



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический  
университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Лабораторная работа № 12

### Структура программы на Prolog

Дисциплина	Функциональное и логическое программирование
Студент	Сиденко А.Г.
Группа	ИУ7-63Б
Преподаватель	Толпинская Н.Б., Строганов Ю.В.

Москва, 2020 г.

## Задание

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.

### Программа «База данных ВУЗов»

```
1 domains
2   name, university = string.
3 predicates
4   universe(name, university).
5 clauses
6   universe("Ellen", "BMSTU").
7   universe("Ellen", "MSU").
8   universe("John", "MSU").
9   universe("Tom", "MFTI").
10  universe("Tom", University) :- universe("John", University).
11  universe("Eric", "HSE").
12  universe("Eric", "BMSTU").
13  universe("Mark", "MIFI").
14  universe("Mark", "HSE").
15  universe("Mark", University) :- universe("Tom", University).
16  universe("Bill", University) :- universe("Tom", University).
```

### Примеры работы

1. Определение в каких университетах учится Mark.

Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». В нашем случае система настроена в режим получения всех возможных вариантов ответа «Да» на поставленный вопрос.

```
1 goal
2   Name = "Mark",
3   write(Name, "_universities:_"), nl,
4   universe(Name, University).
```

```
Mark universities:
Name=Mark, University=MIFI
Name=Mark, University=HSE
Name=Mark, University=MFTI
Name=Mark, University=MSU
4 Solutions
```

## 2. Определение кто учится в BMSTU.

```
1 goal
2   University = "BMSTU",
3   write(University, "_students:_"), nl,
4   universe(Name, University).
```

```
BMSTU students:
University=BMSTU, Name=Ellen
University=BMSTU, Name=Eric
2 Solutions
```

## 3. Определение учится ли Bill в BMSTU.

В данном варианте раздела GOAL система не найдет возможность ответить «Да». В результате полученный ответ.

```
1 goal
2   Name = "Bill",
3   University = "MSTU",
4   write(Name, "_is_student_", University, "?"), nl,
5   universe(Name, University).
```

```
Bill is student MSTU?
No Solution
```

## Ответы на вопросы

что собой представляет программа на Prolog, какова ее структура. Как она реализуется, как формируются результаты работы программы.

**Программа на Prolog представляет собой:** базу знаний и вопрос. База знаний содержит истинностные знания, используя которые программа выдает ответ на запрос.

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

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

## Структура программы

1. директивы компилятора – зарезервированные символьные константы
2. CONSTANTS – раздел описания констант
3. DOMAINS – раздел описания доменов
4. DATABASE – раздел описания предикатов внутренней базы данных
5. PREDICATES – раздел описания предикатов
6. CLAUSES – раздел описания предложений базы знаний
7. GOAL – раздел описания внутренней цели (вопроса).

В программе не обязательно должны быть все разделы.

С помощью подбора ответов на запросы он (Prolog, программа) извлекает хранящуюся (известную в программе) информацию. 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».

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. Упрощенно, процесс унификации можно представить как формальный процесс сравнения терма вопроса с очередным термом знания. При этом, знания по умолчанию просматриваются сверху вниз. В процессе сравнения для переменных «подбираются», исходя из базы знаний, значения или подтверждается истинность вопроса.

### **Переменные**

При поступлении вопроса с переменной в Пролог-систему. Например:

1 `universe ("Mark" , X).`

X – переменная, входящая в вопрос, изначально является неконкретизированной. Пролог просматривает базу данных в поисках факта, сопоставимого с вопросом. Если неконкретизированная переменная появляется в качестве одного из аргументов, то Пролог считает, что такой аргумент сопоставим с любым другим аргументом, находящимся в том же факте. При обнаружении такого факта переменная X становится конкретизированной, обозначая объект, являющийся вторым аргументом найденного факта и устанавливает в этом месте маркер.

Так как, в нашем случае мы ищем все возможные варианты, именно с места отмеченного маркером Пролог начнет поиск.