



**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**

## **О т ч е т**

**по лабораторной работе № 11**

**Дисциплина: «Функциональное и логическое программирование»**

**Выполнила: Овчинникова А. П.**

**Группа: ИУ7-65Б**

**Преподаватель: Толпинская Н. Б.**

**Строганов Ю. В.**

**Москва, 2020**

## Задание

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

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.

### 2. 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 (`_`).

### 3. Составной терм:

- Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется:  $f(t_1, t_2, \dots, t_m)$ , где  $f$  - функтор (функциональный символ),  $t_1, t_2, \dots, t_m$  – 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 аргументности, обозначают разные отношения.

**Программа на Prolog представляет собой базу знаний и вопрос.** База знаний состоит из **предложений** - CLAUSES (отдельных знаний или утверждений): **фактов** и **правил**.

**Предложение** более общего вида – правило – имеет вид:

$$A :- B1, \dots, Bn.$$

$A$  называется заголовком правила, а  $B1, \dots, Bn$  – телом правила.

**Правило** – это предложение, истинность которого зависит от истинности одного или нескольких предложений. Обычно правило содержит несколько хвостовых целей, которые должны быть истинными для того, чтобы правило было истинным. Правило называют условной истиной, а факт, не содержащий тела – безусловной истиной.

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

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

**Факт** констатирует, что между объектами выполнено некоторое отношение. Факт представляет собой безусловно истинное утверждение.

**Вопрос** состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами.

В Prolog существует понятие процедуры. Процедурой называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же аргументность (местность), т. е. это совокупность правил, описывающих одно определенное отношение. Отношение, определяемое процедурой, называется предикатом.

Программа на Прологе состоит из следующих семи разделов.

- директивы компилятора;
- CONSTANTS — раздел описания констант;
- DOMAINS — раздел описания доменов;
- DATABASE — раздел описания предикатов внутренней базы данных;
- PREDICATES — раздел описания предикатов;
- CLAUSES — раздел описания предложений;
- GOAL — раздел описания внутренней цели.

В программе не обязательно должны быть все эти разделы. В программе может быть несколько разделов описаний DOMAINS, PREDICATES, DATABASE и CLAUSES. Однако разделов GOAL не может быть в программе более одного. Порядок разделов может быть произвольным, но при этом константы, домены и предикаты должны быть определены до их использования. Однако в разделе DOMAINS можно ссылаться на домены, которые будут объявлены позже.

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 быть настроена в режим получения всех возможных вариантов ответа «Да» на поставленный вопрос.

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 сравнения (сопоставления) терма вопроса с очередным термом знания. При этом знания по умолчанию просматриваются сверху вниз, хотя такой порядок и не очевиден. В процессе сравнения для переменных «подбираются», исходя из базы знаний, значения (для именованных переменных). И эти подобранные для переменных значения возвращаются в качестве побочного эффекта ответа на поставленный вопрос.

Цель работы программы – определить является ли вопрос логическим следствием программы или нет, что выполняется с применением 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:

- Двухнаправленная 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.

## Практическая часть

### 1. Настройка утилиты Test Goal

На рисунках 1 и 2 показана настройка утилиты Test Goal.

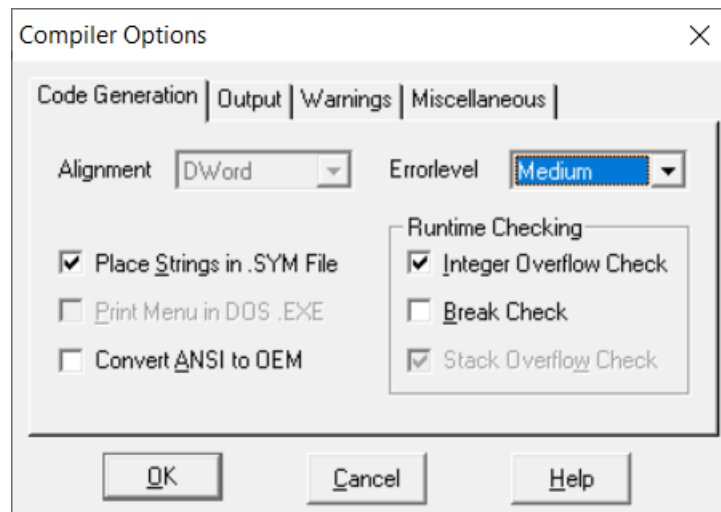


Рисунок 1. Настройка утилиты Test Goal (часть 1).

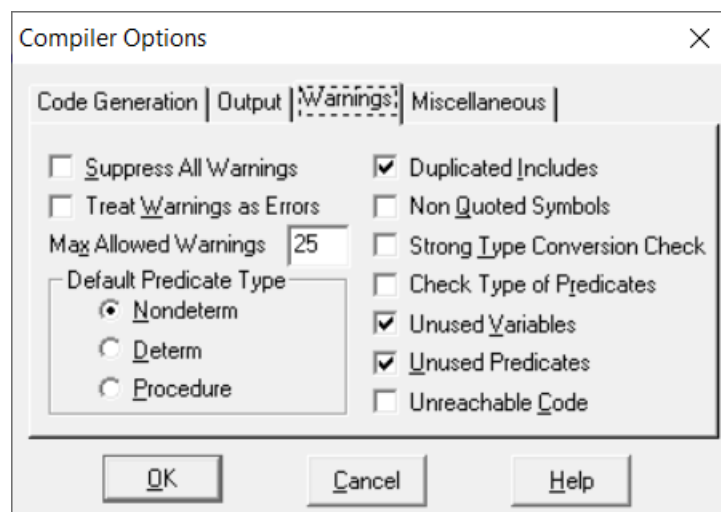


Рисунок 2. Настройка утилиты Test Goal (часть 2).

### 2. Запуск тестовой программы

На рисунках 3-4 представлен текст тестовой программы и результаты ее запуска.

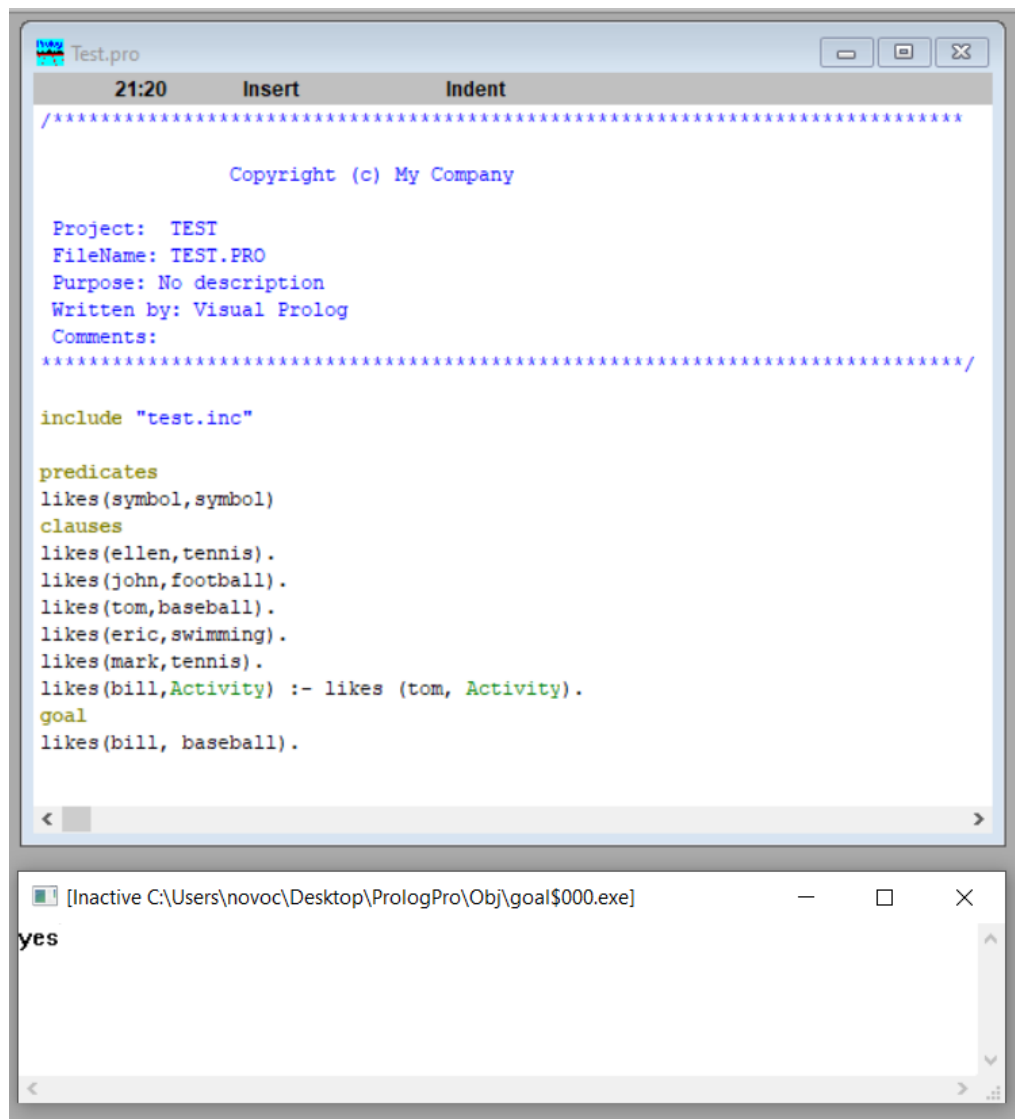


Рисунок 3. Запуск тестовой программы (часть 1).

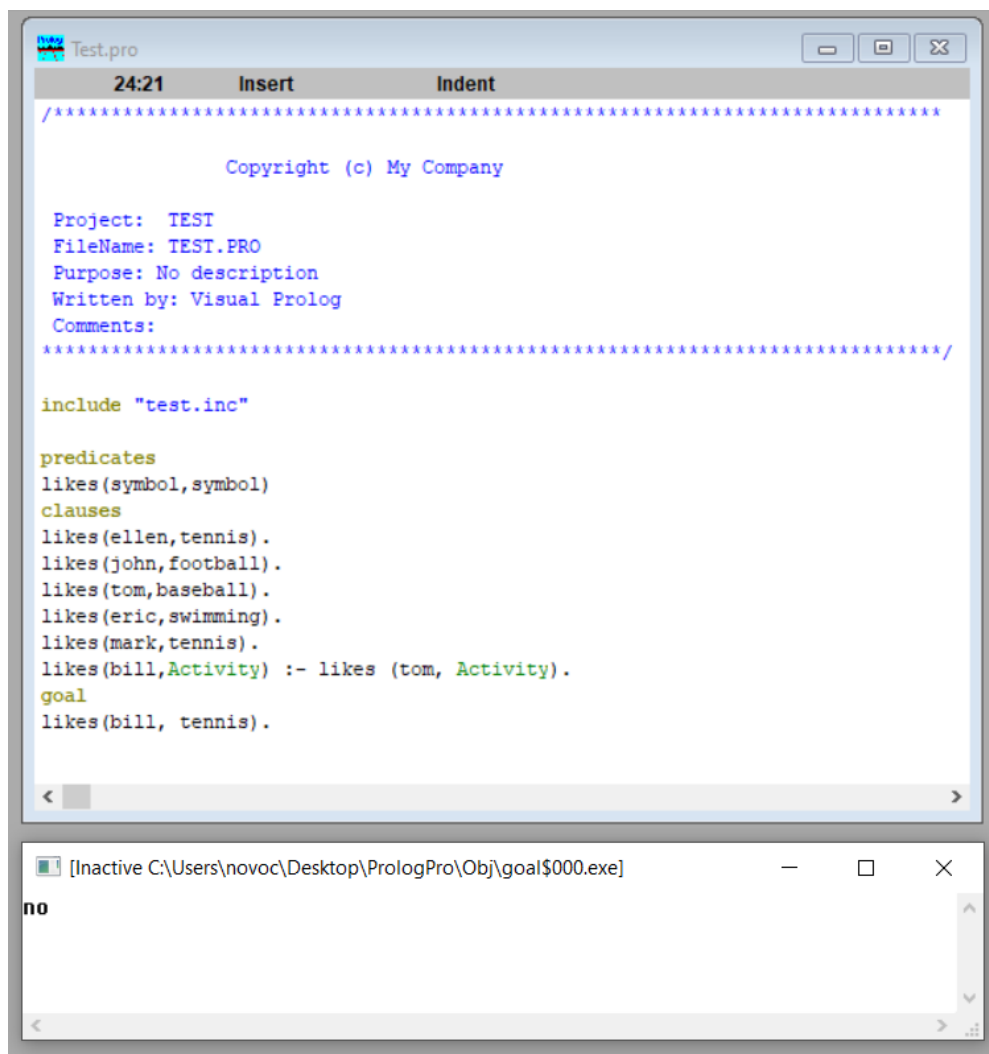


Рисунок 4. Запуск тестовой программы (часть 2).

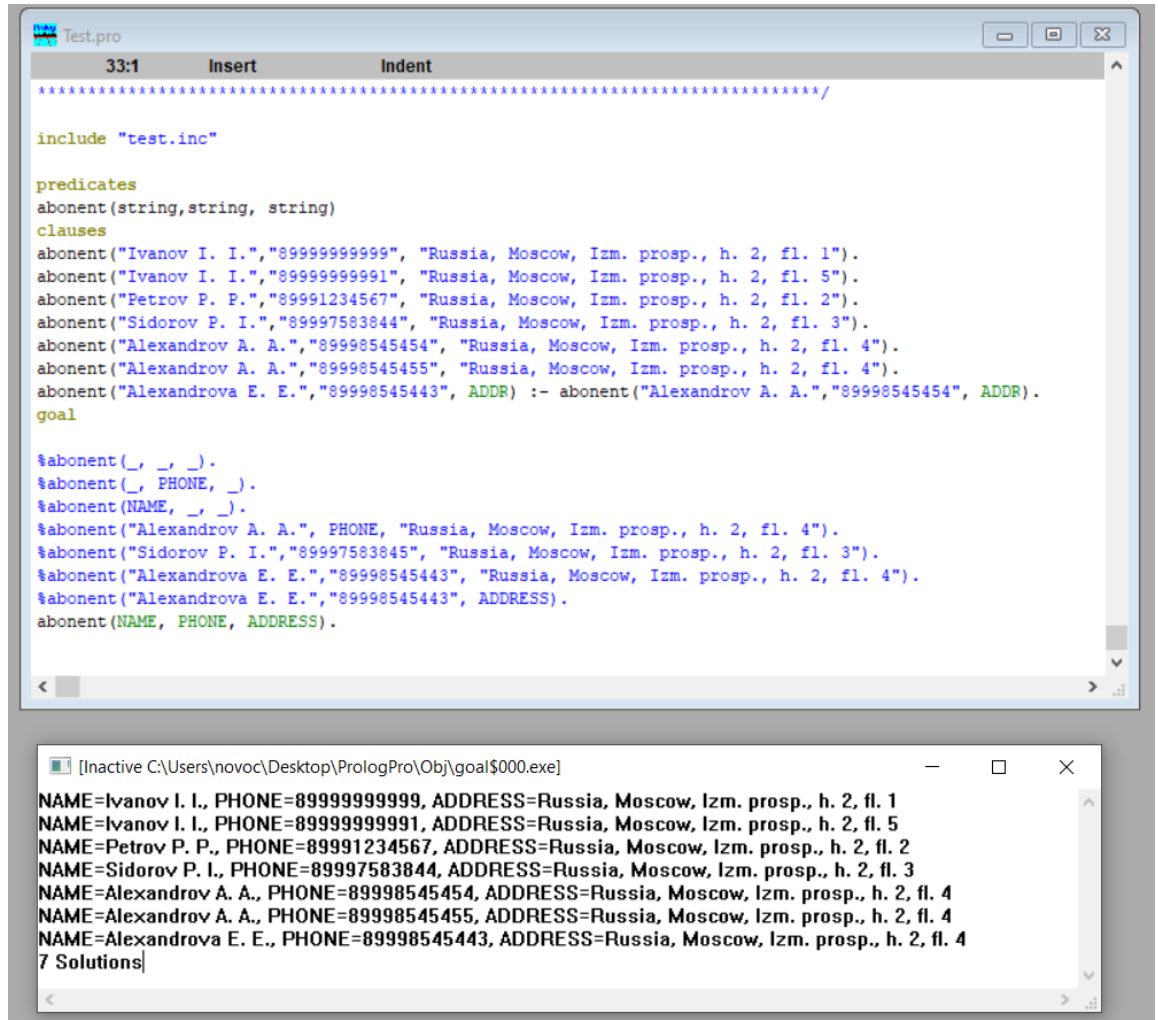
### 3. Программа - «Телефонный справочник»

На рисунках 5-12 представлен текст программы «Телефонный справочник» и результаты ее работы.

На рисунке 5 программа выдает информацию обо всех абонентах. На рисунке 6 программа выдает адрес абонента с именем "Alexandrova E. E." И номером "89998545443". На рисунке 7 программа проверяет, существует ли в базе знаний абонент с именем "Alexandrova E. E.", телефоном "89998545443" и адресом "Russia, Moscow, Izm. prosp., h. 2, fl. 4". На рисунке 8 программа проверяет, существует ли абонент с именем "Sidorov P. I.", номером "89997583845" и адресом "Russia, Moscow, Izm. prosp., h. 2, fl. 3". На рисунке 9 программа находит все номера абонента с именем "Alexandrov A. A." И адресом "Russia, Moscow, Izm. prosp., h. 2, fl. 4". На рисунке 10 программа



выводит имена всех абонентов, которые есть в базе знаний. На рисунке 11 программа выводит все номера телефонов, присутствующих в базе знаний. На рисунке 12 программа проверяет, существует ли в базе знаний хотя бы один абонент.



The image shows a screenshot of a Prolog environment. The top window, titled 'Test.pro', contains the following code:

```
33:1      Insert      Indent
*****/

include "test.inc"

predicates
abonent(string,string, string)
clauses
abonent("Ivanov I. I.", "89999999999", "Russia, Moscow, Izm. prosp., h. 2, fl. 1").
abonent("Ivanov I. I.", "89999999991", "Russia, Moscow, Izm. prosp., h. 2, fl. 5").
abonent("Petrov P. P.", "89991234567", "Russia, Moscow, Izm. prosp., h. 2, fl. 2").
abonent("Sidorov P. I.", "89997583844", "Russia, Moscow, Izm. prosp., h. 2, fl. 3").
abonent("Alexandrov A. A.", "89998545454", "Russia, Moscow, Izm. prosp., h. 2, fl. 4").
abonent("Alexandrov A. A.", "89998545455", "Russia, Moscow, Izm. prosp., h. 2, fl. 4").
abonent("Alexandrova E. E.", "89998545443", ADDR) :- abonent("Alexandrov A. A.", "89998545454", ADDR).
goal

%abonent(_, _, _).
%abonent(_, PHONE, _).
%abonent(NAME, _, _).
%abonent("Alexandrov A. A.", PHONE, "Russia, Moscow, Izm. prosp., h. 2, fl. 4").
%abonent("Sidorov P. I.", "89997583845", "Russia, Moscow, Izm. prosp., h. 2, fl. 3").
%abonent("Alexandrova E. E.", "89998545443", "Russia, Moscow, Izm. prosp., h. 2, fl. 4").
%abonent("Alexandrova E. E.", "89998545443", ADDRESS).
abonent(NAME, PHONE, ADDRESS).
```

The bottom window, titled '[Inactive C:\Users\novoc\Desktop\PrologPro\Obj\goal\$000.exe]', shows the output of the program:

```
NAME=Ivanov I. I., PHONE=89999999999, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 1
NAME=Ivanov I. I., PHONE=89999999991, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 5
NAME=Petrov P. P., PHONE=89991234567, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 2
NAME=Sidorov P. I., PHONE=89997583844, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 3
NAME=Alexandrov A. A., PHONE=89998545454, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 4
NAME=Alexandrov A. A., PHONE=89998545455, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 4
NAME=Alexandrova E. E., PHONE=89998545443, ADDRESS=Russia, Moscow, Izm. prosp., h. 2, fl. 4
7 Solutions
```

Рисунок 5. Программа «Телефонный справочник» (часть 1).

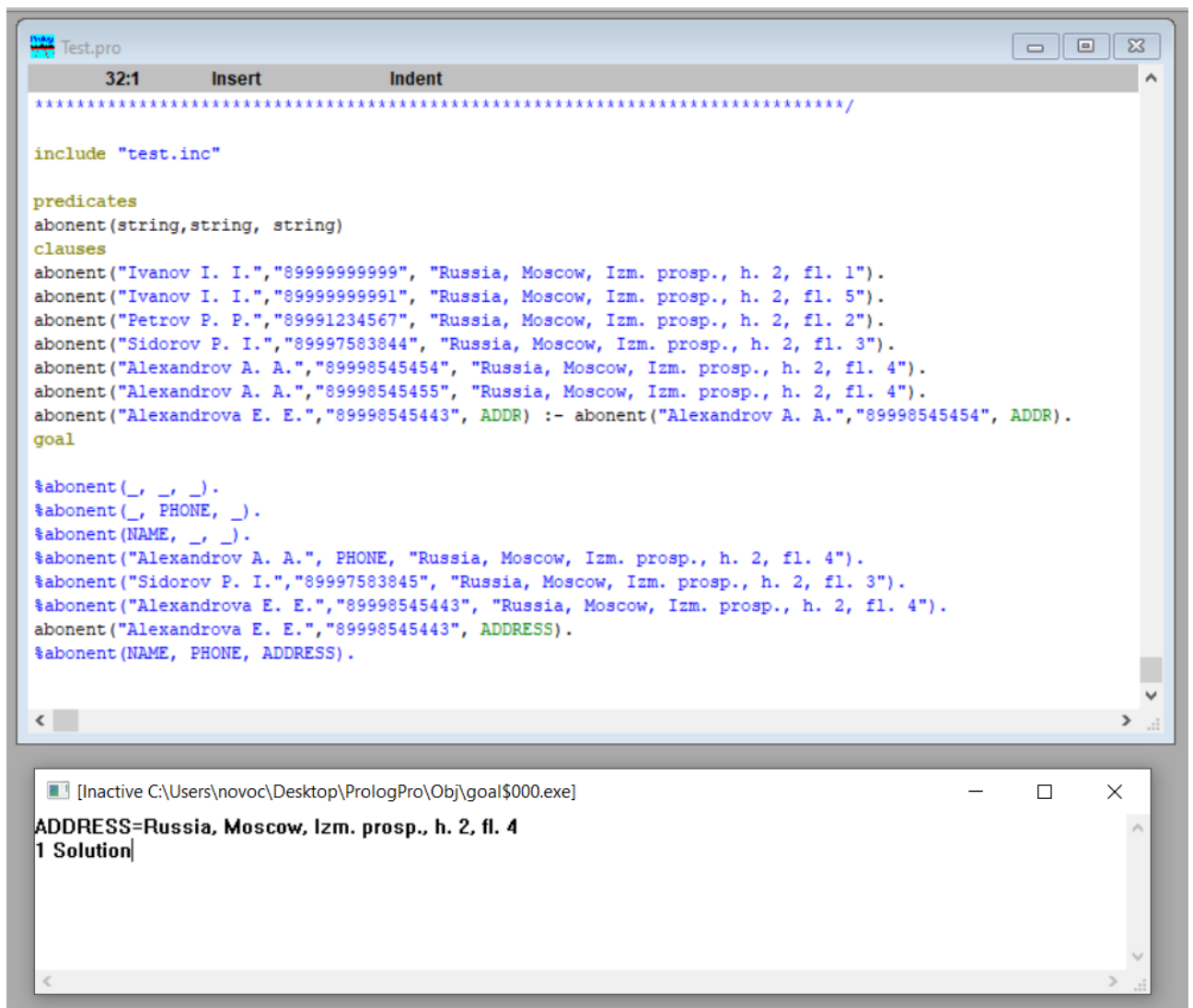


Рисунок 6. Программа «Телефонный справочник» (часть 2).

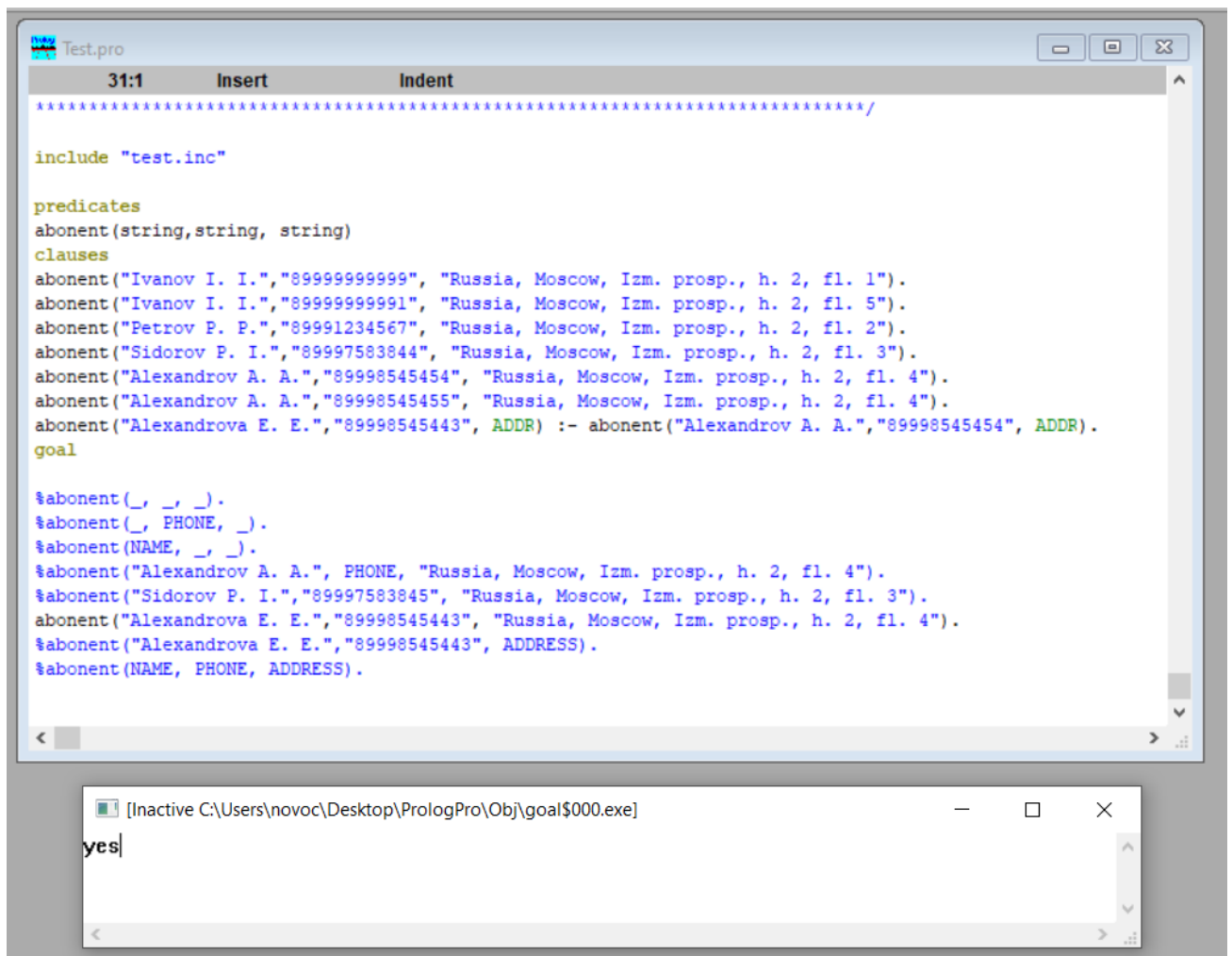


Рисунок 7. Программа «Телефонный справочник» (часть 3).

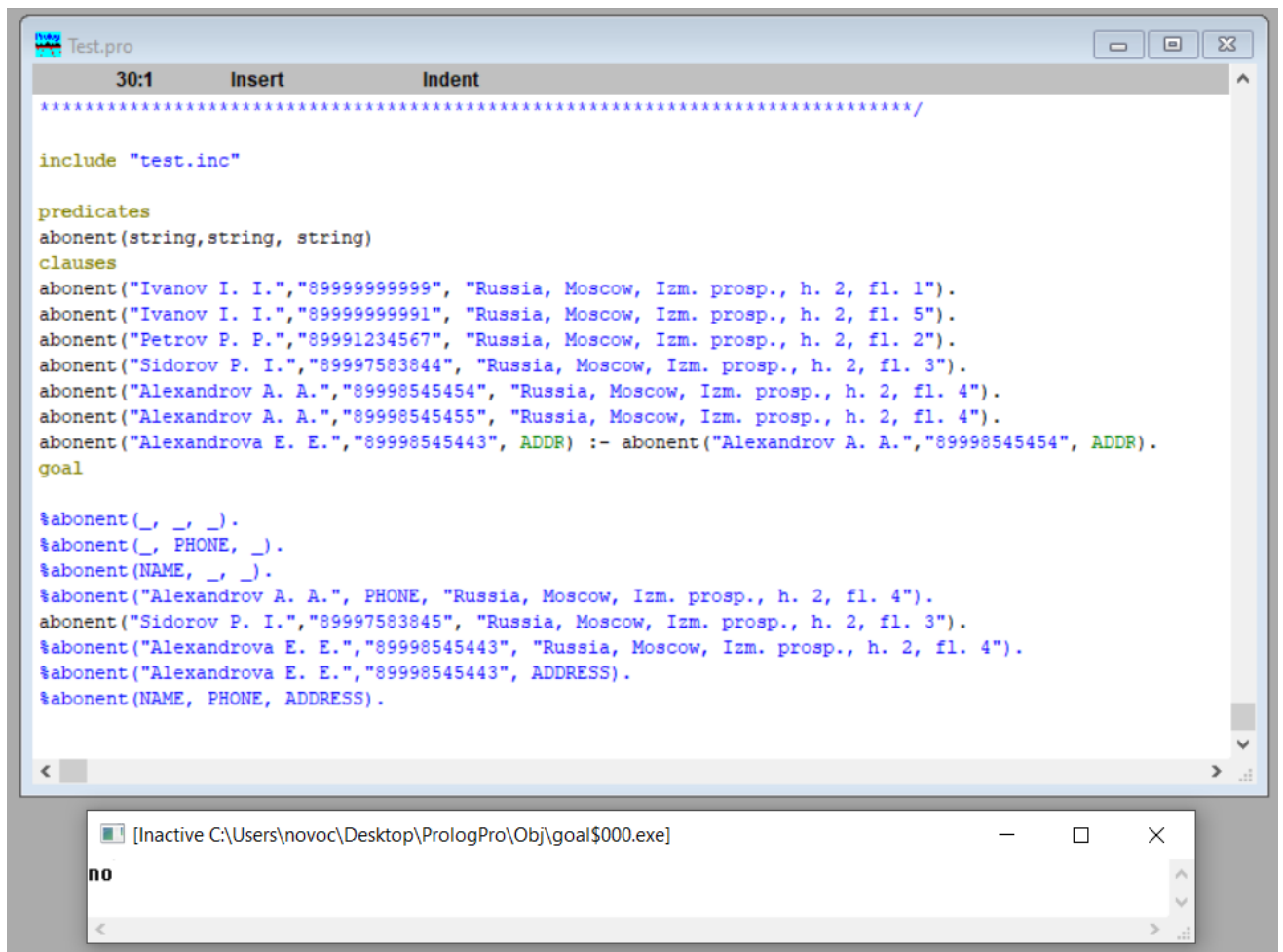


Рисунок 8. Программа «Телефонный справочник» (часть 4).

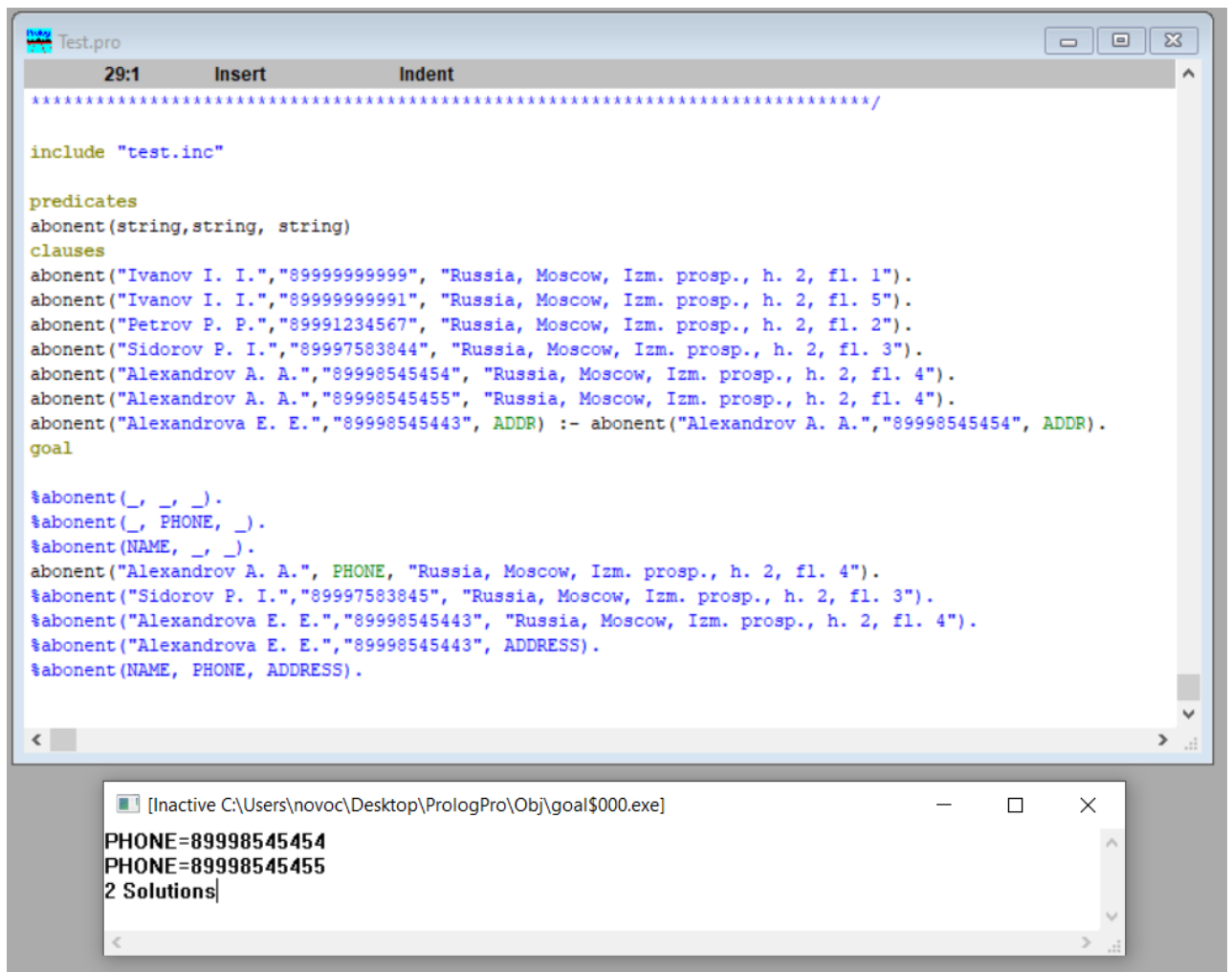


Рисунок 9. Программа «Телефонный справочник» (часть 5).

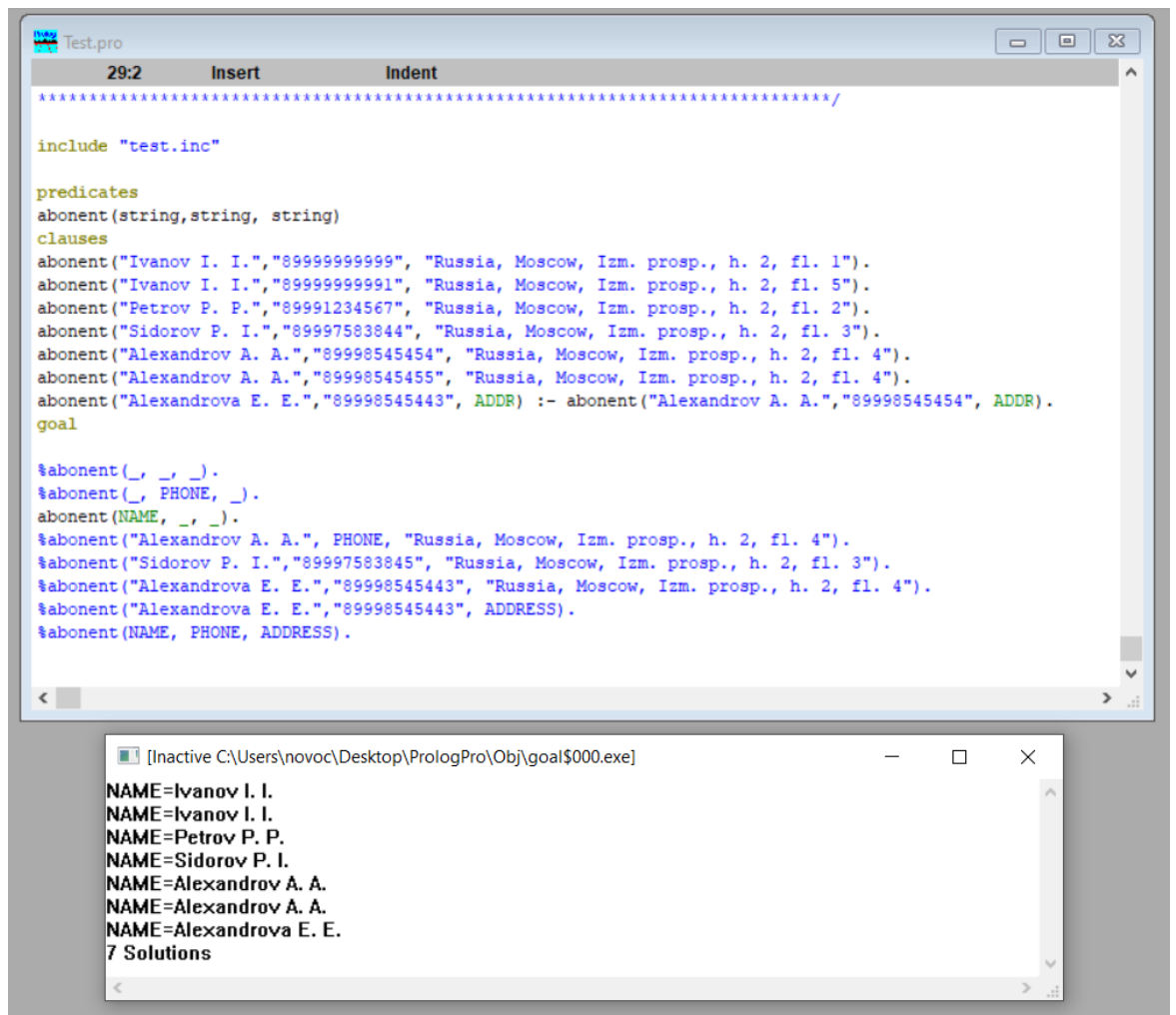


Рисунок 10. Программа «Телефонный справочник» (часть 6).

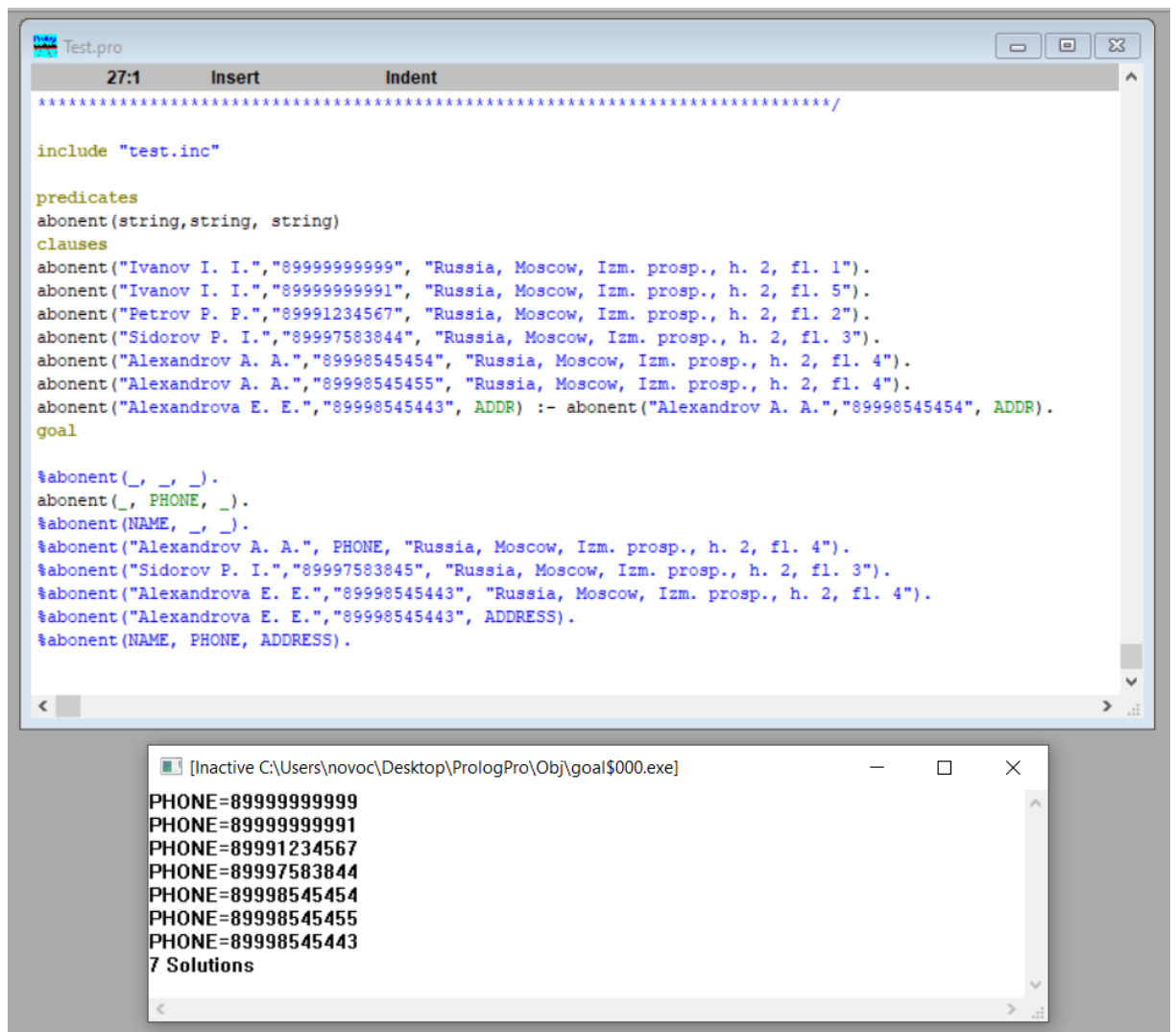


Рисунок 11. Программа «Телефонный справочник» (часть 7).

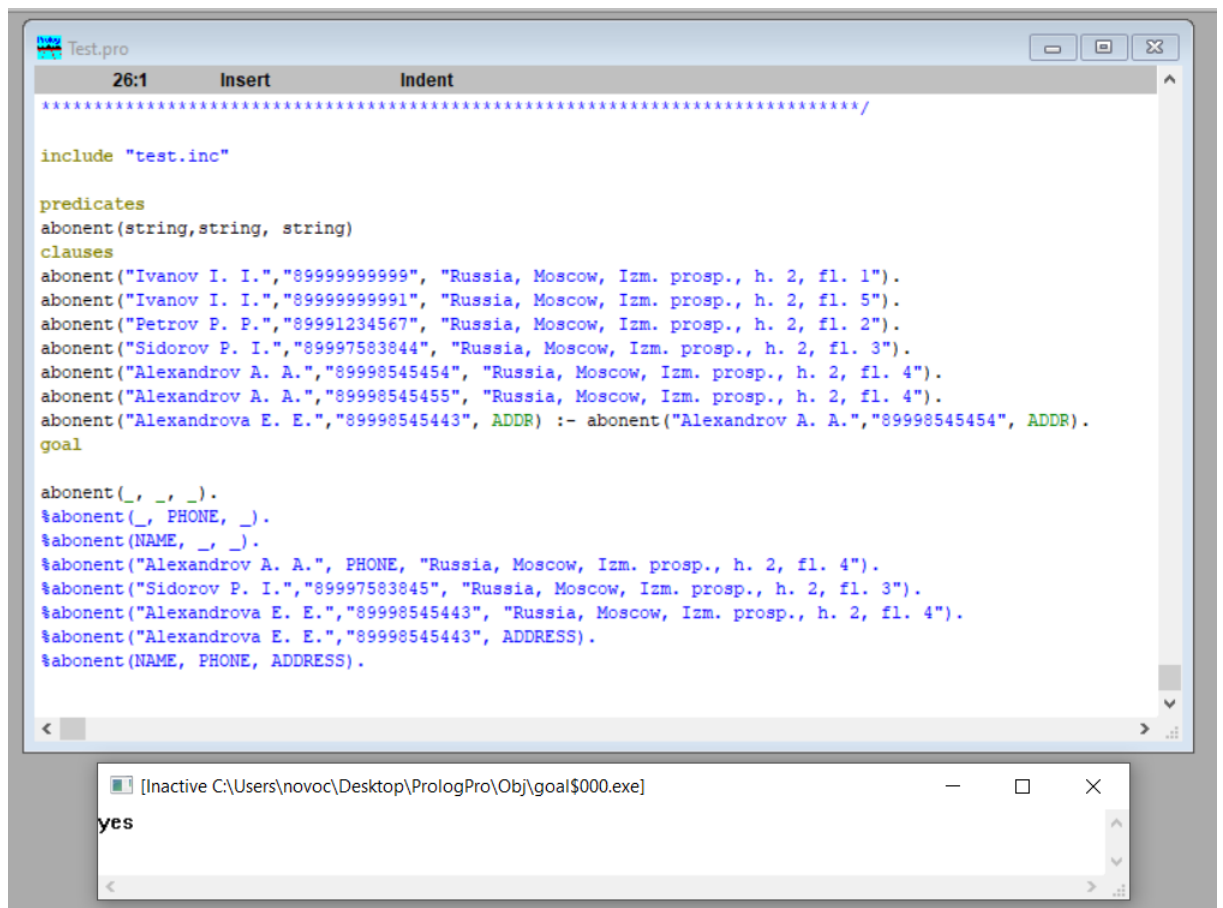


Рисунок 12. Программа «Телефонный справочник» (часть 8).