

Лекция 2. Условные операторы. Циклы

Оператор if .. elif .. else. Логические операторы в условиях (and, or, not, is, in). Условные выражения (тернарный оператор). Вложенные конструкции.

Циклы for и while. Различия и применение. Операторы прерывания цикла break и continue. Проход по списку задом наперед





Условия в python

Оператор if .. elif .. else. Логические операторы в условиях (and, or, not, is, in). Условные выражения (тернарный оператор). Вложенные конструкции

Условия в python

первая проверка

```
if условие_1:
```

```
    КОД_1
```

вторая и следующие
проверки

```
elif условие_2:
```

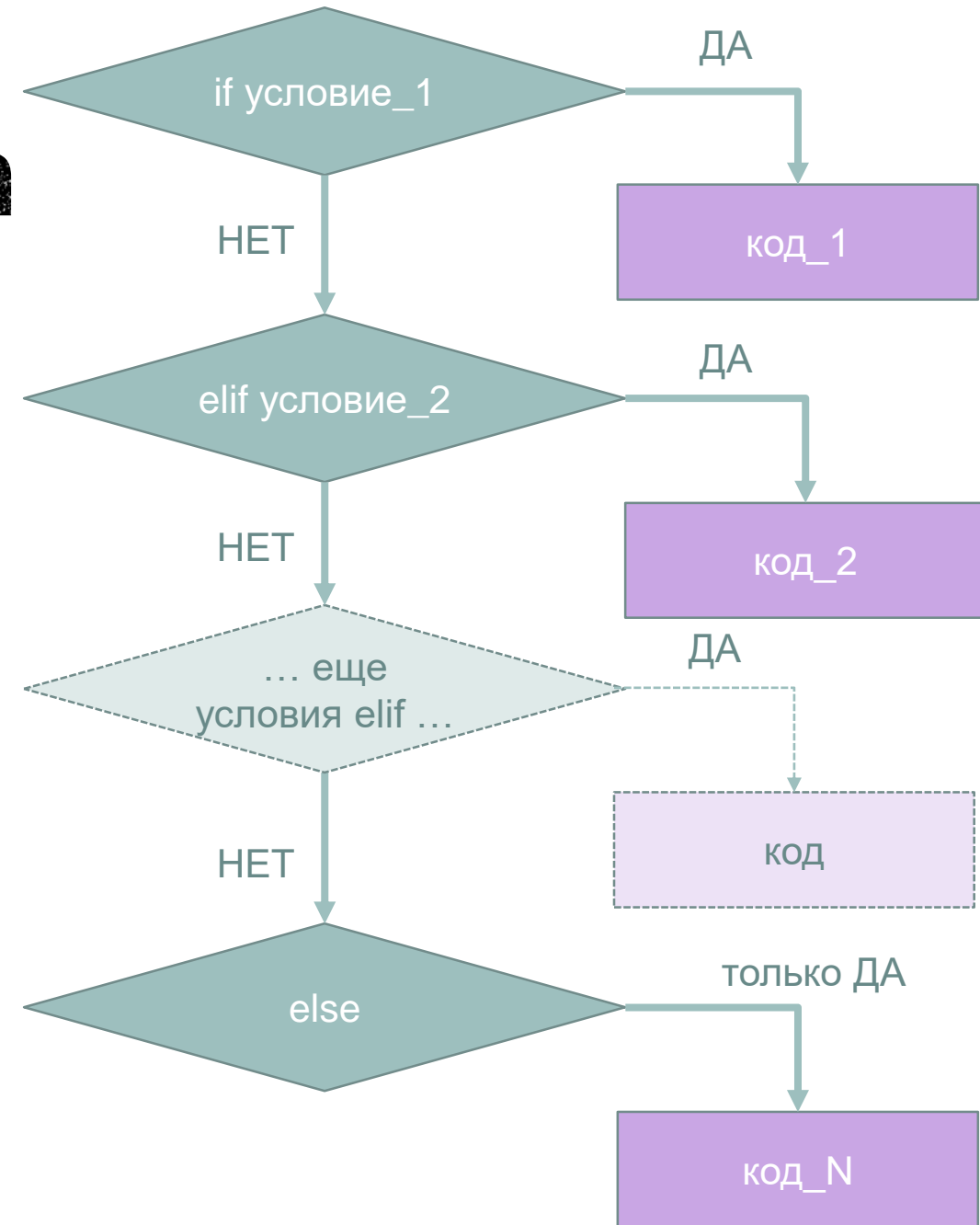
```
    КОД 2
```

последняя проверка

```
else:
```

```
    КОД_N
```

в других языках обычно пишут **else if**



Условия в python (продолжение)

```
1 # по правилам Росавиации регистрации подлежат все БВС массой более 150 г.  
   давайте проверим, нужно ли регистрировать наш...  
2 weight = 151  
3 ▼ if weight <= 150:  
4     print(f"БВС массой {weight}. Не нужно регистрировать")  
5 ▼ else:  
6     print(f"БВС массой {weight}. Нужно регистрировать")
```

БВС массой 151. Нужно регистрировать



Условия в python (продолжение)

```
1 # по правилам Росавиации регистрации подлежат все БВС массой более 150 г.  
  Если БВС массой более 30 кг, то есть особые правила. Давайте проверим,  
  что у нас...  
2 weight = 151  
3 ▼ if weight <= 150:  
4     print(f"БВС массой {weight}. Не нужно регистрировать")  
5 ▼ elif weight <= 30000:  
6     print(f"БВС массой {weight}. Нужно регистрировать")  
7 ▼ else:  
8     print(f"БВС массой {weight}. Нужно регистрировать и есть особые  
   правила")
```

БВС массой 151. Нужно регистрировать



Логические операторы

Их три:

- ❑ **and** (И). Оба операнда должны быть True
- ❑ **or** (ИЛИ). Хотя бы один из операндов должен быть True
- ❑ **not** (НЕ). Инверсия. Операнд не должен быть такой, как указано (инверсия)



Логические операторы (продолжение)

if условие_1 and условие_2: True

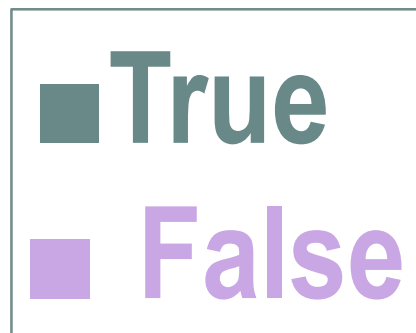
if условие_1 and условие_2: False

if условие_1 or условие_2: True

if условие_1 or условие_2: True

if not условие_1: False

if not условие_1: True



Логические операторы (продолжение)

```
11 # по правилам Росавиации не нужно согласовывать полет, если 1) он в
    прямой видимости, 2) на высоте не более 150 м, 3) выполняется вне
    закрытых зон. Проверяем, нужно ли нам согласование?
12 #vlos = visual line of site, прямая видимость
13 vlos = True
14 height = 120
15 closed_area = True
16 ▼ if vlos and height <=150 and not closed_area:
17     print("Не нужно согласовывать, летайте спокойно!")
18 ▼ else:
19     print("Нужно согласовывать, т.к. минимум одно условие требует
    согласования")
```

Нужно согласовывать, т.к. минимум одно условие требует согласования

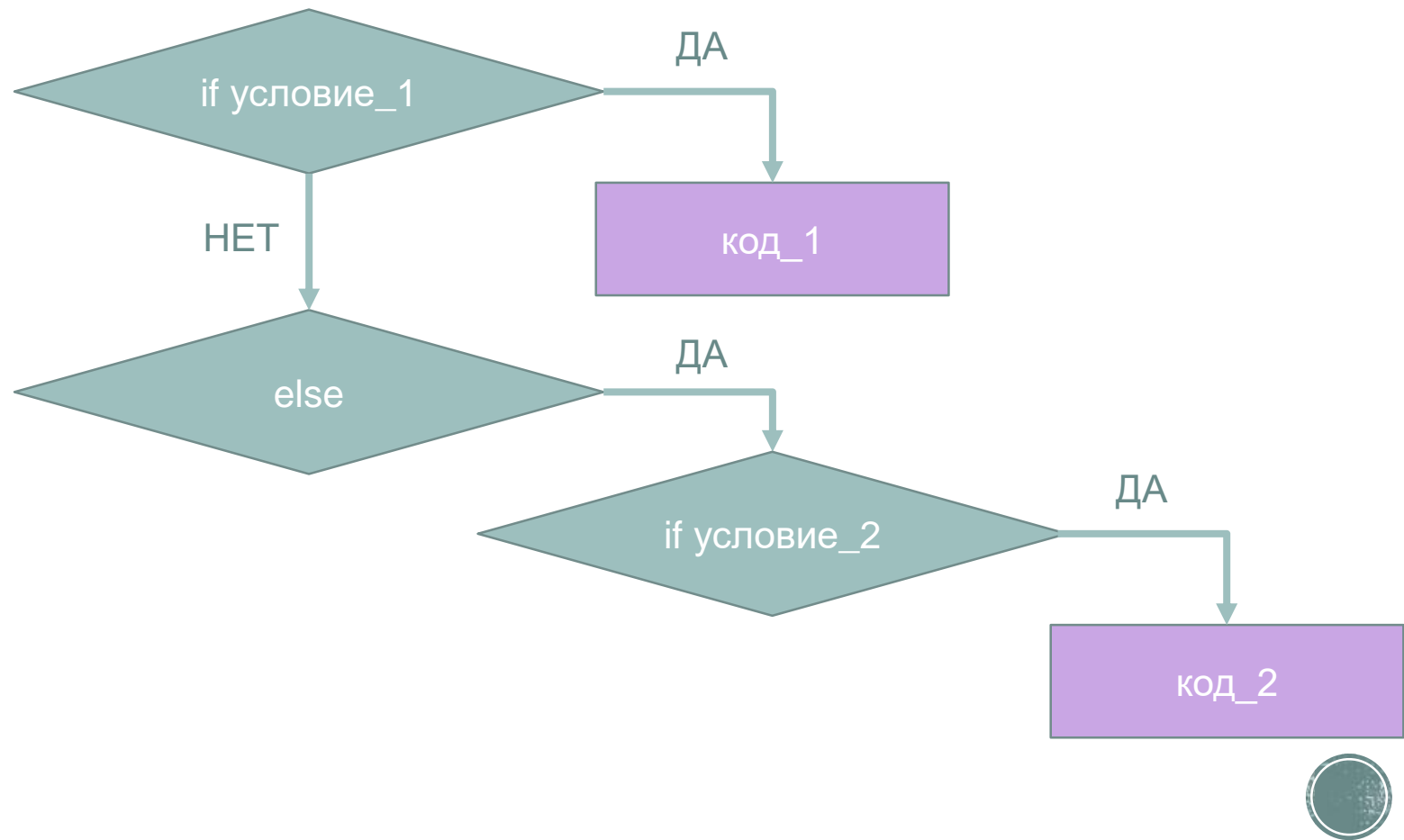


Вложенные условия

```
if условие_1:  
    КОД_1
```

```
else:
```

```
    if условие_2:  
        КОД_2
```



Вложенные условия (продолжение)

```
11 # по правилам Росавиации не нужно согласовывать полет, если 1) он в прямой видимости, 2) на высоте не
    более 150 м, 3) выполняется вне закрытых зон. Проверяем, нужно ли нам согласование?
12 #vlos = visual line of site, прямая видимость
13 vlos = True
14 height = 200
15 closed_area = True
16 ▼ if vlos and height <=150 and not closed_area:
17     print("Не нужно согласовывать, летайте спокойно!")
18 ▼ else:
19     conditions = ""
20 ▼     if not vlos:
21         conditions += "/полет вне зоны прямой видимости/"
22 ▼     if height > 150:
23         conditions += "/высота полета более 150 м/"
24 ▼     if closed_area:
25         conditions += "/полет в закрытой зоне/"
26     print(f"Нужно согласовывать, т.к. {conditions}")
```

Нужно согласовывать, т.к. /высота полета более 150 м//полет в закрытой зоне/



Тернарный оператор

■ условие

■ ВЫЧИСЛЕНИЕ

действие_Т if условие else действие_Ф

действие_Т if условие else
 действие_Т if условие else
 действие_Ф



Тернарный оператор (продолжение)

```
84 # могут летать без согласования БВС легче 150 г или БВС легче 30 кг
85 # в пределах прямой видимости не в населенных пунктах на высоте до 150 м
86 weights = [149, 9000, 151, 31000, 150]
87 no_perm = 0
88 locality = False
89 vlos = True
90 height = 120
91 reg_count = 0
```

```
104 ▼ for weight in weights:
105 ▼     if weight <= 150:
106         no_perm = no_perm + 1
107 ▼     elif weight <= 30_000 and not locality and vlos and height<150:
108         no perm = no perm + 1
109         reg_count = reg_count + 1
110 ▼     else:
111         reg_count = reg_count + 1
112
113 print(f"Могут летать без согласования {no_perm} БВС, нужно \
114 регистрировать {reg_count} БВС")
```

```
93 ▼ for weight in weights:
94     reg_count = reg_count+1 if weight <= 150 else reg_count
95     no_perm = (no_perm+ 1 if weight <= 150 else
96               no_perm+1 if (weight <= 30000 and not locality and
97                           vlos and height<150) else
98               no_perm)
99
100 print(f"Могут летать без согласования {no_perm} БВС, нужно \
101 регистрировать {reg_count} БВС")
```



Тернарный оператор (продолжение)

```
93 ▼ for weight in weights:
94     reg_count = reg_count+1 if weight <= 150 else reg_count
95     no_perm ← (no_perm+ 1 if weight <= 150 else
96               no_perm+1 if (weight <= 30000 and not locality and
97                           vlos and height<150) else
98               no_perm)
99
100 print(f"Могут летать без согласования {no_perm} БВС, нужно \
101 регистрировать {reg_count} БВС")
```





Циклы в python

Циклы for и while. Различия и применение. Операторы прерывания цикла break и continue. Проход по списку задом наперед

Циклы в python

В python два вида циклов:

- while

- for



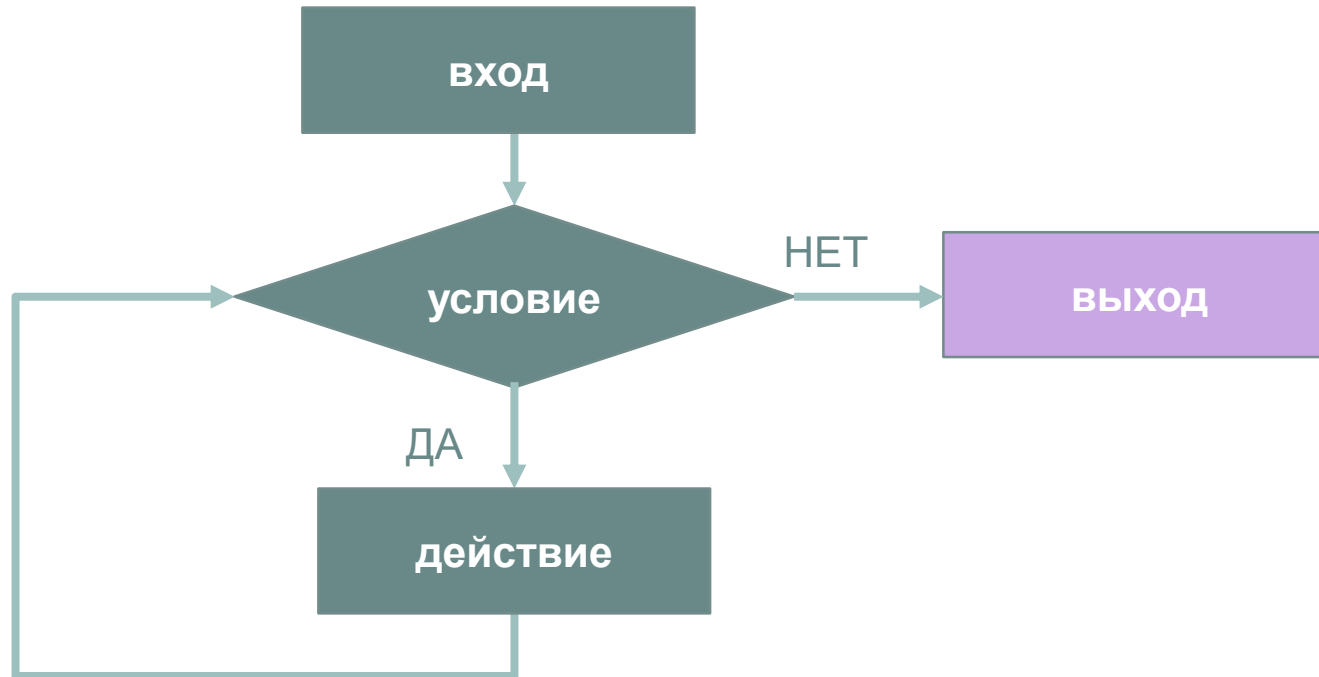
Цикл `while`

- ❑ сначала условие, потом работа
- ❑ может быть бесконечным
- ❑ ВЫХОД ИЗ ЦИКЛА – когда перестанет выполняться условие



Цикл `while` (продолжение)

`while` работает до тех пор, пока условие `True`



Цикл `while` (продолжение)

```
31 stop = False
32 ▼ while not stop:
33     user_input = input("Введите массу БВС в граммах или q, чтобы выйти\n")
34 ▼     if user_input.lower() == "q":
35         stop = True
36 ▼     else:
37         weight = int(user_input)
38 ▼         if weight <= 150:
39             print(f"БВС массой {weight} г. Не нужно регистрировать")
40 ▼         else:
41             print(f"БВС массой {weight} г. Нужно регистрировать")
```



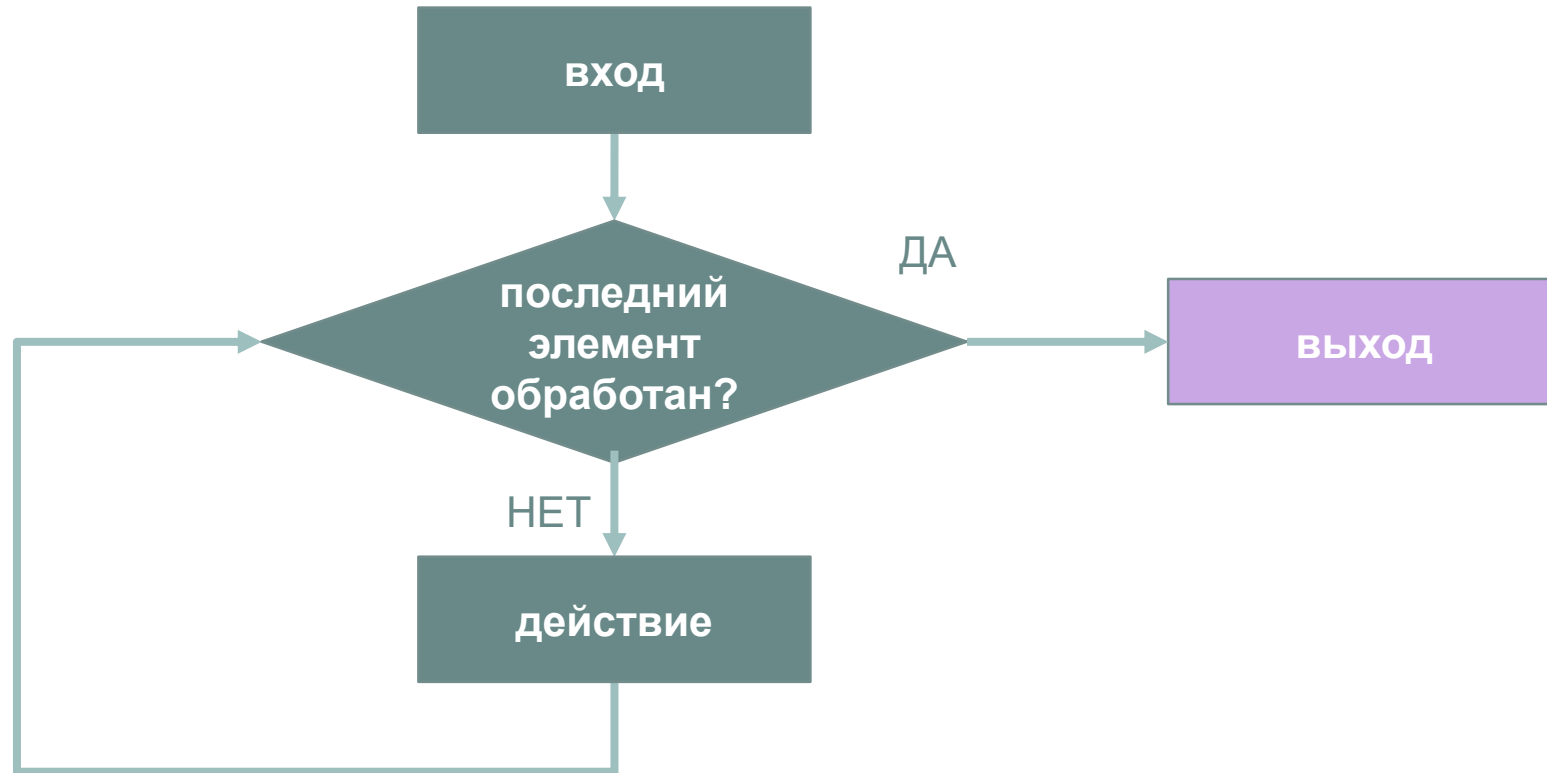
Цикл `for`

- ❑ всегда конечный
- ❑ перебирает элементы последовательности
- ❑ ВЫХОД ИЗ ЦИКЛА – когда все элементы последовательности закончатся



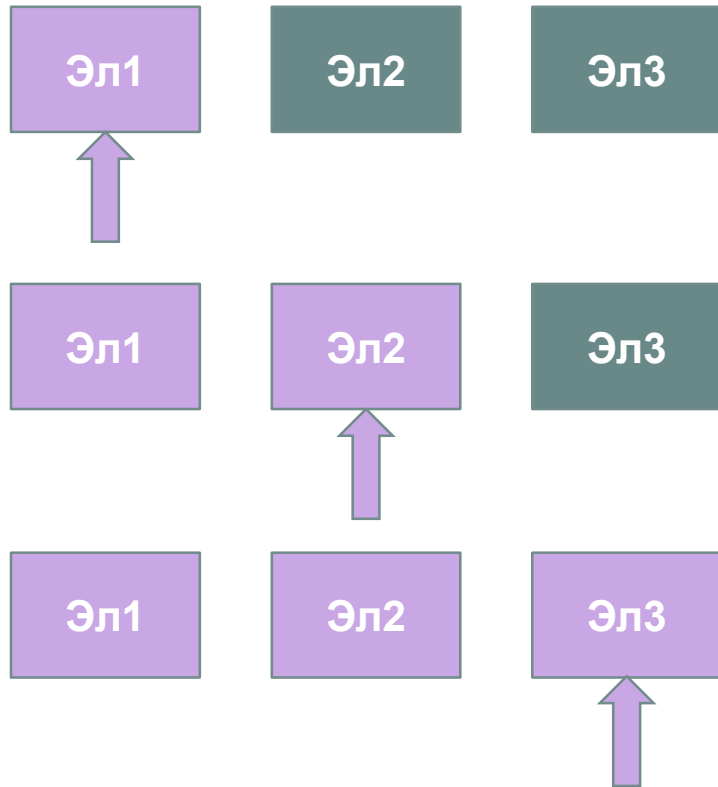
Цикл **for** (продолжение)

for работает до последнего элемента последовательности



Цикл `for` (продолжение)

`for` работает до последнего элемента последовательности



Цикл for (продолжение)

```
43 weights = [149, 9000, 151]
44 ▼ for weight in weights:
45 ▼     if weight <= 150:
46         print(f"БВС массой {weight} г. Не нужно регистрировать")
47 ▼     else:
48         print(f"БВС массой {weight} г. Нужно регистрировать")
```

weight = 149

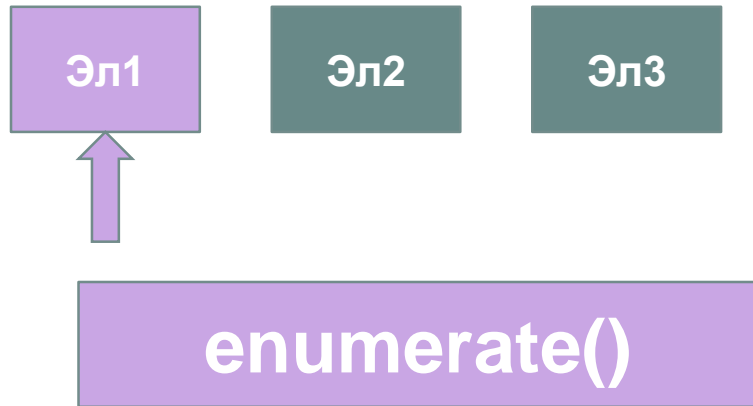
weight = 9000

weight = 151



Цикл `for` (продолжение)

Как узнать номер элемента в цикле `for`?



Цикл for (продолжение)

```
43 weights = [149, 9000, 151]
44 ▼ for id, weight in enumerate(weights):
45     print(f"Проход цикла {id}")
46 ▼     if weight <= 150:
47         print(f"БВС массой {weight} г. Не нужно регистрировать")
48 ▼     else:
49         print(f"БВС массой {weight} г. Нужно регистрировать")
```

id = 0, weight = 149

id = 1, weight = 9000

id = 2, weight = 151

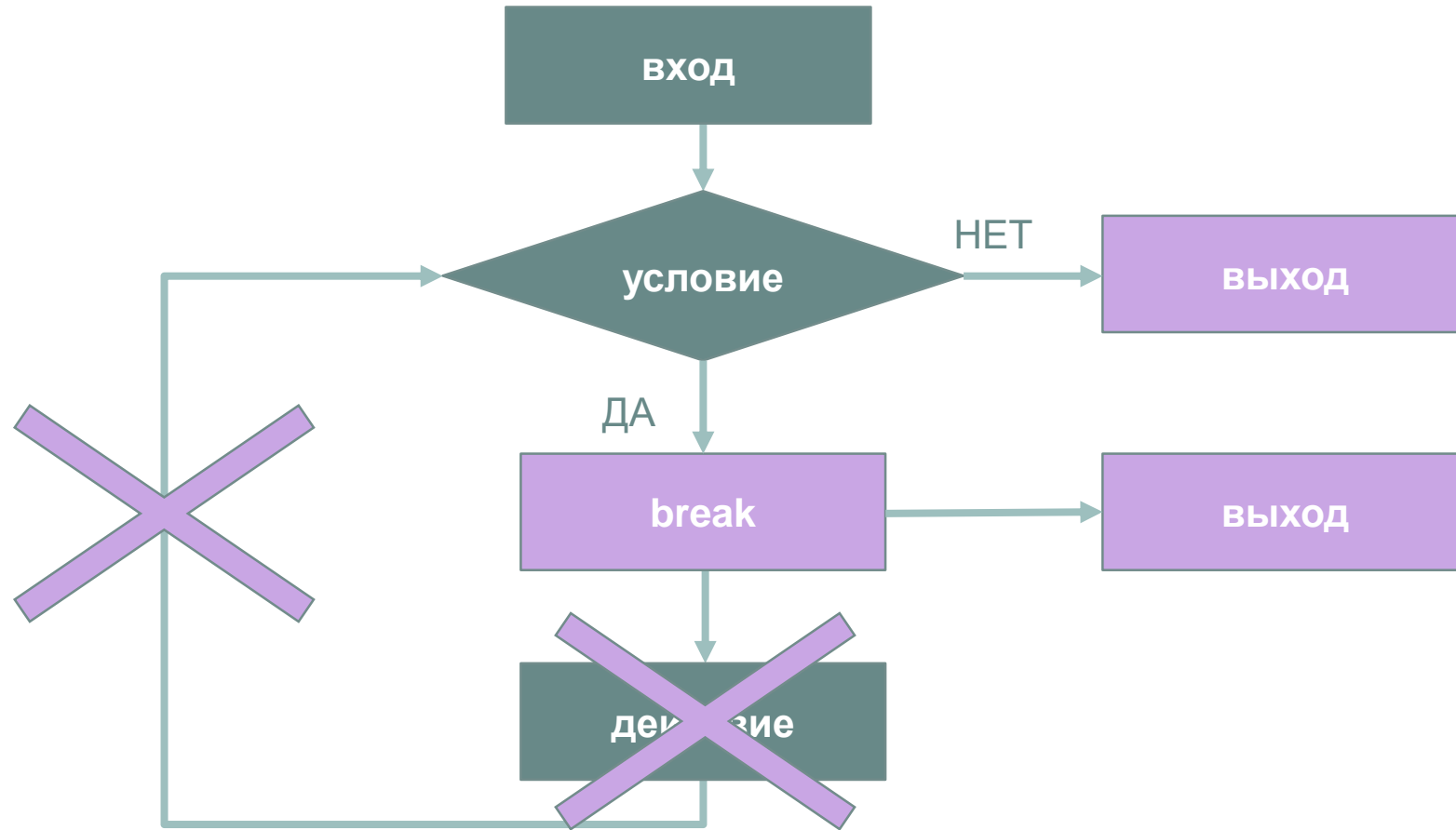


Прерывание цикла

- ❑ **break** – прервать и выйти из цикла
- ❑ **continue** – прервать только текущий проход цикла и выполнять цикл дальше



break в цикле



break в цикле (продолжение)

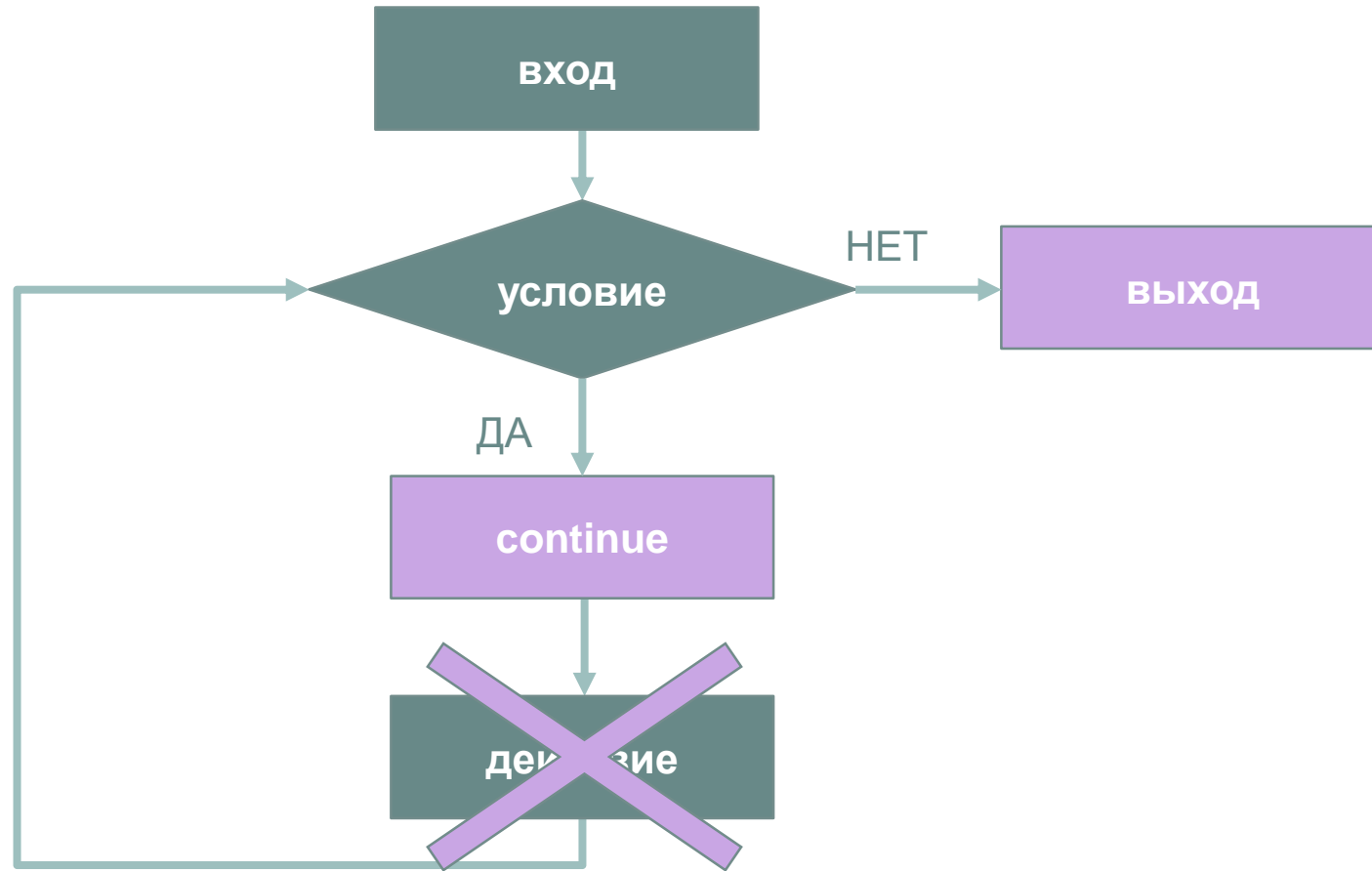
```
43 weights = [149, "9000", 151]
44 ▼ for id, weight in enumerate(weights):
45 ▼     if not isinstance(weight, int):
46         break
47     print(f"Проход цикла {id}")
48 ▼     if weight <= 150:
49         print(f"БВС массой {weight} г. Не нужно регистрировать")
50 ▼     else:
51         print(f"БВС массой {weight} г. Нужно регистрировать")
```

Проход цикла 0

БВС массой 149 г. Не нужно регистрировать



continue в цикле



continue в цикле (продолжение)

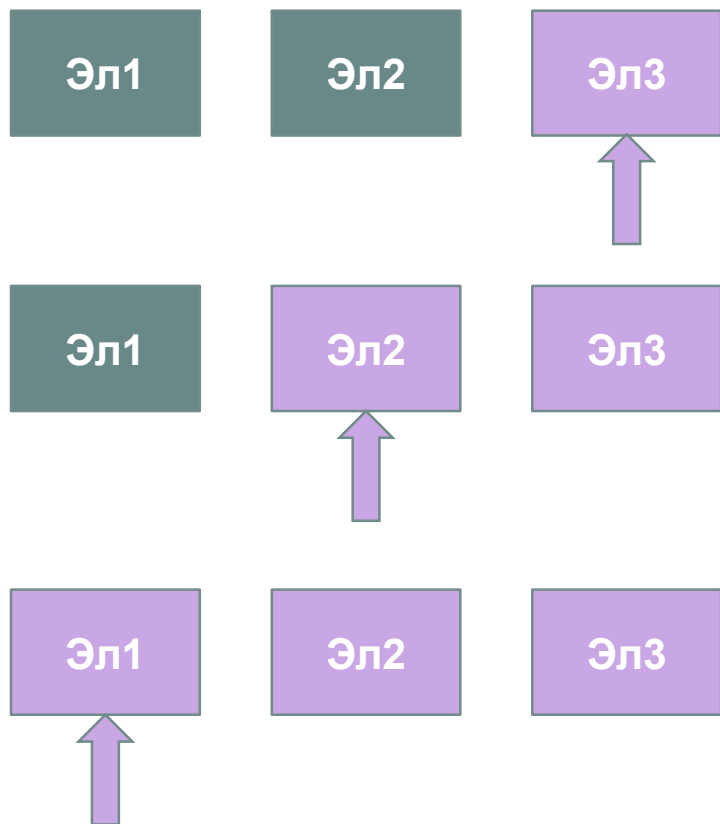
```
43 weights = [149, "9000", 151]
44 ▼ for id, weight in enumerate(weights):
45 ▼     if not isinstance(weight, int):
46         continue
47     print(f"Проход цикла {id}")
48 ▼     if weight <= 150:
49         print(f"БВС массой {weight} г. Не нужно регистрировать")
50 ▼     else:
51         print(f"БВС массой {weight} г. Нужно регистрировать")
```

```
Проход цикла 0
БВС массой 149 г. Не нужно регистрировать
Проход цикла 2
БВС массой 151 г. Нужно регистрировать
> □
```



Проход по циклу задом наперед

К последовательности нужно добавить `::-1`



Проход по циклу задом наперед (продолжение)

```
54 weights = [149, "9000", 151]
55 ▼ for id, weight in enumerate(weights[::-1]):
56     print(f"Проход цикла {id}, масса БВС {weight}")
```

```
Проход цикла 0, масса БВС 151
Проход цикла 1, масса БВС 9000
Проход цикла 2, масса БВС 149
➤ □
```





Домашнее задание

ТЕКСТ ДЗ