

# Алгоритм вычисления алгебраических выражений

**Дано выражение в естественной записи  
со скобками вида:**

$$5+8/6+(45-7)/3$$

**Необходимо вычислить значение  
выражения**

**Первый этап – разбивка на лексемы  
(парсинг)**

## Лексема это:

- Число
- Знак операции { + - \* / }
- Круглые скобки { ( ) }

# Структуры данных:

Используется два стека:

стек операндов (**S1**)

и стек операций (**S2**)

# Присваиваем операциям приоритеты:

- { ( ) } – приоритет 0;
- { + - } – приоритет 1;
- { \* / } – приоритет 2;
- { ^ } - приоритет 3.

**Второй этап - вычисление**

1. Список лексем исчерпан? Да – на п.7;
2. Берем очередную лексему;
3. Это **число** – кладем его в стек S1. На п.1;
4. Это **открывающая скобка** – кладем её в стек S2. На п.1;
5. Это **операция** – сравниваем её приоритет с приоритетом вершины стека S2.
  - Если приоритет операции **ВЫШЕ** – кладём операцию в стек S2. На п.1;
  - Иначе извлекаем из стека S1 ДВА элемента, а из стека S2 – операцию, **ВЫПОЛНЯЕМ** операцию и результат кладем в стек S1. **Повторяем, пока в стеке операций есть операции с приоритетом, равным приоритету пришедшей операции.** Пришедшую операцию – в стек!
6. Это **закрывающая скобка** – извлекаем из стека S2 операцию, а из стека S1 пары чисел, **ВЫПОЛНЯЕМ** операцию, результат кладем в стек S1. Повторяем п.6 до появления открывающей скобки. Удаляем ее. На п.1;
7. Извлекаем из стека S2 операцию, а из стека S1 пары чисел, **ВЫПОЛНЯЕМ** операцию, результат кладем в стек S1. Повторяем п.7 до исчерпания стека S2. Результат – в S1



$$1+5*(6-2)/2$$

1  1

2  +

3  5

4  \*

5  (

6  6

7  -

8  2

9  )

10  /

11  2

$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ **{ }**

**S2** □ **{ }**

**Вых. Сост:**

**S1** □ **{ 1 }**

**S2** □ **{ }**

$$1+5^*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 }

**S2** □ { }

**Вых. Сост:**

**S1** □ { 1 }

**S2** □ { + }

$$1+5^*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 }

**S2** □ { + }

**Вых. Сост:**

**S1** □ { 1 5 }

**S2** □ { + }

$$1+5^*(6-2)/2$$

**Вх. Сост:**

S1 □ { 1 5 }

S2 □ { + }

**Prty(\*) > Prty(+)**

**Вых. Сост:**

S1 □ { 1 5 }

S2 □ { + \* }

$$1+5^*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 5 }

**S2** □ { + \* }

**Вых. Сост:**

**S1** □ { 1 5 }

**S2** □ { + \* ( }

$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 5 }

**S2** □ { + \* ( }

**Вых. Сост:**

**S1** □ { 1 5 6 }

**S2** □ { + \* ( }

$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 5 6 }

**S2** □ { + \* ( }

**Prty(-) > Ptry((**

**Вых. Сост:**

**S1** □ { 1 5 6 }

**S2** □ { + \* ( - }



$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 5 6 }

**S2** □ { + \* ( - }

**ВЫХ. Сост:**

**S1** □ { 1 5 6 2 }

**S2** □ { + \* ( - }

$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 5 6 2 }

**S2** □ { + \* ( - }

**A2=2**

**A1=6**

**R=(A1-A2)=4**

**Вых. Сост:**

**S1** □ { 1 5 4 }

**S2** □ { + \* }

$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 5 4 }

**S2** □ { + \* }

**Prty(/)=Prty(\*)**

**A2=4**

**A1=5**

**R=A1\*A2=20**

**Вых. Сост:**

**S1** □ { 1 20 }

**S2** □ { + / }

$$1+5*(6-2)/2$$

**Вх. Сост:**

**S1** □ { 1 20 }

**S2** □ { + / }

**Вых. Сост:**

**S1** □ { 1 20 **2** }

**S2** □ { + / }

# Список лексем исчерпан. Опустошаем стек S2

Вх. Сост:

S1 □ { 1 20 2 }

S2 □ { + / }

A2=2

A1=20

R=A1/A2=10

Вых. Сост:

S1 □ { 1 10 }

S2 □ { + }

# Продолжаем опустошать стек S2

**Вх. Сост:**

**S1** □ { 1 10 }

**S2** □ { + }

**A2=10**

**A1=1**

**R=A1+A2=11**

**Вых. Сост:**

**S1** □ { **11** }

**S2** □ { }

**Результат = 11**

**УРА!!!**

**Еще один пример:**

$$5+8-6*3$$

**Необходимо вычислить значение  
выражения**



$$5+8-6*3$$

1  5

2  +

3  8

4  -

5  6

6  \*

7  3

$$5+8-6*3$$

**Вх. Сост:**

**S1** □ **{ }**

**S2** □ **{ }**

**Вых. Сост:**

**S1** □ **{ 5 }**

**S2** □ **{ }**

$$5+8-6*3$$

**Вх. Сост:**

**S1** □ { 5 }

**S2** □ { }

**Вых. Сост:**

**S1** □ { 5 }

**S2** □ { + }

$$5+8-6*3$$

**Вх. Сост:**

**S1** □ { **5** }

**S2** □ { **+** }

**Вых. Сост:**

**S1** □ { **5 8** }

**S2** □ { **+** }

$$5+8-6*3$$

**Вх. Сост:**

**S1** □ { **5 8** }

**S2** □ { **+** }

**Prty(+)=Prty(-)**

**A2=5**

**A1=8**

**R=A1+A2=13**

**Вых. Сост:**

**S1** □ { **13** }

**S2** □ { **-** }

$$5+8-6^*3$$

**Вх. Сост:**

**S1** □ { 13 }

**S2** □ { - }

**Вых. Сост:**

**S1** □ { 13 6 }

**S2** □ { - }

$$5+8-6^*3$$

**Вх. Сост:**

**S1** □ { 13 6 }

**S2** □ { - }

**Prty(\*) > Prty(-)**

**Вых. Сост:**

**S1** □ { 13 6 }

**S2** □ { - \* }

$$5+8-6^*3$$

**Вх. Сост:**

**S1** □ { 13 6 }

**S2** □ { - \* }

**Вых. Сост:**

**S1** □ { 13 6 3 }

**S2** □ { - \* }



Список лексем исчерпан  
Опустошаем стек S2

Вх. Сост:

S1 □ { 13 6 3 }

S2 □ { - \* }

A2=3

A1=6

R=A1\*A2=18

Вых. Сост:

S1 □ { 13 18 }

S2 □ { - }

## Продолжаем опустошать стек S2

**Вх. Сост:**

**S1** □ { 13 18 }

**S2** □ { - }

**A2=18**

**A1=13**

**R=A1-A2=-5**

**Вых. Сост:**

**S1** □ { -5 }

**S2** □ { }

**Результат = -5**

**УРА!!!**

# Применение деревьев поиска

**“Угадай животное”** – программа,  
способная обучаться

# Постановка задачи:

Программа должна задавать человеку вопросы и “отгадать” загаданное человеком животное. Вопросы должны носить “двоичный характер” (да-нет).

Если программа не в состоянии отгадать животное, она “сдаётся”, спрашивает человека, как **называется** животное и **чем отличается** от названного...

**Эта информация запоминается в “базе знаний”  
(БЗ) и может быть использована при  
следующих сеансах игры.**

# Проектируем...

**Очевидно, что “сердцем” программы является хранилище данных.**

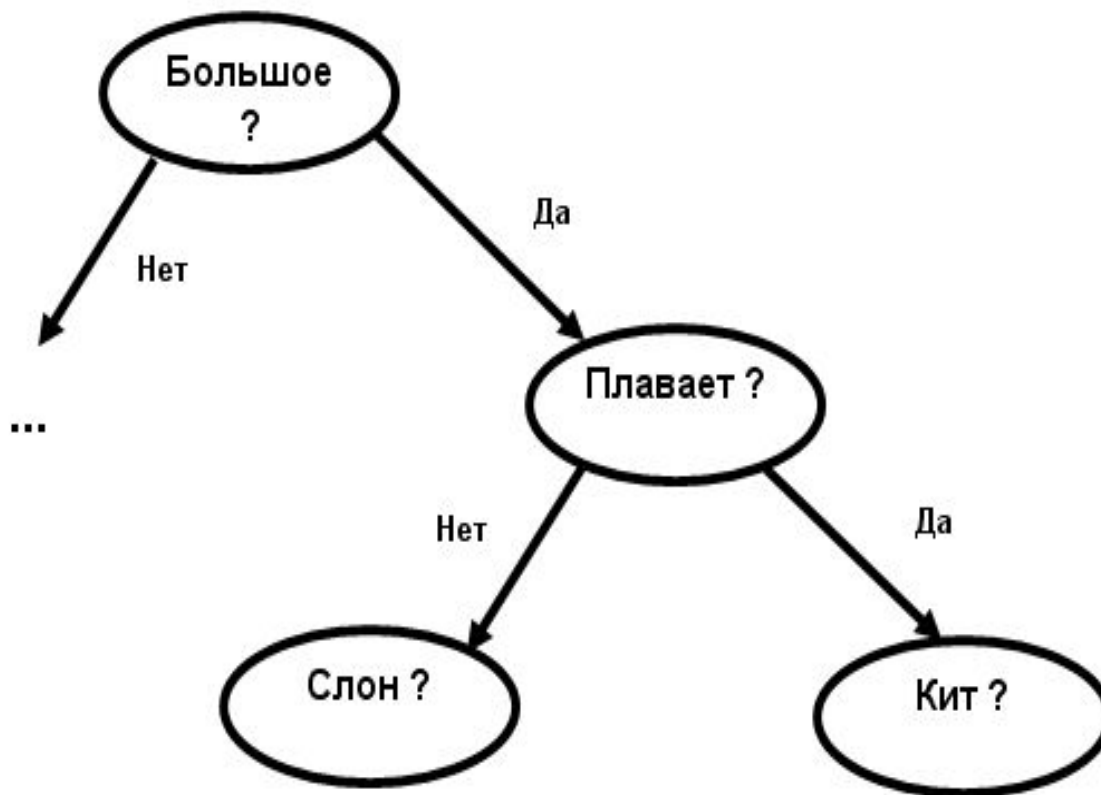
**Какую структуру данных выбрать?**

# Выбираем двоичное дерево!

Узел дерева будет хранить текст вопроса, а правая и левая ссылки будут указывать на поведение программы при ответе “**Да**” (правая) и “**Нет**” (левая).

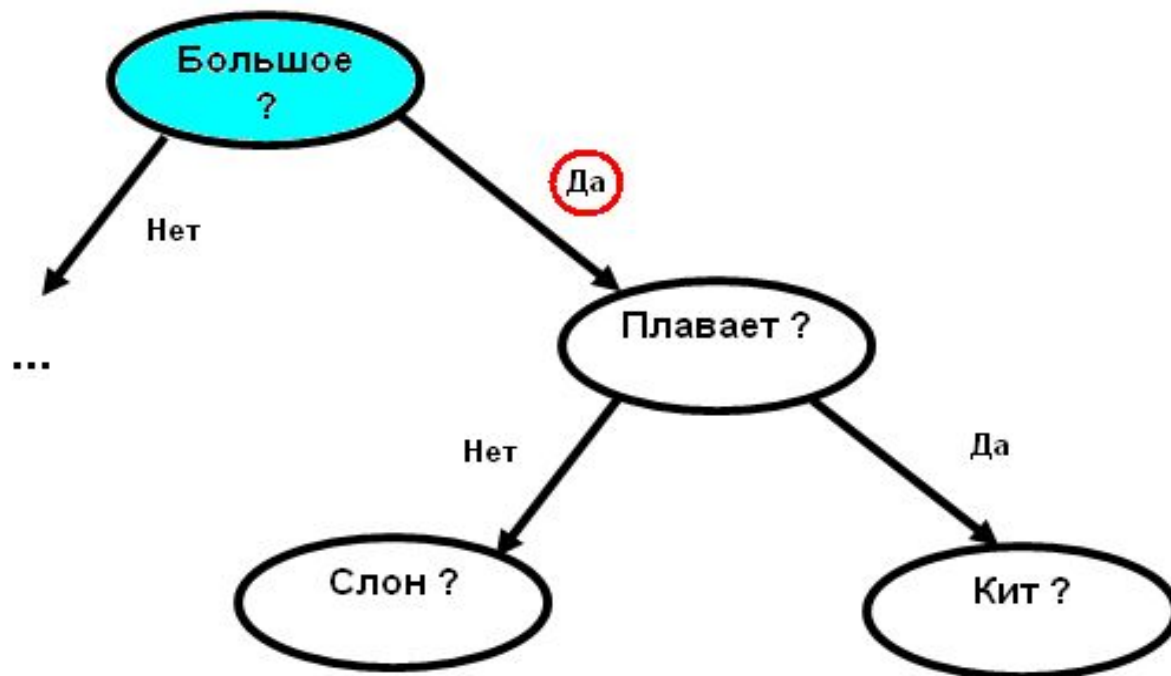


# Исходное состояние БЗ

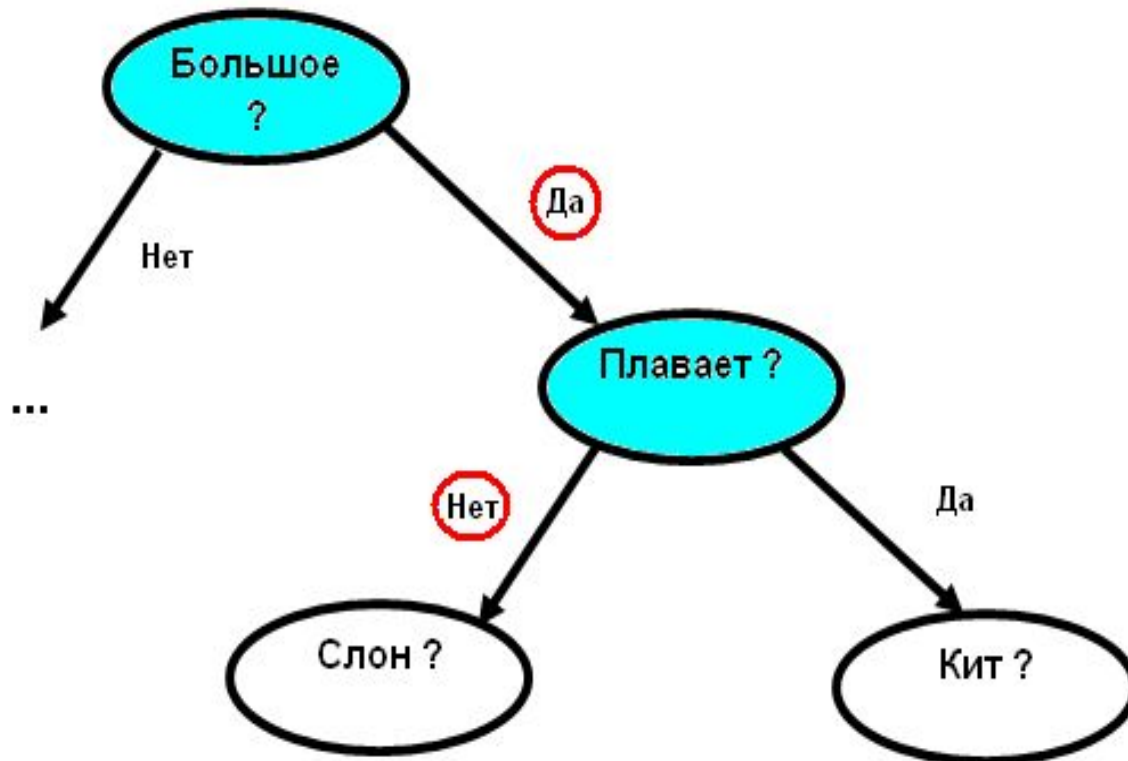


**Загадываем слона.**

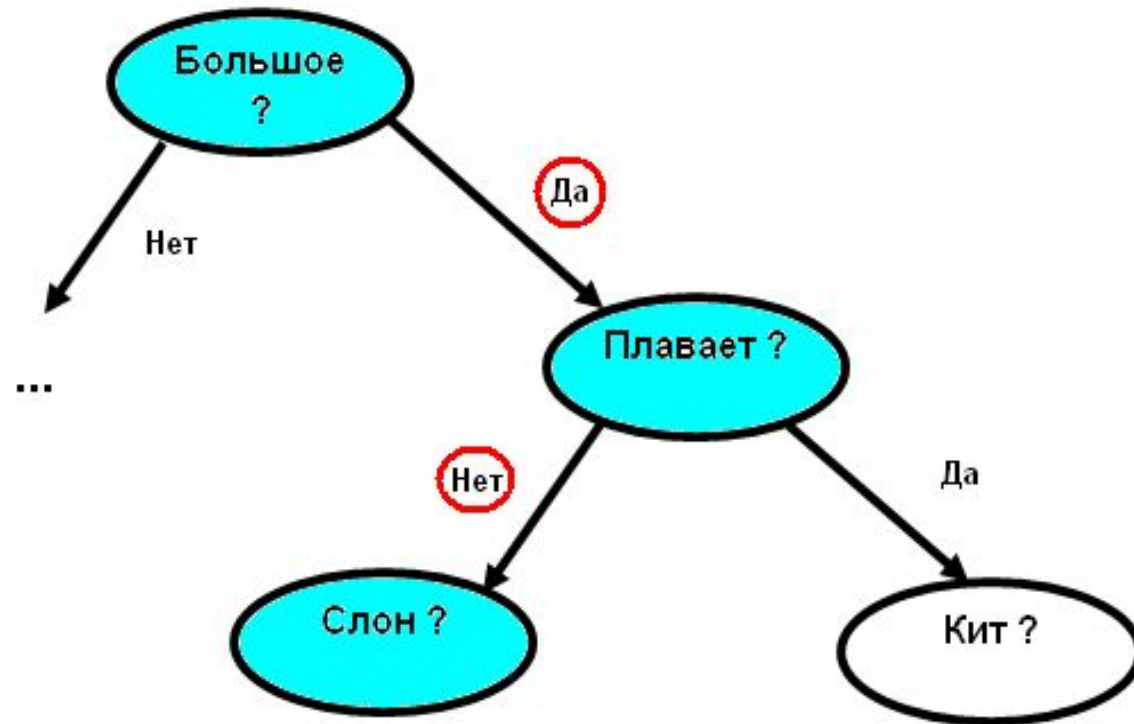
# 1-й вопрос программы



# 2-й вопрос программы



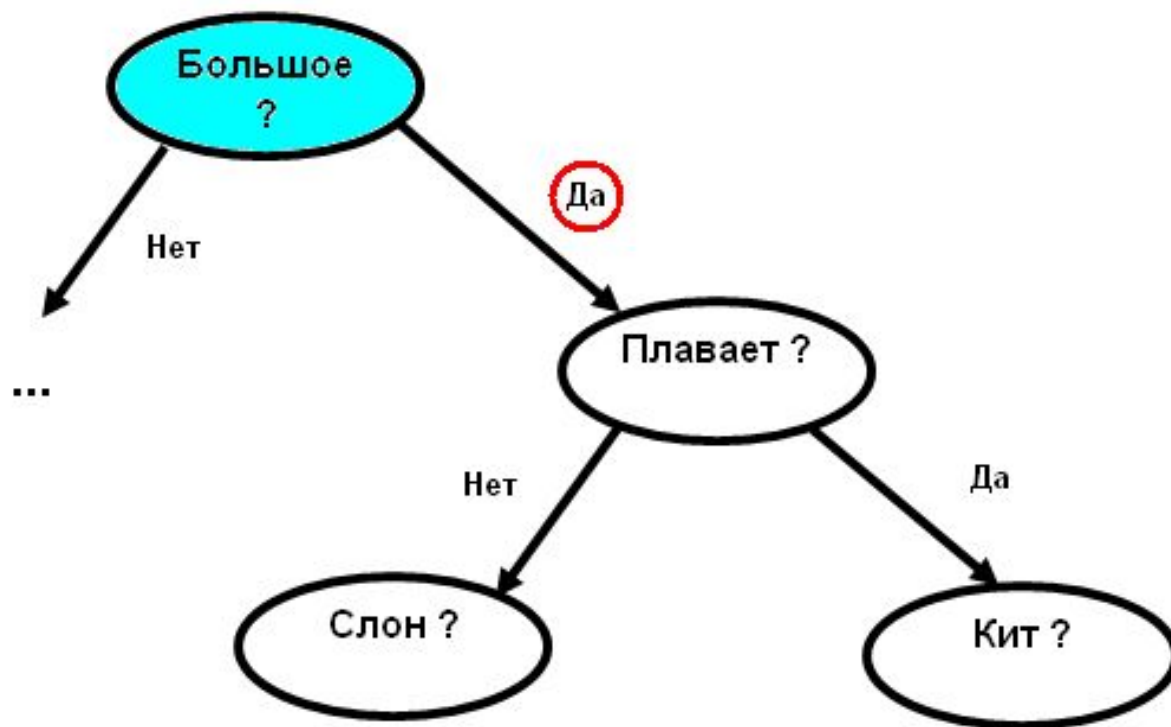
# 3-й вопрос программы



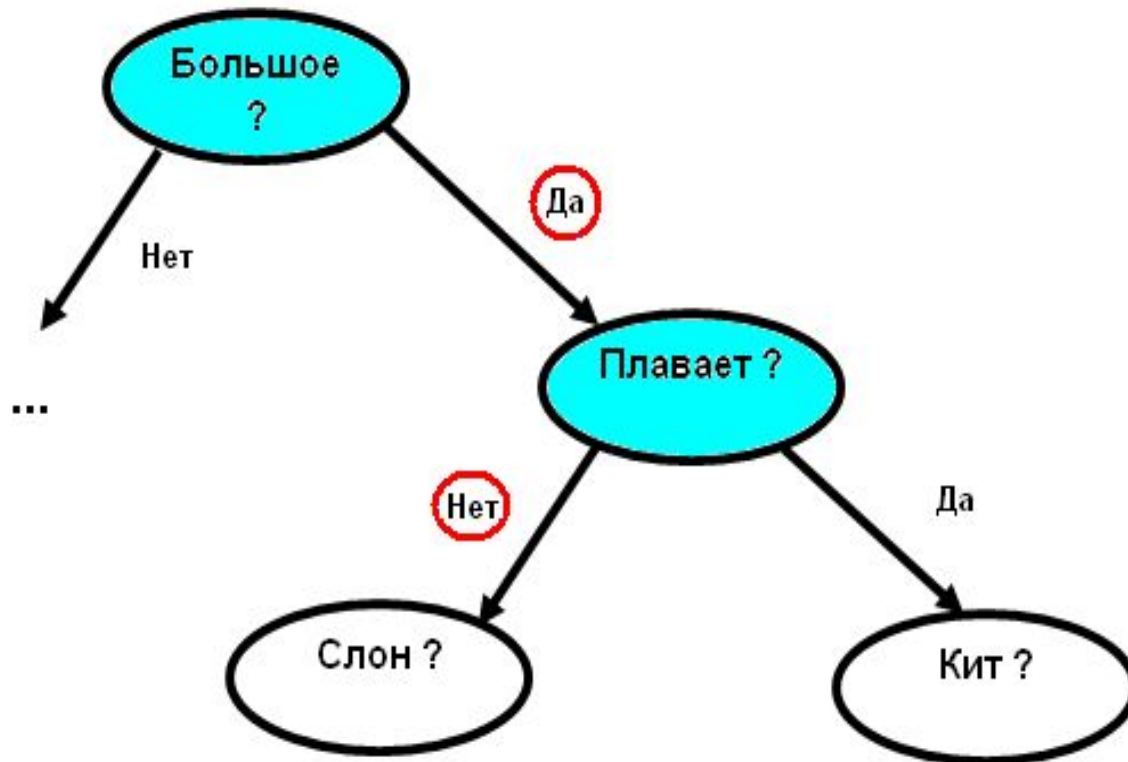
**Ответ “Да” – и Слон вычислен!**

**А теперь загадываем мамонта.**  
**Мамонта в базе знаний нет.**  
**Посмотрим на поведение**  
**программы...**

# 1-й вопрос программы

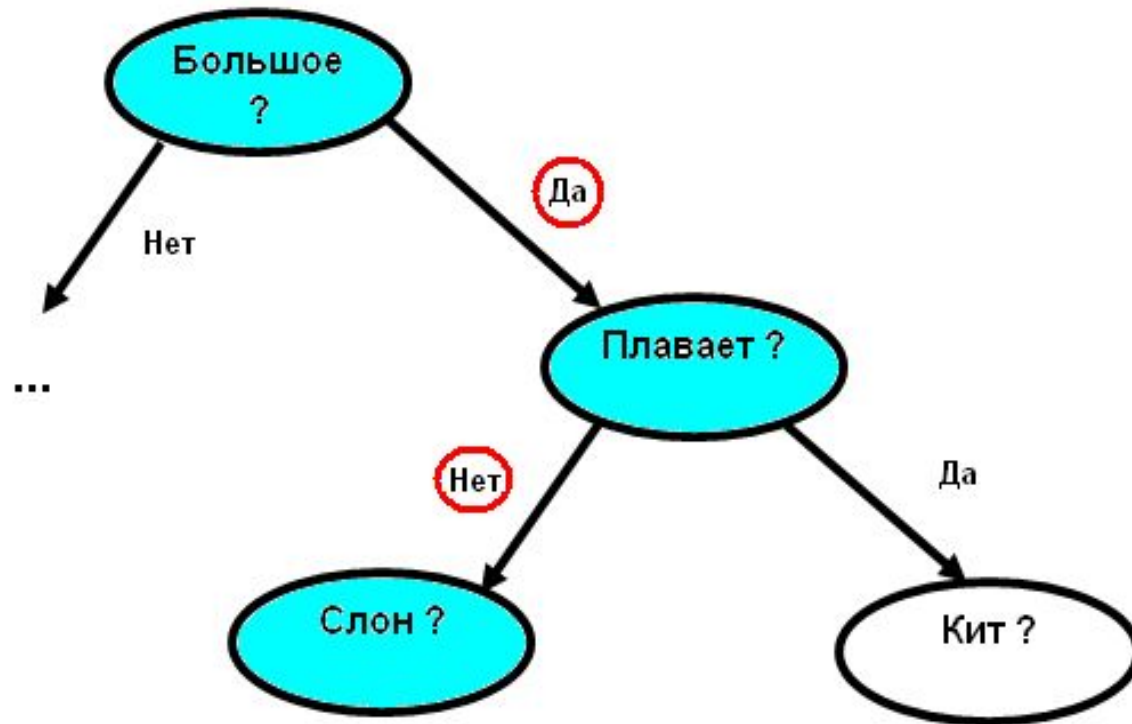


# 2-й вопрос программы





# 3-й вопрос программы



**Ответ “нет” и у слона нет потомков в дереве поиска...**

**Программа признаёт проигрыш и спрашивает, как называется загаданное животное.**

**Человек отвечает: “Мамонт”**

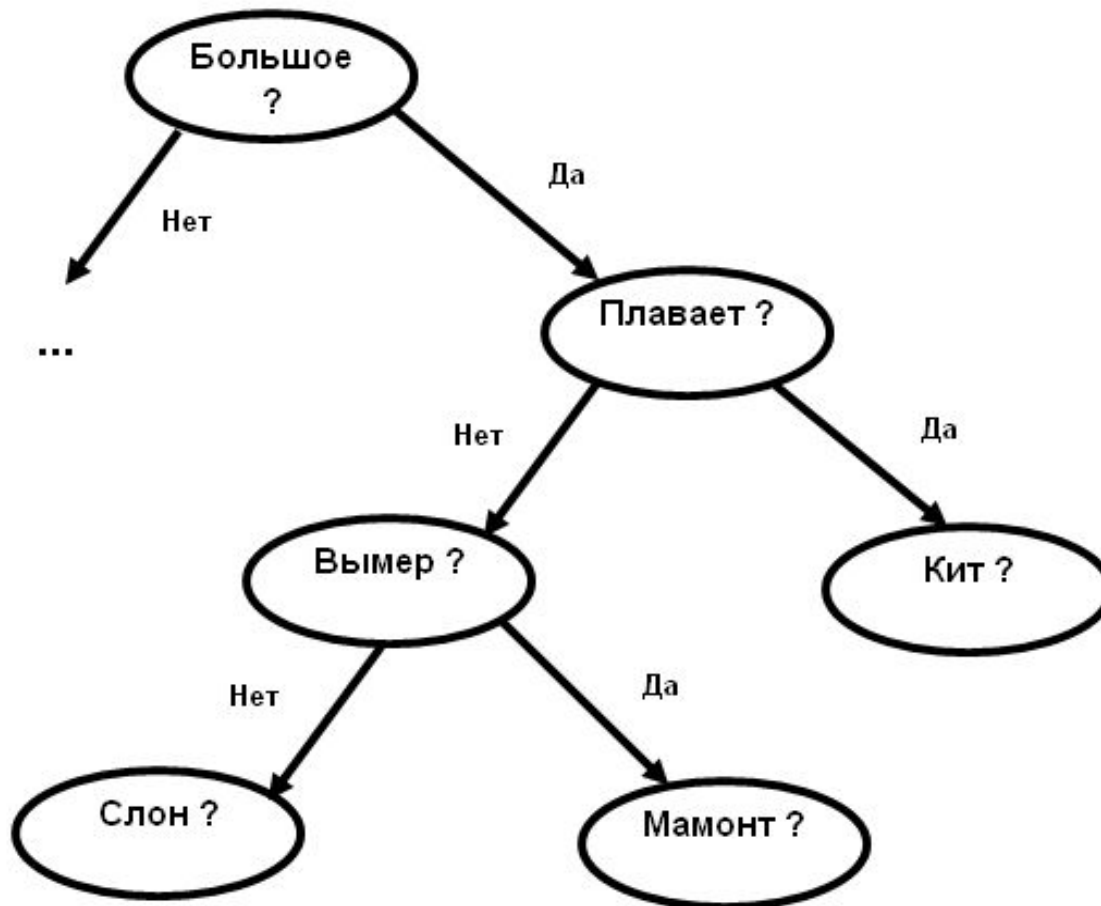
**Поскольку программа остановилась на Слоне, то нужно спросить человека, чем Мамонт отличается от Слона.**

**Человек отвечает “Животное вымерло”**

# Что должна сделать программа?

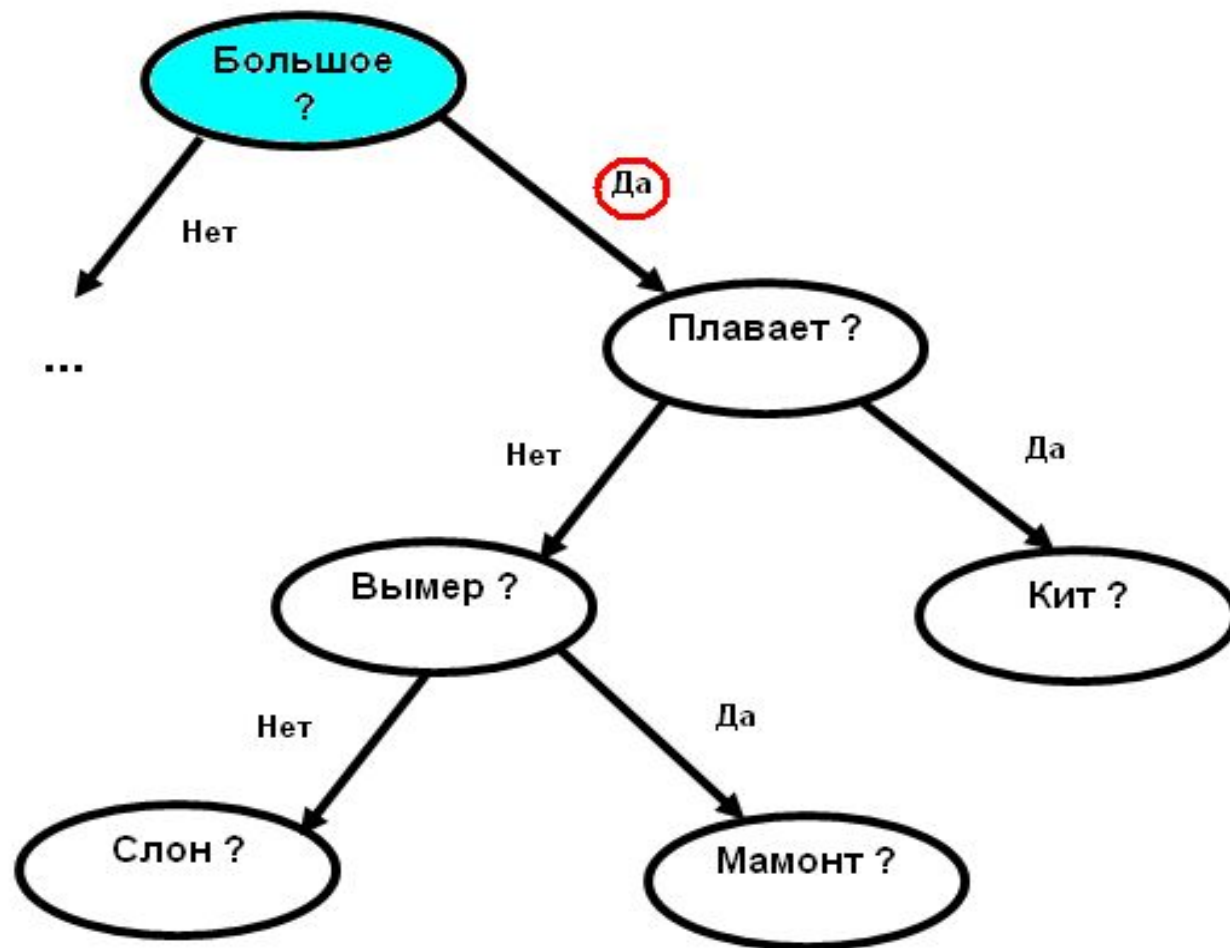
- 1) Создать новый узел в дереве поиска и занести в него...  
вопрос **“Животное вымерло?”**
- 2) **“Подвесить”** этот узел к предку Слона
- 3) У этого нового узла сделать левым потомком (почему?) узел **“Слон?”**, а правым – еще один новый узел (в него записать **“Мамонт?”**)

# Новое состояние БЗ

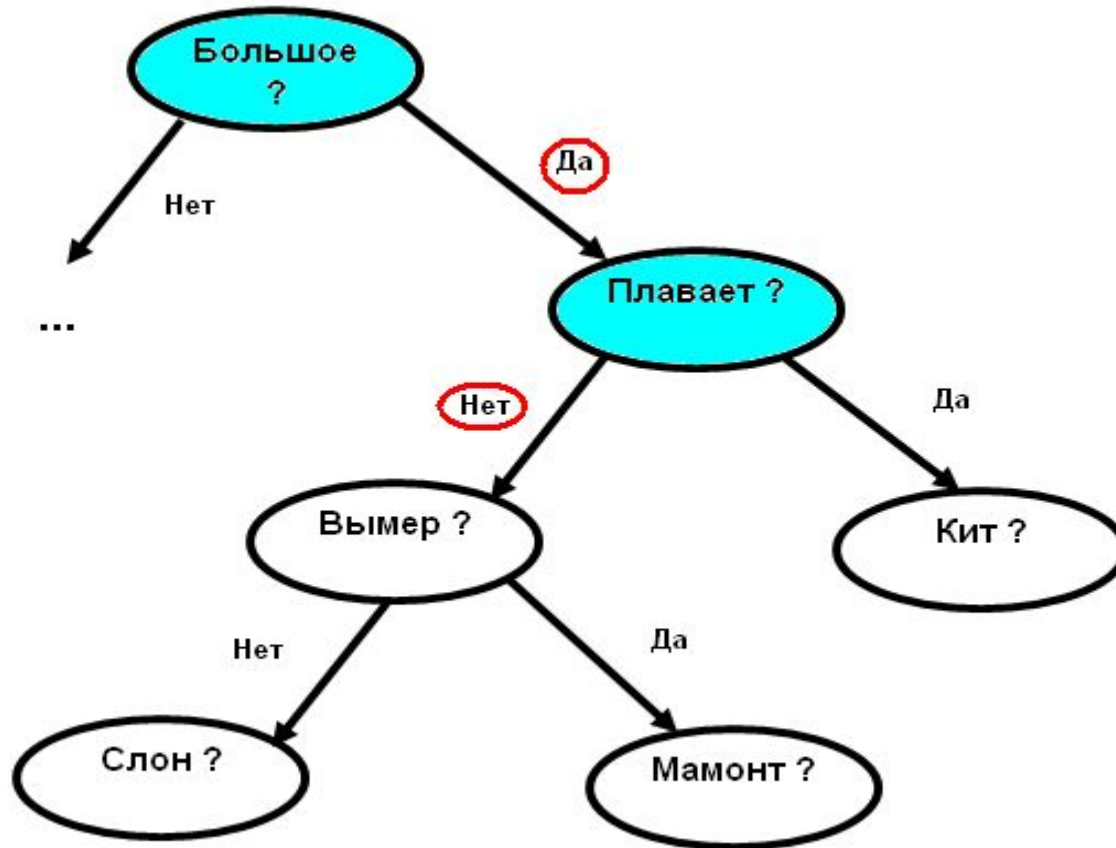


**Еще раз загадаем мамонта и пройдем  
по дереву поиска...**

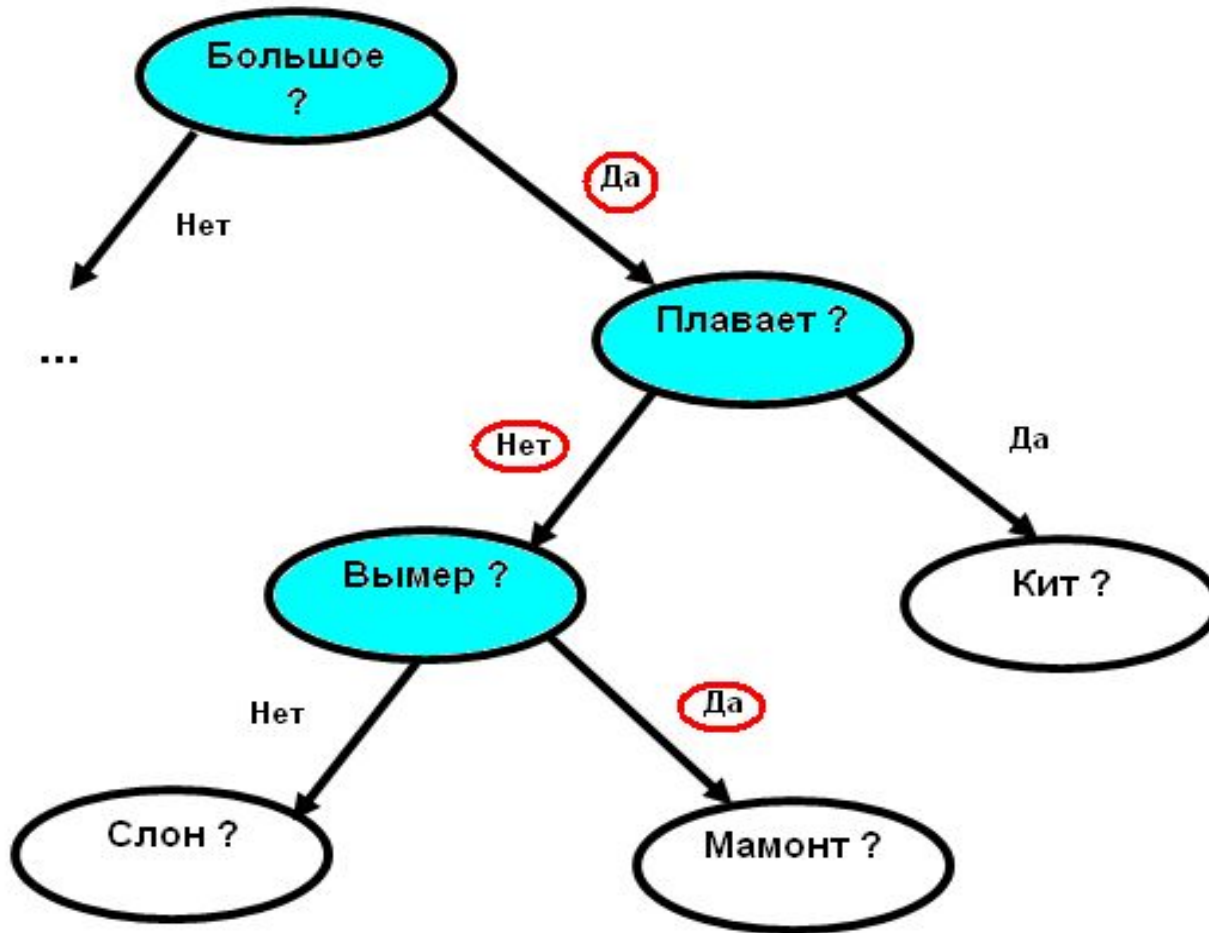
# Первый вопрос:



# Второй вопрос:

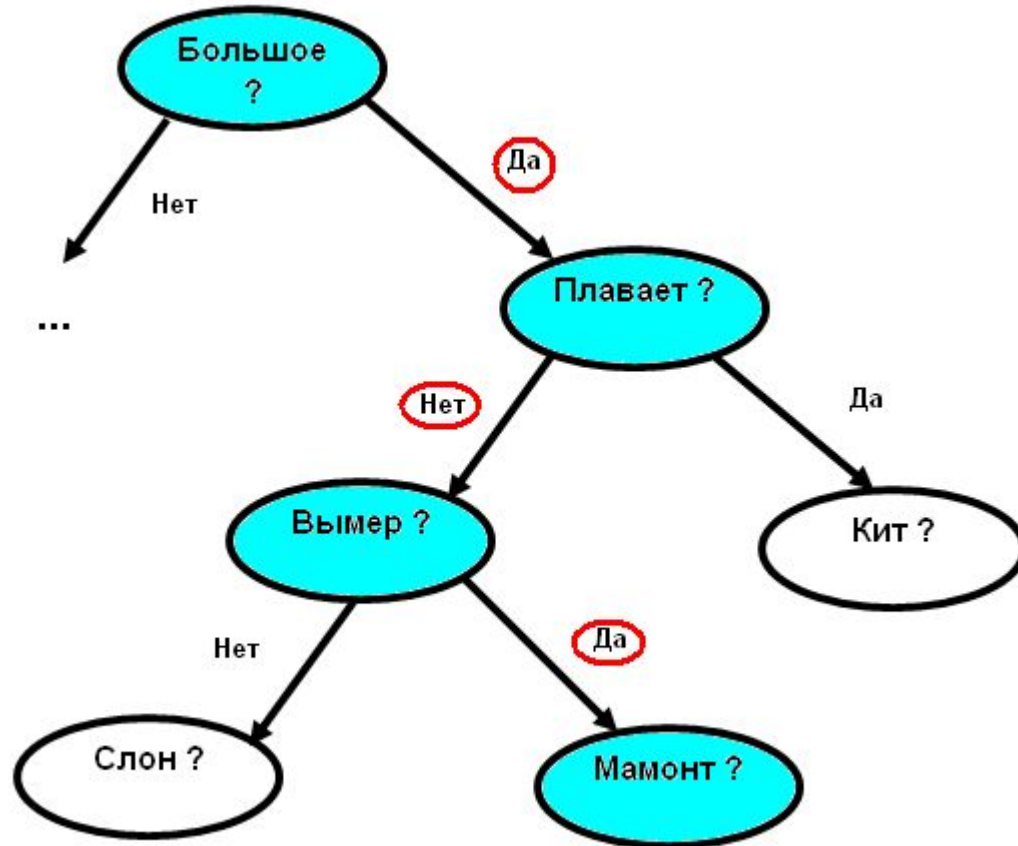


# Третий вопрос (уже новый!)





# Последний вопрос



**Мамонт успешно угадан...**

**Вот реальная программа, работающая  
по описанному принципу:**

**<http://ru.akinator.com/>**