

Министерство образования и науки Российской Федерации  
Омский государственный университет им. Ф.М. Достоевского

**О.Г. Чанышев**  
**ПРОграммирование в ЛОГике**  
Учебное пособие

УДК 519  
ББК 32.973 – 018я73  
Ч 188

*Рекомендовано к изданию в качестве учебного пособия  
редакционно-издательским советом ОмГУ 5 октября 2004 г.,  
протокол № 4*

Рецензенты:  
д-р физ.-мат. наук, проф., зав. кафедрой геометрии ОмГПУ *А.Н. Зубков*;  
д-р техн. наук, проф. *В.А. Филимонов*

**Чанышев О.Г.**

**Ч 188** ПРОграммирование в ЛОГике: Учеб. пособие. – Омск:  
Изд-во ОмГУ, 2004. – 64 с.

**ISBN 5-7779-0510-3**

Учебное пособие обобщает опыт автора по проведению лекций по теме "Введение в искусственный интеллект" и "Автоматический анализ текста" на математическом факультете Омского государственного университета в 2000–2004 гг. в части, касающейся программирования на языке PDC Prolog. Представлены история языка, разделы программы, основные структуры данных и методы программирования. Изложение сопровождается большим количеством примеров. Полностью приведен исходный текст, разработанной автором в учебных целях «игрушечной» экспертной системы, позволяющей пользователю формулировать вопросы на естественном русском языке.

Пособие написано с учетом Государственных образовательных стандартов высшего профессионального образования 431 ен/бак по направлению 511800 «Математика. Компьютерные науки» и 199 ен/сп по специальности «Прикладная математика и информатика».

Для студентов математического факультета.

**УДК 519**  
**ББК 32.973 – 018я73**

Изд-во  
ОмГУ

Омск  
2004

ISBN 5-7779-0510-3

© Чанышев О.Г., 2004  
© Омский госуниверситет, 2004

## ОГЛАВЛЕНИЕ

<b>Введение</b> .....	4
<b>1. История Пролога</b> .....	10
<b>2. Синтаксис и семантика Пролог-программ</b> .....	13
2.1. Объекты данных .....	13
2.2. Декларативный смысл Пролог-программ .....	14
2.3. Основные определения .....	15
<b>3. Практическое программирование на Прологе</b> .....	17
3.1. Структура Пролог-программы .....	18
3.2. Свободные и связанные переменные .....	20
3.3. Внутренняя БД Пролога .....	21
3.4. Обработка условий и организация циклов в Прологе .....	23
3.4.1. Обработка условия .....	23
3.4.2. Использование предиката типа repeat .....	24
3.5. Списки в Прологе .....	26
3.5.1. Примеры списков .....	26
3.5.2. Разделение списков на "голову" и "хвост" .....	26
3.5.3. Некоторые полезные программы для работы со списками .....	28
3.6. Ввод и вывод .....	30
3.6.1. Файловая система .....	30
3.6.2. Операции с именами файлов .....	32
3.6.3. Чтение и запись .....	33
3.7. Строки и функции работы со строками .....	34
<b>4. Простенькая экспертная система</b> .....	40
<b>5. Базовые понятия и термины Пролога</b> .....	51
5.1. Объекты .....	51
5.2. Внутренние дела Пролога .....	55
5.3. Что такое шаблоны? .....	55
5.4. Управление поиском .....	56
<b>Заключение</b> .....	58
<b>Контрольные вопросы и задания</b> .....	59
<b>Список используемой литературы</b> .....	63

## Введение

В средние века знание латинского и греческого языков являлось существенной частью образования любого ученого. Ученый, владеющий только одним языком, неизбежно чувствовал себя неполноценным, поскольку он был лишен той полноты восприятия, которая возникает благодаря возможности посмотреть на мир сразу с двух точек зрения. Таким же неполноценным ощущает себя сегодняшний исследователь в области искусственного интеллекта, если он не обладает основательным знакомством как с Лиспом, так и с Прологом – этими двумя основополагающими языками искусственного интеллекта, без знания которых невозможен более широкий взгляд на предмет исследования.

*Иван Братко*

Логическое программирование базируется на убеждении, что не человека надо обучать мышлению в терминах операций компьютера... а компьютер должен выполнять инструкции, свойственные человеку.

*Леон Стерлинг, Эхуд Шапиро*

Эволюция языков программирования – это движение от языков императивных (как делать) к языкам декларативным (что делать). Или иначе – от программирования в инструкциях компьютера к ПРОграммированию в ЛОГике. Отсюда происходит название языка – ПроЛог, к изучению которого мы и приступаем.

Для иллюстрации важнейших черт Пролога воспользуемся примером Братко [1], в котором задается вопрос базе знаний: «Кто у нас темный и большой?» (Номера перед фактами не кодируются.)

Программа:

1. большой(медведь).
2. большой(слон).
3. маленький(кот).
4. коричневый(медведь).
5. черный(кот).
6. серый(слон).

7. темный(Z):-черный(Z).  
8. темный(Z):-коричневый(Z).

Вопрос:

темный(X),большой(X).

Шаги вычисления:

1) Исходный список целевых утверждений: темный(X), большой(X).

2) Просмотр всей программы от начала к концу и поиск предложения, у которого голова сопоставима с первым целевым утверждением: темный(X).

Найдено предложение 7: темный(Z):-черный(Z).

Замена первого целевого утверждения конкретизированным телом предложения 7 – порождение нового списка целевых утверждений: черный(X), большой(X).

3) Просмотр программы для нахождения предложения, сопоставимого с черным(X). Найдено предложение 5: черный(кот). У этого предложения нет тела, поэтому список целей при соответствующей конкретизации сокращается до большой(кот).

4) Просмотр программы в поисках этой цели завершается неуспехом, и происходит возврат к шагу 3 и отмены конкретизации X=кот. Список целей вновь: черный(X), большой(X).

Продолжение просмотра ниже предложения 5. Ни одно предложение не найдено. Возврат к шагу 2 и продолжение просмотра ниже предложения 7. Найдено предложение 8: темный(Z):-коричневый(Z). Замена первой цели в списке на коричневый(X) дает: коричневый(X), большой(X).

5) Просмотр программы для обнаружения предложения, сопоставимого с коричневым(X) дает коричневый(медведь). У этого предложения нет тела, поэтому список целей уменьшается до большой(медведь).

6) Просмотр программы и обнаружение предложения большой(медведь).

У него нет тела, поэтому список целей становится пустым. Это указывает на успешное завершение, а соответствующая конкретизация переменных: X=медведь.

Забегаая немного вперед, представляю запись этой же программы на Turbo Prolog'e. (Если используете VP v.5.2, можете писать по-русски все, за исключением типов встроенных переменных, наименований встроенных предикатов, наименований разделов программы).

```
/* Раздел объявления предикатов*/
predicates
большой(string)
черный(string)
коричневый(string)
серый(string)
темный(string)
маленький(string)
/*Главная цель-вопрос*/
goal
темный(X), большой(X),
write(X).
/*Правила*/
clauses
большой(медведь).
большой(слон).
маленький(кот).
коричневый(медведь).
черный(кот).
серый(слон).
темный(Z):-черный(Z).
темный(Z):-коричневый(Z).
```

Можно изменить программу: записать факты в специальный файл (с любым именем), а затем загрузить их в раздел фактов:

```
facts – fl
большой(string)
маленький(string)
коричневый(string)
черный(string)
темный(string)
серый(string)
```

```

goal
% Читаем файл bear4.dbs в раздел facts при помощи
% встроенного предиката consult
consult("bear4.dbs",f1),
большой(X), темный(X),!,
% Если во внутренней БД существует существует объект X
% со свойствами "большой" и "темный", то печатаем
% имя этого объекта
% знак "!" – откат – есть указание Прологу прекратить про-
верки
% перейти на альтернативный вариант запроса

write(X);
% Иначе переформулируем вопрос
большой(X), коричневый(X),!,
write(X);
!.

```

Понимание того, что алгоритм – это аксиоматическое задание функции, а его выполнение – частный случай логического вывода, и привело к возникновению логического программирования и языка Пролог. Почти 30 лет прошло с момента его появления. На мой взгляд, сегодня, с появлением Visual Prolog'а пятой версии (компания PDC), Пролог – это самый технологичный язык программирования. Интерес к Прологу вообще на Западе несколько спал после неудачи проекта пятого поколения ЭВМ, однако кризис доверия быстро миновал, и сегодня он более популярен, чем прежде. В России же, если судить по программистской литературе последних лет, Пролог никому не нужен. И этому есть свое объяснение.

В Советском Союзе существовало множество высокопрофессиональных коллективов, занимающихся разработкой программного обеспечения для различных отраслей народного хозяйства.

Технология предполагала четкое разделение труда, при котором собственно программирование (кодирование) отдельных небольших модулей производилось кодировщиками на основании подробной "гостированной" технической документации, разрабо-

танной постановщиками задач под руководством главного конструктора проекта. В таком случае рабочим языком мог быть ассемблер. Итоговый программный продукт, пройдя комплексную отладку, получался и надежным, и быстрым. Пролог же был еще во многом языком экзотическим и не позволял создавать программы, столь же оптимальные по затратам памяти и быстродействию. К тому же автоматизировались задачи хорошо формализованные, при решении которых главное достоинство Пролога – логический вывод из системы аксиом – как бы затушевывалось вычислительной мощностью вечно молодого дедушки – Фортрана.

Подавляющее большинство кодировщиков не имело никакого опыта в проектировании программных систем, и, фигурально выражаясь, их любимым занятием было изобретение наиболее быстрых алгоритмов извлечения квадратного корня, которыми, как писал Дж. Мартин, "у меня заполнена вся корзина для ненужных бумаг" [3].

После распада Союза большинство организаций типа ПКБ АСУ прекратило свое существование; сильные в единстве коллективы конструкторов, постановщиков и кодировщиков распались. Наука потеряла в общественном сознании статус интеллектуального эталона, и в программистском сообществе стали задавать тон кодировщики, вооруженные философией наиболе быстрой извлечения квадратного корня.

Пролог как язык концептуального программирования – незаменимое средство индивидуальной разработки программного продукта, но для этого нужно концептуально мыслить.

Какой смысл я вкладываю в термин язык концептуального программирования?

Прежде всего это средство решения при помощи компьютера широкого класса задач таким образом, чтобы максимально большая часть нейронов мозга была занята самой задачей и следствиями, которые могут быть получены в результате ее правильного решения, а не способом описания решения для компьютера. (В конце концов, это одна из причин возникновения научного направления «искусственный интеллект».) Пролог прост, поскольку использует небольшой набор базовых механизмов, включающих сопоставление с образом и "бэктрекинг" (автоматический возврат

## Список используемой литературы

1. *Братко И.* Программирование на языке Пролог для искусственного интеллекта: Пер. с англ. М.: Мир, 1990.
2. *Стерлинг Л., Шапиро Э.* Искусство программирования на языке Пролог: Пер. с англ. М.: Мир, 1990.
3. *Мартин Дж.* Надежность программного обеспечения: Пер. с англ. М.: Мир, 1986.
4. Построение экспертных систем. М.: Мир, 1987.
5. *Лорьер Ж.-Л.* Системы искусственного интеллекта: Пер. с фр. М.: Мир, 1991.
6. *Швыркин И.* Пролог. Генезис // Мир ПК. 2000. № 5.
7. *Ин Ц., Соломон Д.* Использование Турбо-Пролога: Пер. с англ. М.: Мир, 1993.
8. Логическое программирование и Visual Prolog. СПб.: БХВ-Петербург, 2003.

Учебное издание

Олег Георгиевич Чанышев

**ПРОграммирование в ЛОГике**

Учебное пособие

Технический редактор *Н.В. Москвичёва*

Редактор *Е.В. Коськина*

---

Подписано в печать 11.11.04. Формат бумаги 60х84 1/16.

Печ. л. 4,0. Уч.-изд. л. 4,0. Тираж 100 экз. Заказ 582.

---

*Издательство Омского государственного университета  
644077, г. Омск-77, пр. Мира, 55а, госуниверситет*