План на занятие:

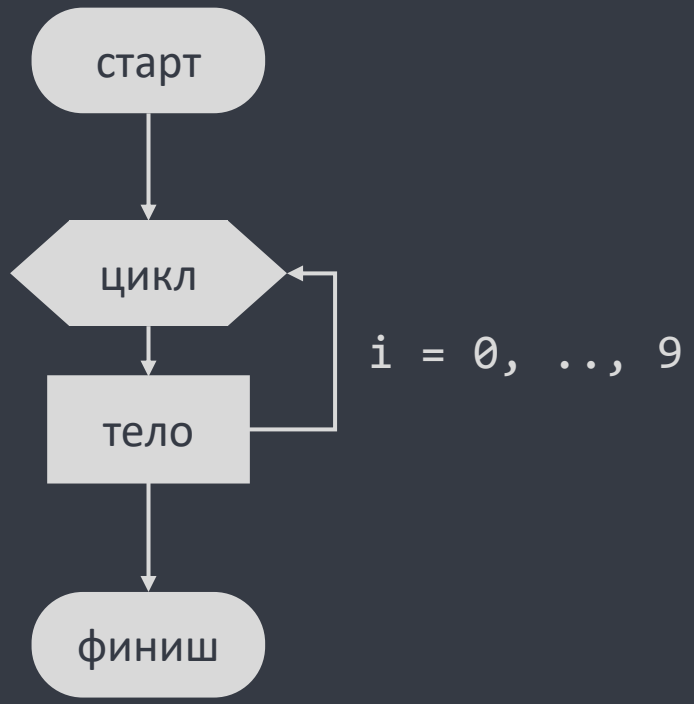
1. Итерации
    1.1. For .. in ..
    1.2. While ..
2. Логические выражения
    2.2. Типы данных: bool, NoneType
    2.2. Операторы сравнения
    2.3. Логические операции
4. Разные фишки по циклам в python

План на занятие:

1. Итерации
   - 1.1. For .. in ..
   - 1.2. While ..
2. Логические выражения
   - 2.2. Типы данных: bool, NoneType
   - 2.2. Операторы сравнения
   - 2.3. Логические операции
4. Разные фишки по циклам в python

Итерации и циклы

- многократное повторение действия



Циклы

Цикл for:

- позволяет итерироваться по элементам итериуемого объекта

Синтаксис:

```
for item in iterable:
    ____...
```

Пример:

```
1. models = ['A-152', 'A-8731', 'A-72']
   for model in models:
       print(model[2:])
   # '152'
     '8731'
     '72'
```

Циклы

Цикл for:

- позволяет итерироваться по элементам итериуемого объекта

Синтаксис:

```
for item in iterable:
    ____...
```

Пример:

```
models = ['A-152', 'A-8731', 'A-72']
for i in range(len(models)):
    print(models[i][2:])
# '152'
  '8731'
  '72'
```

Циклы

Цикл for:

- позволяет итерироваться по элементам итериуемого объекта

Синтаксис:

```
for item in iterable:
    ____...
```

Пример:

```
3. models = ['A-152', 'A-8731', 'A-72']
   for i in range(len(models)):
       models[i] = models[i].split('-')
   # [['A', '152'], ['A', '8731'], ['A', '72']]
```

Циклы

Цикл for:

- позволяет итерироваться по элементам итериуемого объекта

Синтаксис:

```
for item in iterable:
    ____...
```

Пример:

```
4. models = ['A-152', 'A-8731', 'A-72']
   for letter in models[0]:
       print(letter, end=' ')
   # A - 1 5 2
```

Циклы

Пример:

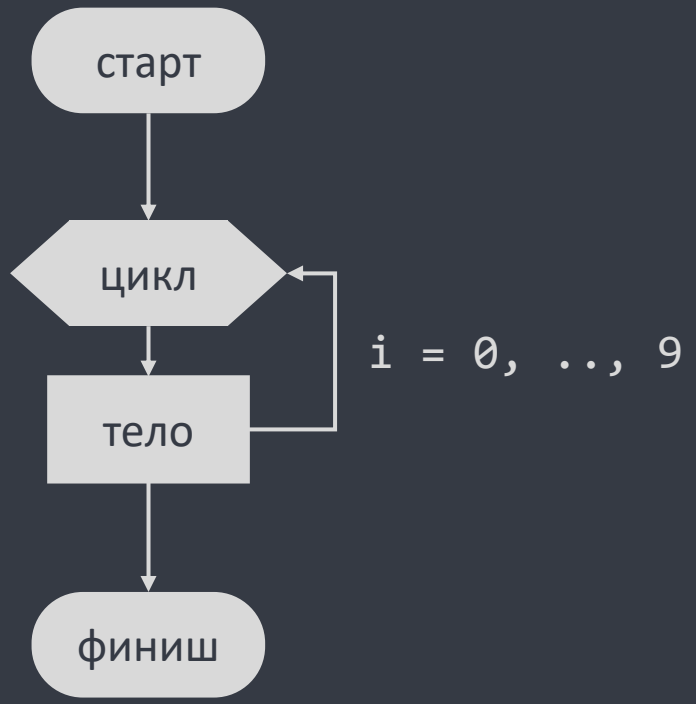
```
5. columns = ['name', 'surname', 'group']
   rows = [
       ['Elon', 'Mask', '201'],
       ['Jeff', 'Bezos', '201'],
       ['Steve', 'Ballmer', '212']
   ]
   separator = '\t'
   print(separator.join(columns))
   for row in rows:
       print(separator.join(row))
```

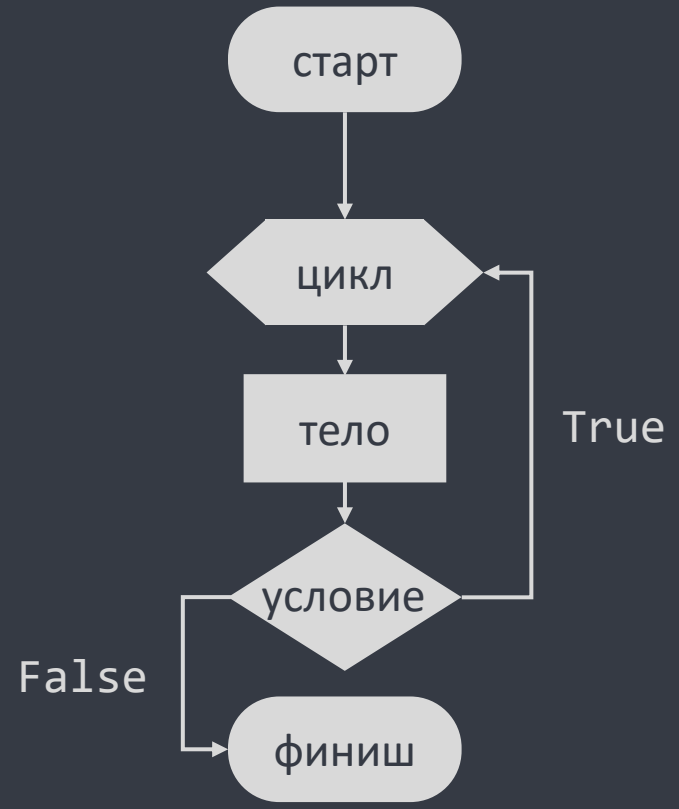
План на занятие:

1. Итерации
   - 1.1. For .. in ..
   - 1.2. While ..
2. Логические выражения
   - 2.2. Типы данных: bool, NoneType
   - 2.2. Операторы сравнения
   - 2.3. Логические операции
4. Разные фишки по циклам в python

Итерации и циклы

- многократное повторение действия





Циклы

Цикл while:

- выполняется до момента достижения условия

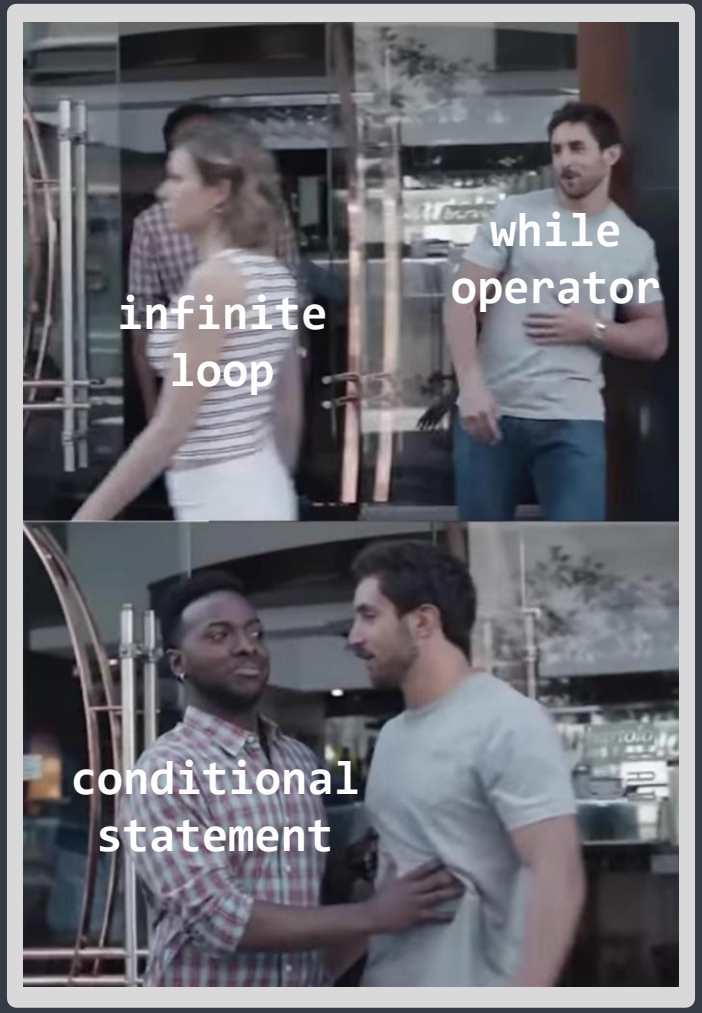
Синтаксис:

```
while <condition>:
    ____...
```

Пример:

```
1. while True:
       print('hi')
   # 'hi'
     'hi'
     ...
```

Циклы



Цикл while:

- выполняется до момента достижения условия

Синтаксис:

```
while <condition>:
    ____...
```

Пример:

2.
```
n = 15162
digits = 0
while n > 0:
    n //= 10
    digits += 1
print(digits)  # 5
```

Циклы

Цикл while:

- выполняется до момента достижения условия

Синтаксис:

```
while <condition>:
    ____...
```

Пример:

```
import random
found = False
while not found:
    i = random.randint(0, 100)
    if i == 5:
        found = True
```

План на занятие:

1. Итерации
   1.1. For .. in ..
   1.2. While ..
2. Логические выражения
   2.2. Типы данных: bool, NoneType
   2.2. Операторы сравнения
   2.3. Логические операции
4. Разные фишки по циклам в python

Логические выражения

Типы данных bool и NoneType:

- логические типы данных, позволяющие строить ветвления и создавать условия

Синтаксис:

```
True

False

None
```

План на занятие:

1. Итерации
   1.1. For .. in ..
   1.2. While ..
2. Логические выражения
   2.2. Типы данных: bool, NoneType
   2.2. Операторы сравнения
   2.3. Логические операции
4. Разные фишки по циклам в python

Логические выражения

Операторы сравнения:

Синтаксис:

```
>
<
>=
<=
==
!=
```

План на занятие:

1. Итерации
   - 1.1. For .. in ..
   - 1.2. While ..
2. Логические выражения
   - 2.2. Типы данных: bool, NoneType
   - 2.2. Операторы сравнения
   - 2.3. Логические операции
4. Разные фишки по циклам в python

Логические выражения

Проверки (ветвления):

- операторы, выбирающий вариант действия на основе результата проверки

Синтаксис:

```
if <condition>:
    ...
elif <condition>:
    ...
else:
    ...
```

Логические выражения

Пример:

```
1. models = ['A-152-old', 'B-8731', 'A-72']
   models_a = []
   for model in models:
       if model.split('-')[0] == 'A':
           models_a.append(model)
   print(models_a)
   # ['A-152-old', 'A-72']
```

Логические выражения

Пример:

2.
```
models = ['A-152-old', 'B-8731', 'A-72']
models_a = []
for model in models:
    if (model.split('-')[0] == 'A'
            and models[:-3] != 'old'):
        models_a.append(model)
print(models_a)
# ['A-72']
```

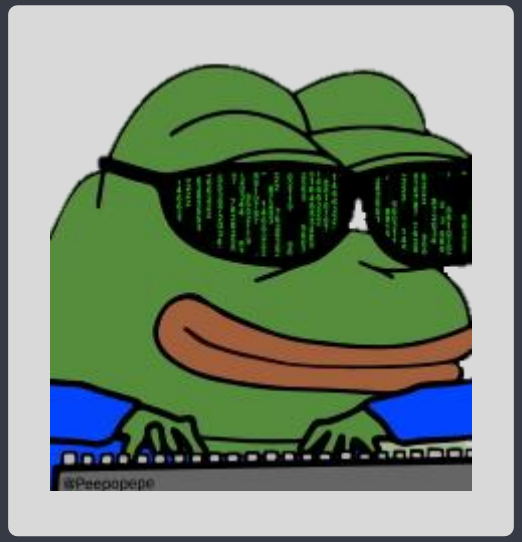
Логические выражения

`True and True`	`True`
`True and False`	`False`
`False and False`	`False`
`True or True`	`True`
`True or False`	`True`
`False or False`	`False`

Логические выражения

Пример:

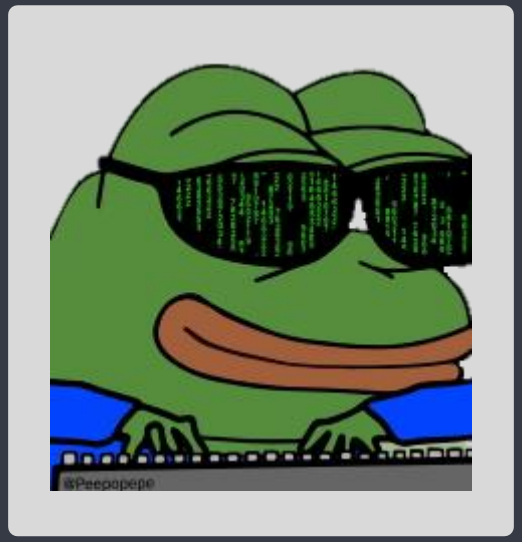
1. Определить к какой координатной четверти относится точка



Логические выражения

Пример:

2. Определить к какому направлению относится студент



План на занятие:

1. Итерации
   - 1.1. For .. in ..
   - 1.2. While ..
2. Логические выражения
   - 2.2. Типы данных: bool, NoneType
   - 2.2. Операторы сравнения
   - 2.3. Логические операции
4. Разные фишки по циклам в python

Разные фишки

List comprehensions:

- способ создания списков с итерированием и условиями

Синтаксис:

```
[expression for item in <iterable>]

[expression for item in <iterable>
 if item <condition> else item2]
```

Пример:

1.
```
models = ['A-152-old', 'B-8731', 'A-72']
models_a = [model for model in models
            if model.split('-')[0] == 'A']
print(models_a)  # ['A-152-old', 'B-8731', 'A-72']
```

Разные фишки

List comprehensions:

- способ создания списков с итерированием и условиями

Синтаксис:

```
[expression for item in <iterable>]

[expression for item in <iterable>
 if item <condition> else item2]
```

Пример:

2.
```
models = ['A-152', 'A-8731', 'A-72']

models = [
    [model.split('-')[0], int(model.split('-')[1]]
    for model in models]

# [['A', 152], ['A', 8731], ['A', 72]]
```

Разные фишки

List comprehensions:

- способ создания списков с итерированием и условиями

Синтаксис:

```
[expression for item in <iterable>]

[expression for item in <iterable>
 if item <condition> else item2]
```

Пример:

3.
```
close_price = [116.79, 116.56, 116.43, 116.34]
yields = [close_price[i+1] / close_price[i]
          for i in range(len(close_price) - 1)]
# [0.9980306533093587, 0.9988846945778999, 0.9992270033496521]
```