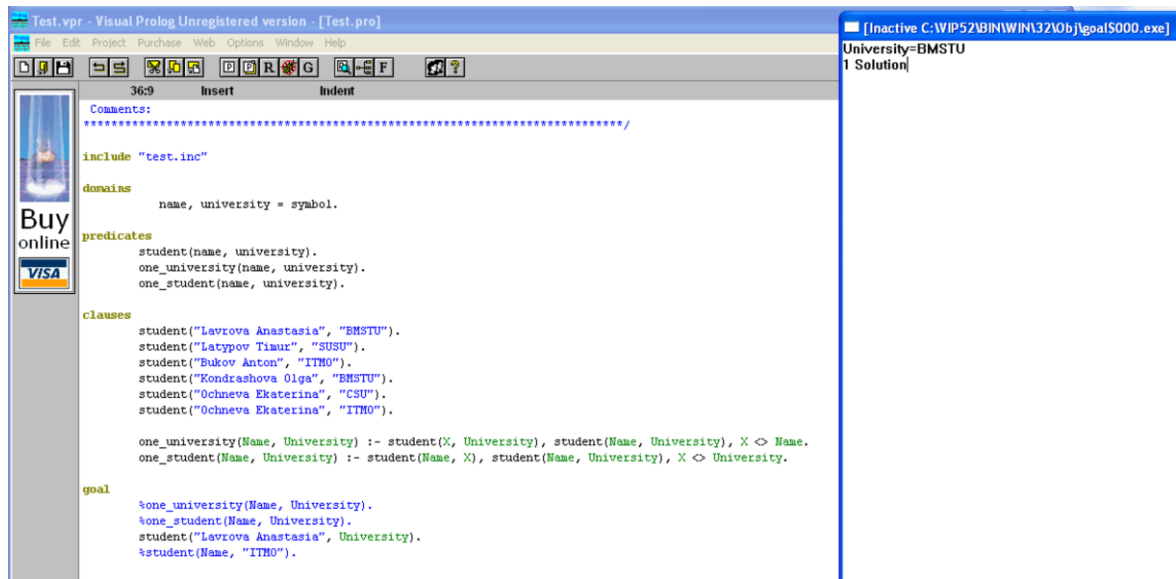
The screenshot shows the Visual Prolog IDE with the same Prolog program as the first image. The output pane shows the results of the second query, which finds individuals enrolled in more than one university. The output lists two individuals: Ochneva Ekaterina, who is enrolled in both ITMO and CSU, and Lavrova Anastasia, who is enrolled in both BMSTU and ITMO.

```
Test.vpr - Visual Prolog Unregistered version - [Test.pro]
File Edit Project Purchase Web Options Window Help
35:9 Insert Indent
Comments:
*****/
include "test.inc"
domains
    name, university = symbol.
predicates
    student(name, university).
    one_university(name, university).
    one_student(name, university).
clauses
    student("Lavrova Anastasia", "BMSTU").
    student("Latypov Timur", "SUSU").
    student("Bukov Anton", "ITMO").
    student("Kondrashova Olga", "BMSTU").
    student("Ochneva Ekaterina", "CSU").
    student("Ochneva Ekaterina", "ITMO").

    one_university(Name, University) :- student(X, University), student(Name, University), X <> Name.
    one_student(Name, University) :- student(Name, X), student(Name, University), X <> University.
goal
    %one_university(Name, University).
    one_student(Name, University).
    %student("Lavrova Anastasia", University).
    %student(Name, "ITMO").
```

[Inactive C:\WP52\BIN\WIN32\Obj\goal\$000.exe]  
Name=Ochneva Ekaterina, University=ITMO  
Name=Ochneva Ekaterina, University=CSU  
2 Solutions

Пример вывода третьего вопроса (университет, в котором учится Lavrova Anastasia):

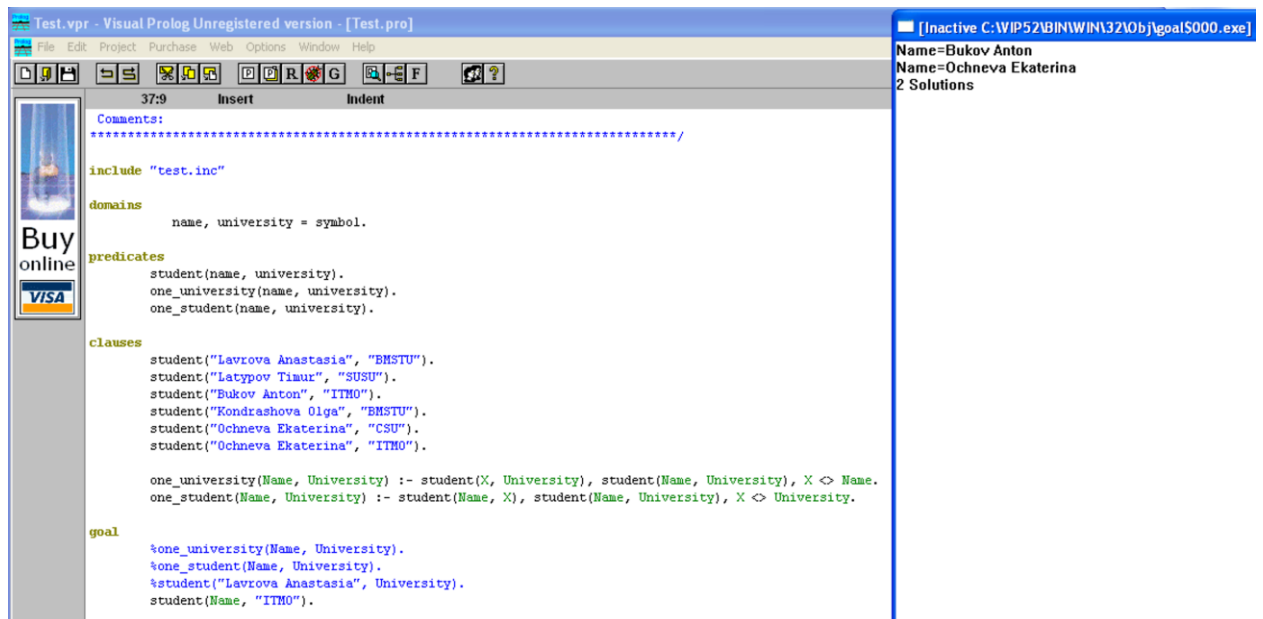


The screenshot shows the Visual Prolog IDE with the file 'Test.pro' open. The code defines a database of students and their universities. The third query is executed, and the result is displayed in the right-hand pane.

```
Test.vpr - Visual Prolog Unregistered version - [Test.pro]
File Edit Project Purchase Web Options Window Help
36:9 Insert Indent
Comments:
*****/
include "test.inc"
domains
    name, university = symbol.
predicates
    student(name, university).
    one_university(name, university).
    one_student(name, university).
clauses
    student("Lavrova Anastasia", "BMSTU").
    student("Latypov Timur", "SUSU").
    student("Bukov Anton", "ITMO").
    student("Kondrashova Olga", "BMSTU").
    student("Ochneva Ekaterina", "CSU").
    student("Ochneva Ekaterina", "ITMO").
    one_university(Name, University) :- student(X, University), student(Name, University), X <> Name.
    one_student(Name, University) :- student(Name, X), student(Name, University), X <> University.
goal
    %one_university(Name, University).
    %one_student(Name, University).
    student("Lavrova Anastasia", University).
    %student(Name, "ITMO").
```

[[Inactive C:\WP52\BIN\WIN32\Obj\goal5000.exe]  
University=BMSTU  
1 Solution

Пример вывода четвертого вопроса (студенты, которые учатся в ITMO):



The screenshot shows the Visual Prolog IDE with the file 'Test.pro' open. The code is the same as in the previous screenshot. The fourth query is executed, and the result is displayed in the right-hand pane.

```
Test.vpr - Visual Prolog Unregistered version - [Test.pro]
File Edit Project Purchase Web Options Window Help
37:9 Insert Indent
Comments:
*****/
include "test.inc"
domains
    name, university = symbol.
predicates
    student(name, university).
    one_university(name, university).
    one_student(name, university).
clauses
    student("Lavrova Anastasia", "BMSTU").
    student("Latypov Timur", "SUSU").
    student("Bukov Anton", "ITMO").
    student("Kondrashova Olga", "BMSTU").
    student("Ochneva Ekaterina", "CSU").
    student("Ochneva Ekaterina", "ITMO").
    one_university(Name, University) :- student(X, University), student(Name, University), X <> Name.
    one_student(Name, University) :- student(Name, X), student(Name, University), X <> University.
goal
    %one_university(Name, University).
    %one_student(Name, University).
    student("Lavrova Anastasia", University).
    student(Name, "ITMO").
```

[[Inactive C:\WP52\BIN\WIN32\Obj\goal5000.exe]  
Name=Bukov Anton  
Name=Ochneva Ekaterina  
2 Solutions

В данном примере кроме фактов присутствуют два правила: «Университеты, где учатся более одного человека» и «Человек, который учится в двух университетах одновременно». Также, благодаря тому, что Prolog выдает те значения переменных, которые приводят к решению (в вопросы переменные входят только с квантором существования, а в факты и правила входят только с квантором всеобщности), мы можем задать вопросы «В каком вузе учится Лаврова Анастасия» и «Кто учится в ITMO».

## Теоретическая часть

Prolog — это декларативный язык программирования, при использовании которого решение задачи получается путем логического вывода из ранее известных положений. Программа на Prolog не является определением последовательности действий, она представляет собой набор фактов и правил, обеспечивающих получение заключений на основе этих утверждений. Prolog базируется на предложениях Хорна, являющихся подмножеством формальной системы, которая называется логикой предикатов. Prolog использует упрощенную версию синтаксиса логики предикатов – предикаты первого порядка.

Основным элементом языка является терм. Терм – это константа, переменная или составной терм.

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:

$$A :- B_1, \dots, B_n.$$

A называется заголовком правила, а  $B_1, \dots, B_n$  – телом правила.

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

Причем, A,  $B_1, \dots, B_n$  – 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.

Еще одним 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.

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:

- Двухнаправленная передача параметров процедурам,
- Неразрушающее присваивание,
- Проверка условий (доказательство).