

Лекция 1.

Определение отношений в программе

РЕКУРСИВНО-ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Понятие логической программы

Логическая программа — это последовательность предложений, описывающих отношения между элементами, или объектами, некоторой задачи.

Термы определяются индуктивно — это константы, переменные или выражения вида $f(t_1, t_2, \dots, t_m)$

где f - функциональный символ, а t_1, t_2, \dots, t_m - термы.

Термы без переменных называются *основными*, или *замкнутыми* термами.

Выражения вида $p(t_1, t_2, \dots, t_n)$

где p — имя предиката, а t_1, t_2, \dots, t_n - термы, называются *атомарными формулами*.

Атомарная формула или ее отрицание называется *литералом*.

Понятие логической программы

Предложение логической программы имеет вид:

$$A_1 \& A_2 \& \dots \& A_k \rightarrow A_0$$

На языке Пролог такое предложение записывается следующим образом: $A_0 : -A_1, A_2, \dots, A_k$

заголовок : —тело

Правило с пустым телом называется *фактом*.

Правило с пустым заголовком — это *цель, запрос* или *вопрос* к программе.

Понятие логическое программы (пример)

млекопитающее("слон").

млекопитающее("зебра").

животное("страус").

животное("уж").

животное(X) :- млекопитающее(X).

Правило выражает импликацию $\forall x(\text{млекопитающее}(x) \rightarrow \text{животное}(x))$

? — животное("зебра")

? — животное(Z)

Понятие логической программы

Переменные:

- локальны
- математический смысл
- значение сохраняется на протяжении одного правила
- не объявляются
- имеет значение регистр только первого символа
- начинаются с прописной буквы или знака подчеркивания
- анонимные переменные не принимают значений

Понятие логической программы

Комментарии:

% комментарий до конца строки

/* комментарий блока произвольной длины */

Декларативная семантика логической программы

Подстановкой термов вместо переменных называется множество θ пар термов вида $x = t$, где x – переменная, а t - терм, не содержащий переменную x .

$A = \text{животное}(X)$

$\theta = \{X = \text{слон}\}$

$A\theta = \text{животное}(\text{слон})$

$B_0 : -B_1, B_2, \dots, B_k$ - вариант исходного правила $A_0 : -A_1, A_2, \dots, A_k$

Декларативная семантика логической программы

Эрбранов универсум логической программы — это множество U термов, построенных индуктивно из констант и функциональных символов, входящих в программу (если в программе не содержится констант, то берется какая-нибудь произвольная константа).

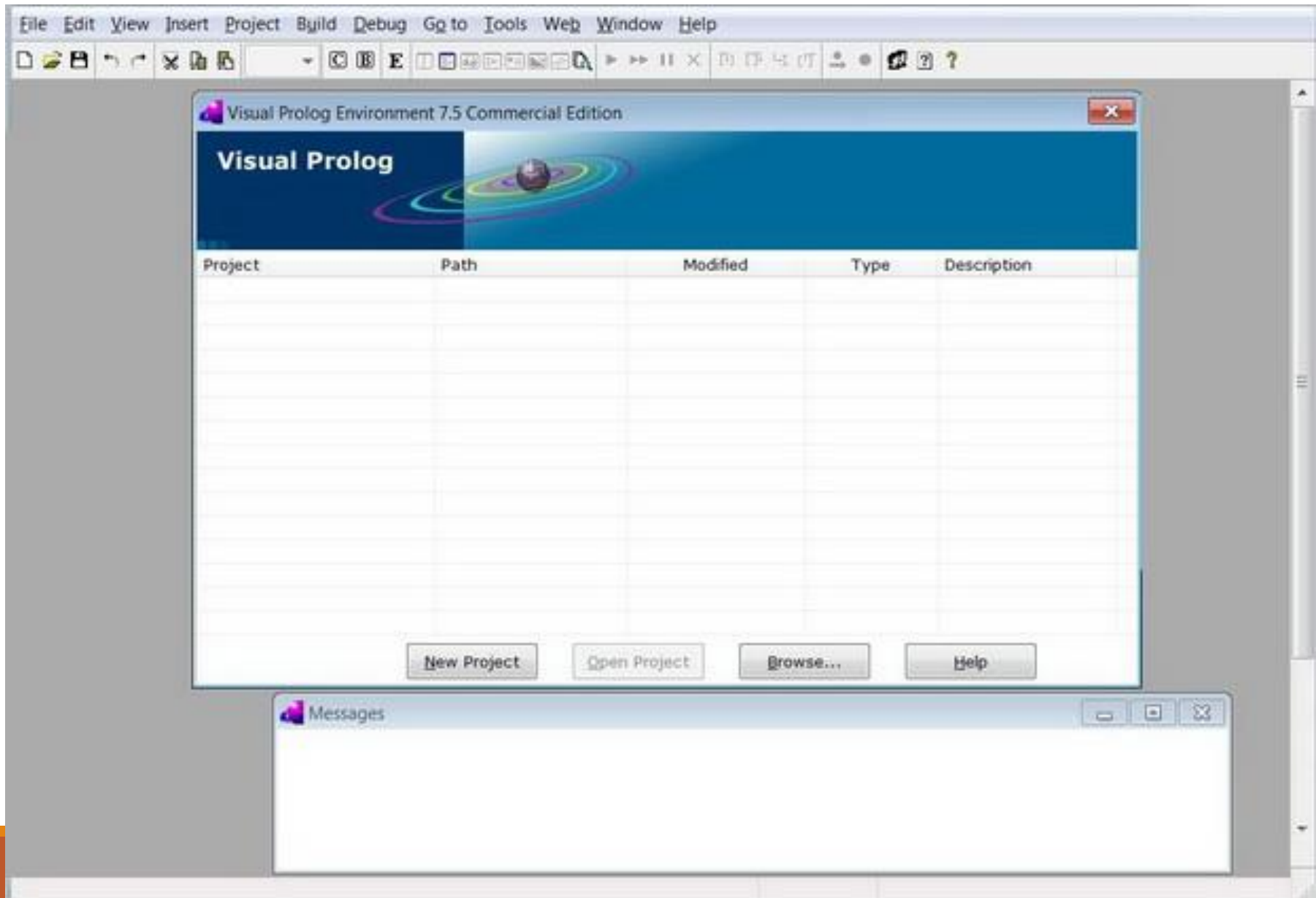
$$U = \{\text{слон, зебра, страус, уж}\}$$

Эрбранов базис логической программы — это множество B простых замкнутых целей, которые можно индуктивно построить с помощью предикатных символов, входящих в программу, и термов из эрбранова универсума.

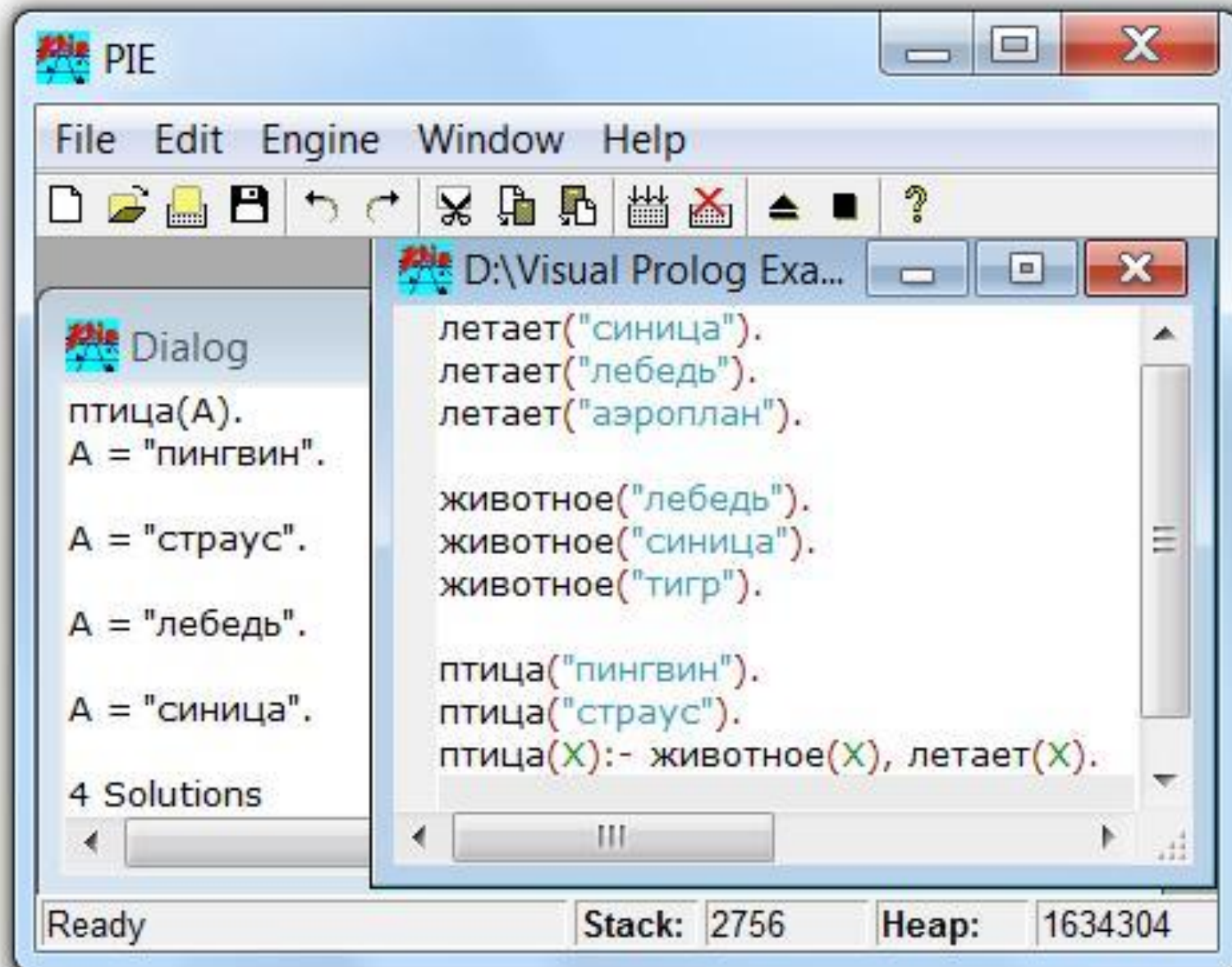
В нашем примере

$B = \{\text{млекопитающее(слон), млекопитающее(зебра), млекопитающее(страус), млекопитающее(уж), животное(слон), животное(зебра), животное(страус), животное(уж)}\}.$

Приложение Prolog Inference Engine



Пример «Птицы»



Пример «Родственные отношения»

родитель("Иван", "Мария").
родитель("Анна", "Мария").
родитель("Мария", "Павел").
родитель("Мария", "Петр").

супруг("Иван", "Анна").
супруг("Павел", "Юлия").

мужчина("Иван").
мужчина("Павел").
мужчина("Петр").

женщина("Мария").
женщина("Анна").
женщина("Юлия").

отец(F, C):-
 родитель(F, C),
 мужчина(F).

мать(M, C):-
 родитель(M, C),
 женщина(M).

Возможные запросы

Частный простой запрос (является ли Иван родителем Петра?):

`родитель("Иван", "Петр").`

Общий простой запрос (найти всех мужчин):

`мужчина(M).`

Конъюнктивный составной общий запрос (найти сыновей Марии):

`родитель("Мария", S), мужчина(S).`

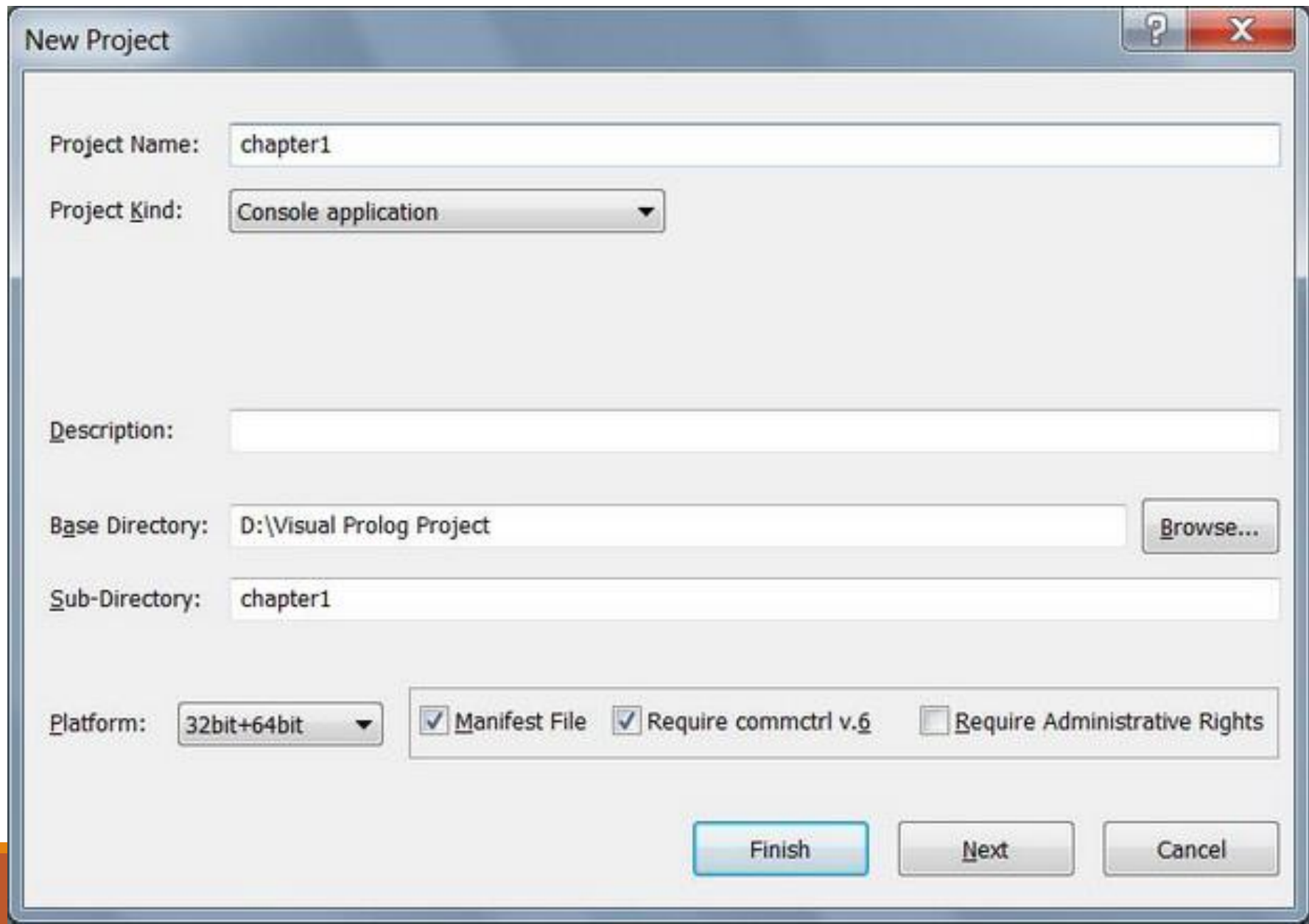
Простой общий запрос с отбором информации при помощи анонимной переменной (кто из женщин замужем?):

`супруг(_, W).`

Дизъюнктивный составной общий запрос (найти всех персон):

`мужчина(P); женщина(P).`

Создание консольных приложений



New Project

Project Name: chapter1

Project Kind: Console application

Description:

Base Directory: D:\Visual Prolog Project Browse...

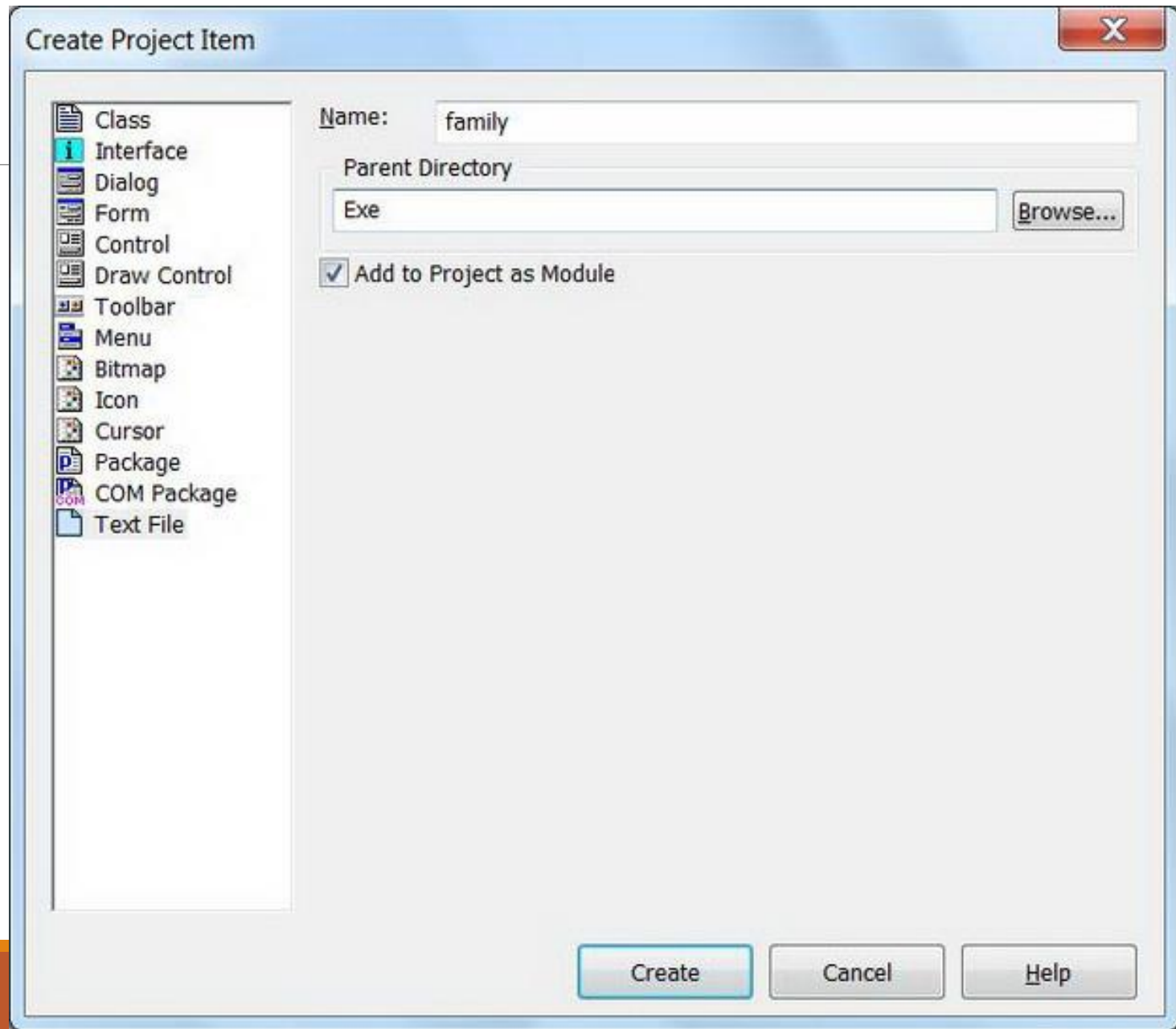
Sub-Directory: chapter1

Platform: 32bit+64bit

☒ Manifest File ☒ Require commctrl v.6 ☐ Require Administrative Rights

Finish Next Cancel

Создание консольных приложений



Файл family.txt. База данных

clauses

```
parent("Иван", "Мария").  
parent("Анна", "Мария").  
parent("Мария", "Павел").  
parent("Мария", "Петр").  
parent("Мария", "Елизавета").
```

```
spouse("Иван", "Анна").  
spouse("Павел", "Юлия").
```

```
male("Иван").  
male("Павел").  
male("Петр").
```

```
female("Мария").  
female("Анна").  
female("Елизавета").  
female("Юлия").
```

main.pro (имплементация класса main)

```
class facts - relatives
    parent: (string Родитель, string Ребенок).
    spouse: (string Муж, string Жена).
    male: (string).
    female: (string).

class predicates
    father: (string Отец, string Ребенок) nondeterm anyflow.
    mother: (string Мать, string Ребенок) nondeterm (o,o).
clauses
    father(X, Y):-
        parent(X, Y),
        male(X).

    mother(X, Y):-
        parent(X, Y),
        female(X).

run():-
    init(),
    file::consult("family.txt", relatives),
    father(X, Y),
        write("отец - ", X, ", ребенок - ", Y), nl,
    fail;
    mother(X, Y),
        write("мать - ", X, ", ребенок - ", Y), nl,
    fail;
    if father("Иван", "Петр") then
        write("\nИван является отцом Петра")
    else
        write("\nИван не является отцом Петра")
    end if,
    _ = readLine().
```


Основные разделы программы

`(class) facts` — объявление предикатов, описывающих факты

`(class) predicates` — объявление предикатов

`domains` — объявление доменов (типов данных)

`constants` — объявление констант

`clauses` — раздел предложений, которые определяют предикаты

`goal` — раздел, в котором определяется цель программы

Основные разделы программы

`class <имя класса> ... end class <имя класса>` — декларация класса;

`class <имя класса > : <имя интерфейса> ... end class <имя класса>` — декларация класса, порождающего объекты;

`interface <имя интерфейса> ... end interface < имя интерфейса >` — интерфейс;

`implement <имя класса > ... end implement <имя класса >` — имплементация класса;

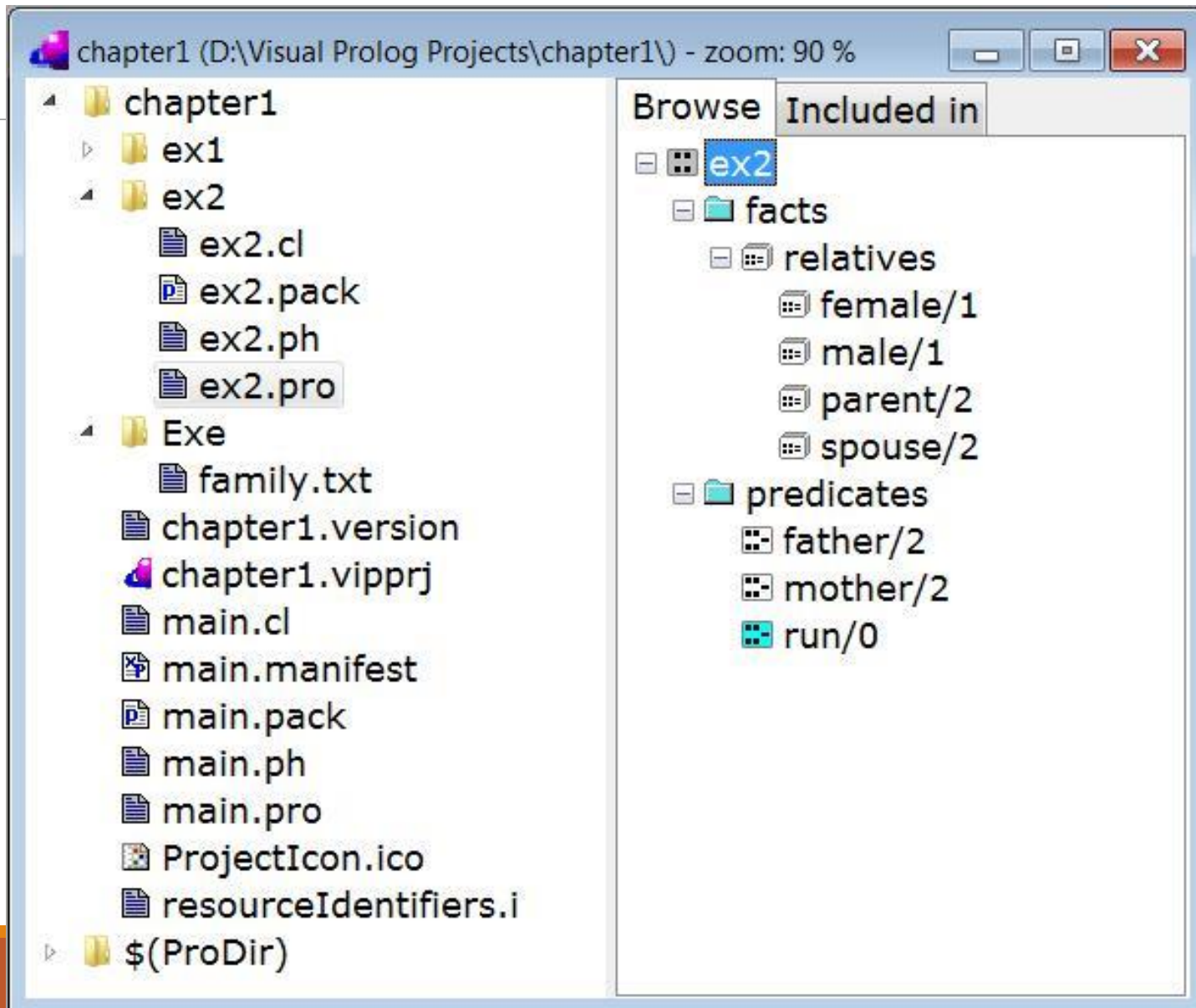
`open` — имена "открытых" классов и интерфейсов;

`properties` — объявление свойств;

`constructors` — объявление конструкторов (в декларациях классов).

Кроме этого, имеются разделы `supports`, `resolve`, `inherits`, `delegates`, `predicates from` и другие.

Создание модулей



Создание модулей

New In New Package -> Create Project Item

(файл ex1.cl)

```
predicates
```

```
    run: ().
```

(файл ex1.pro)

```
open core, console
```

(файл main.pro)

```
goal
```

```
    mainExe::run(main::run),
```

```
    ex1::run().
```

Упражнения к листингу 1.1

1. Какие элементы принадлежат отношению "птица", определенному в программе "Птицы" (см. листинг 1.1)?
2. Запустите приложение PIE, поместите в него программу "Птицы" и найдите ответ на запрос к программе. Для этого откройте новый файл, поместите в него текст программы (факты и правила), сделайте активным окно с программой и выберите команду меню Engine > Reconsult. В окне Dialog введите запрос птица(A).

Упражнения к листингу 1.2

Задайте следующие вопросы программе:

1. Частный простой запрос (является ли Иван родителем Петра?):
родитель("Иван", "Петр").
2. Общий простой запрос (найти всех мужчин):
мужчина(M).
3. Конъюнктивный составной общий запрос (найти сыновей Марии):
родитель("Мария", S), мужчина(S).
4. Простой общий запрос с отбором информации при помощи анонимной переменной (кто из женщин замужем?):
супруг(_, W).
5. Дизъюнктивный составной общий запрос (найти всех персон):
мужчина(P); женщина(P).

Упражнения

1. Определите в программе о родственных отношениях следующие бинарные отношения:

- внук, внучка, дедушка и бабушка;
- ребенок как обратное к отношению родитель;
- сын, дочь, сестра и брат;
- супруги как симметричное замыкание отношения супруг.

2. Напишите программу, которая с помощью бинарных отношений владелец, животное и цвет, хранящих пары хозяин — кличка, кличка — животное и кличка — окрас, соответственно, описывает следующие сведения. Майкл владеет рыжим котом, Сьюзен шоколадным, Дэн и Пит владеют серыми котами. Билл имеет собаку серого окраса, а Бетти шоколадного. Все животные имеют уникальные клички (придумайте их самостоятельно). Составьте запросы к программе:

- найти владельцев серых котов;
- найти животного Билла и цвет этого животного;
- найти животных, которыми владеют Бетти и Майкл;
- найти владельцев животных шоколадного окраса.