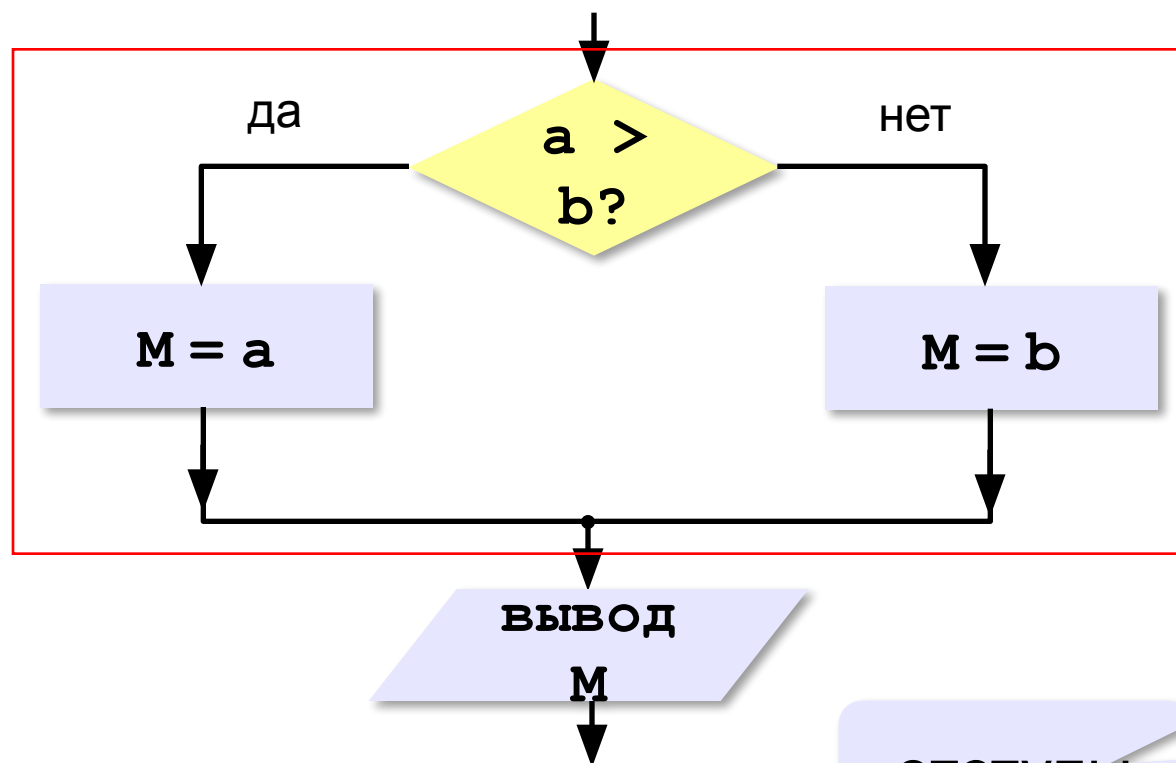


Программирование (Python)

Ветвления

Выбор наибольшего из двух чисел

Задача: **изменить порядок действий** в зависимости от выполнения некоторого условия.



полная
форма
ветвления

? Если $a = b$?

```
if a > b:  
    M = a  
else:  
    M = b
```

отступы

Вариант 1. Программа

```
print("Введите два целых числа")  
a = int(input())  
b = int(input())
```

```
if a > b:
```

```
    M = a
```

```
else:
```

```
    M = b
```

полная форма
условного
оператора

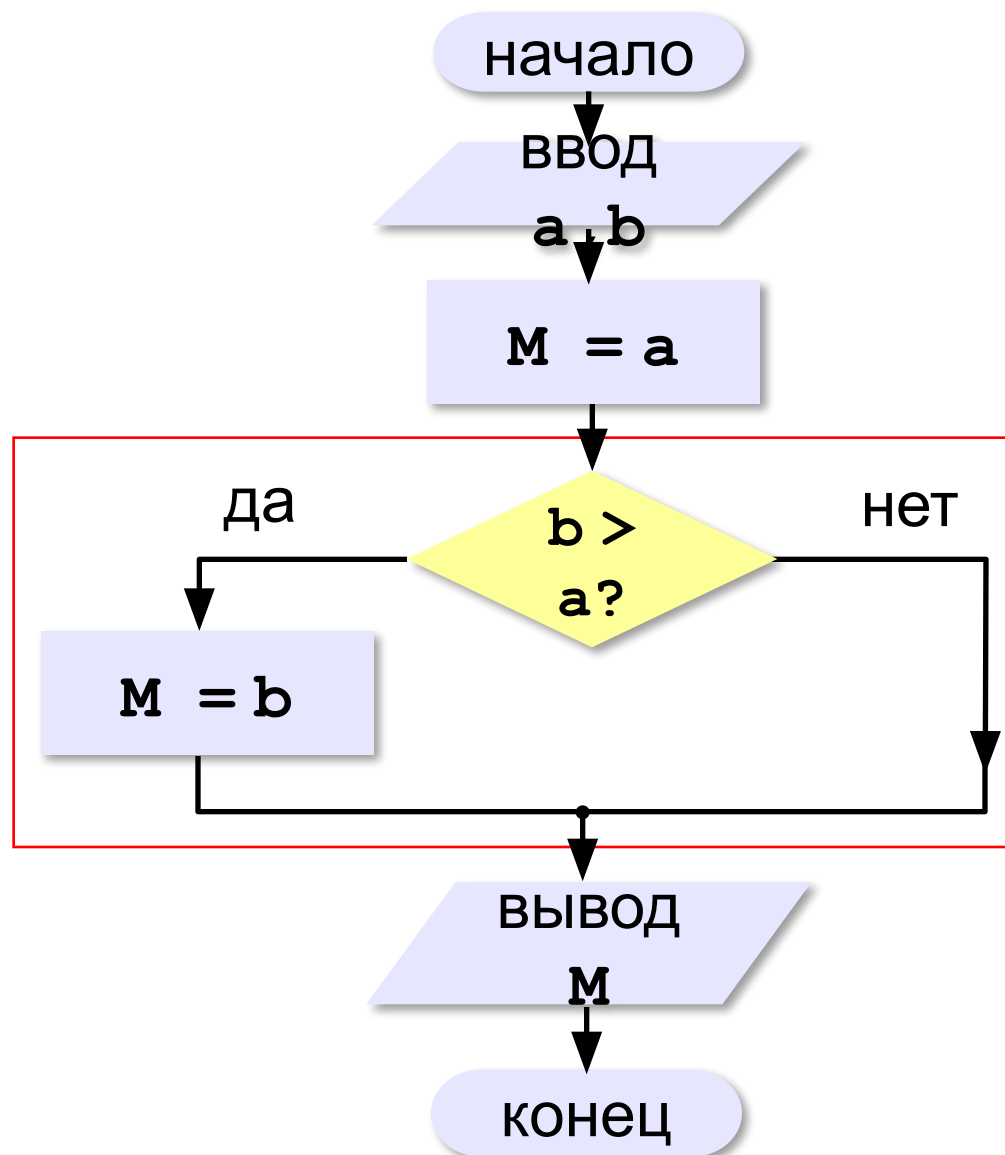
```
print("Наибольшее число", M)
```

Решение в стиле Python:

```
M = max(a, b)
```

```
M = a if a > b else b
```

Выбор наибольшего из двух чисел-2



неполная
форма
ветвления

Вариант 2. Программа

```
print("Введите два целых числа")
```

```
a = int(input())
```

```
b = int(input())
```

```
M = a
```

```
if b > a:
```

```
    M = b
```

неполная форма
условного
оператора

```
print("Наибольшее число", M)
```

Примеры

Поиск минимального:

```
if a < b:  
    M = a  
if b < a:  
    M = b
```



Что плохо?



Когда работает неверно?

Примеры

```
if a < b:
```

```
    c = a
```

```
    a = b
```

```
    b = c
```



Что делает эта программа?



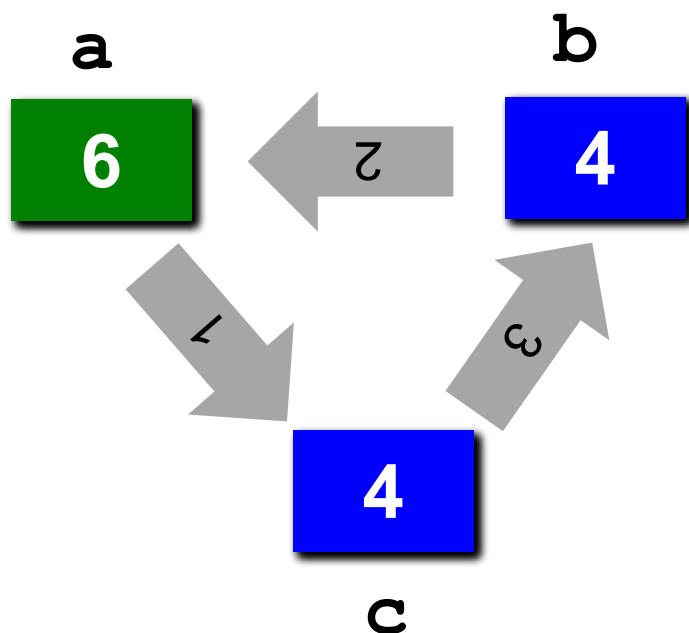
В чём отличие?

```
if a < b:
```

```
    c = a
```

```
    a = b
```

```
    b = c
```



Решение в стиле Python:

```
a, b = b, a
```

В других языках программирования

Паскаль:

```
if a < b then begin
    c := a;
    a := b;
    b := c;
end;
```

C:

```
if (a < b) {
    c = a;
    a = b;
    b = c;
}
```


Знаки отношений

> **<** больше, меньше

>= больше или равно

<= меньше или равно

== равно

!= не равно

Вложенные условные операторы

Задача: в переменных **a** и **b** записаны возрасты Андрея и Бориса. Кто из них старше?



Сколько вариантов?

```
if a > b:  
    print("Андрей старше")  
else:  
    if a == b:  
        print("Одного возраста")  
    else:  
        print("Борис старше")
```



Зачем нужен?

вложенный
условный оператор

Каскадное ветвление

```
if a > b:  
    print("Андрей старше")  
elif a == b:  
    print("Одного возраста")  
else:  
    print("Борис старше")
```



elif = else if

Каскадное ветвление

```
cost = 1500
if cost < 1000:
    print ( "Скидок нет." )
elif cost < 2000:
    print ( "Скидка 2%." )
elif cost < 5000:
    print ( "Скидка 5%." )
else:
    print ( "Скидка 10%." )
```

первое сработавшее
условие



Что выведет?

Скидка 2%.

Сложные условия

Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ "подходит" или "не подходит").

Особенность: надо проверить, выполняются ли два условия одновременно:

возраст \geq 25

возраст \leq 40



Можно ли решить известными методами?

Плохое решение

```
print("Введите ваш возраст")
```

```
v = int(input())
```

```
if v >= 25:
```

```
    if v <= 40:
```

```
        print("Подходит!")
```

```
    else:
```

```
        print("Не подходит.")
```

```
else:
```

```
    print("Не подходит.")
```

вложенный
условный
оператор

Хорошее решение (операция «И»)

Задача: набор сотрудников в возрасте **25-40 лет**
(включительно).

сложное условие

```
if v >= 25 and v <= 40 :  
    print ("подходит")  
else:  
    print ("не подходит")
```

and «И»: одновременное выполнение
всех условий!

Примеры

Задача. Вывести "Да", если число в переменной `a` – двузначное.

```
if 10 <= a and a <= 99:  
    print("Да")
```

Задача. Вывести "Да", если число в переменной `a` – двузначное и делится на 7.

```
if 10 <= a and a <= 99 and (a % 7) == 0:  
    print("Да")
```


Сложные условия: «ИЛИ»

Задача. Самолёт летает по понедельникам и четвергам.
Ввести номер дня недели и определить, летает ли в этот день самолёт.

Особенность: надо проверить, выполняется ли **одно из двух** условий:

день = 1

день = 4

```
if d == 1 or d == 4 :  
    print ("Летает")  
else:  
    print ("Не летает")
```

сложное
условие

or «ИЛИ»: выполнение **хотя бы одного**
из двух условий!

Ещё пример

Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ "подходит" или "не подходит"). Использовать «ИЛИ».

```
if v < 25 or v > 40 :  
    print ("не подходит")  
else:  
    print ("подходит")
```

Сложные условия: «НЕ»

```
if not (a < b):  
    print("Старт!")
```



Как без «НЕ»?

not «НЕ»: если выполняется обратное условие

```
if a >= b:  
    print("Старт!")
```

Простые и сложные условия

Простые условия (отношения)

равно

< <= > >= == !=

не равно

Сложное условие – это условие, состоящее из нескольких простых условий (отношений), связанных с помощью логических операций:

- **and** – одновременное выполнение условий

$$v \geq 25 \text{ and } v \leq 40$$

- **or** – выполнение хотя бы одного из условий

$$v \leq 25 \text{ or } v \geq 40$$

- **not** – отрицание, обратное условие

$$\text{not } (v > 25) \quad \Leftrightarrow$$

$$v \leq 25$$

Порядок выполнения операций

- выражения в скобках
- <, <=, >, >=, =, !=
- not
- and
- or

```
      4      1      6      2      5      3  
if not a > 2 or c != 5 and b < a:  
    ...
```

Сложные условия

Истинно или ложно

a	2
b	3
c	4

`not (a > b)`

Да

`a < b and b < c`

Да

`a > c or b > c`

Нет

`a < b and b > c`

Нет

`a > c and b > d`

Нет

`not (a >= b) or c == d`

Да

`a >= b or not (c < b)`

Да

`a > c or b > c or b > a`

Да

Логические переменные

```
b = True  
b = False  
type(b)
```

ТОЛЬКО ДВА
ВОЗМОЖНЫХ
ЗНАЧЕНИЯ

```
<class 'bool'>
```

логическая (булевская)
переменная



Джордж Буль

Пример:

```
freeDay = (d==6 or d==7)  
...  
if not freeDay:  
    print("Рабочий день.")  
else:  
    print("Выходной!")
```

Экспертная система

Экспертная система — это компьютерная программа, задача которой — заменить человека-эксперта при принятии решений в сложной ситуации.

База знаний = факты + правила вывода:

- если у животного есть перья, то это **птица**;
- если животное кормит детенышей молоком, то это — **млекопитающее**;
- если животное — млекопитающее и ест мясо, то это — **хищник**.

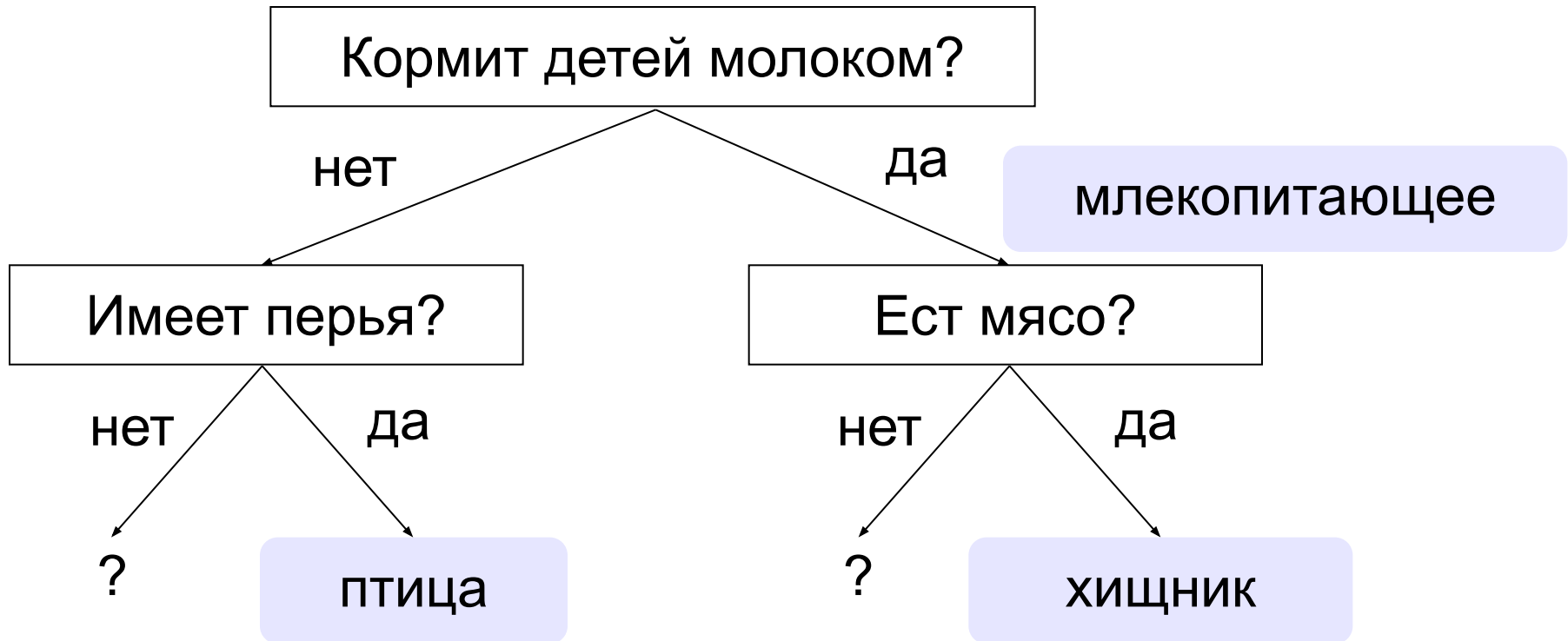
Диалог:

Это животное кормит детей молоком? **Нет**

Это животное имеет перья? **Да**

Это **птица**.

Дерево решений



Программирование экспертной системы

Ответы пользователя: **да** и **нет** – символьные строки.

```
ans = input("Кормит детей молоком? ")
if ans == "да":
    ... # вариант 1
else:
    ... #
```

```
    # вариант 1
    print("Млекопитающее.")
    ans = input("Ест мясо? ")
    if ans == "да":
        print("Хищник.")
    else:
        print("Не знаю.")
```

Заглавные и строчные буквы

```
if ans == "да":  
    ...
```

не сработает
на "Да"



Как исправить?

```
if ans == "да" or ans == "Да":  
    ...
```

Ещё лучше:

```
if ans.lower() == "да":  
    ...
```

преобразовать все
заглавные в строчные

```
if ans.upper() == "ДА":  
    ...
```