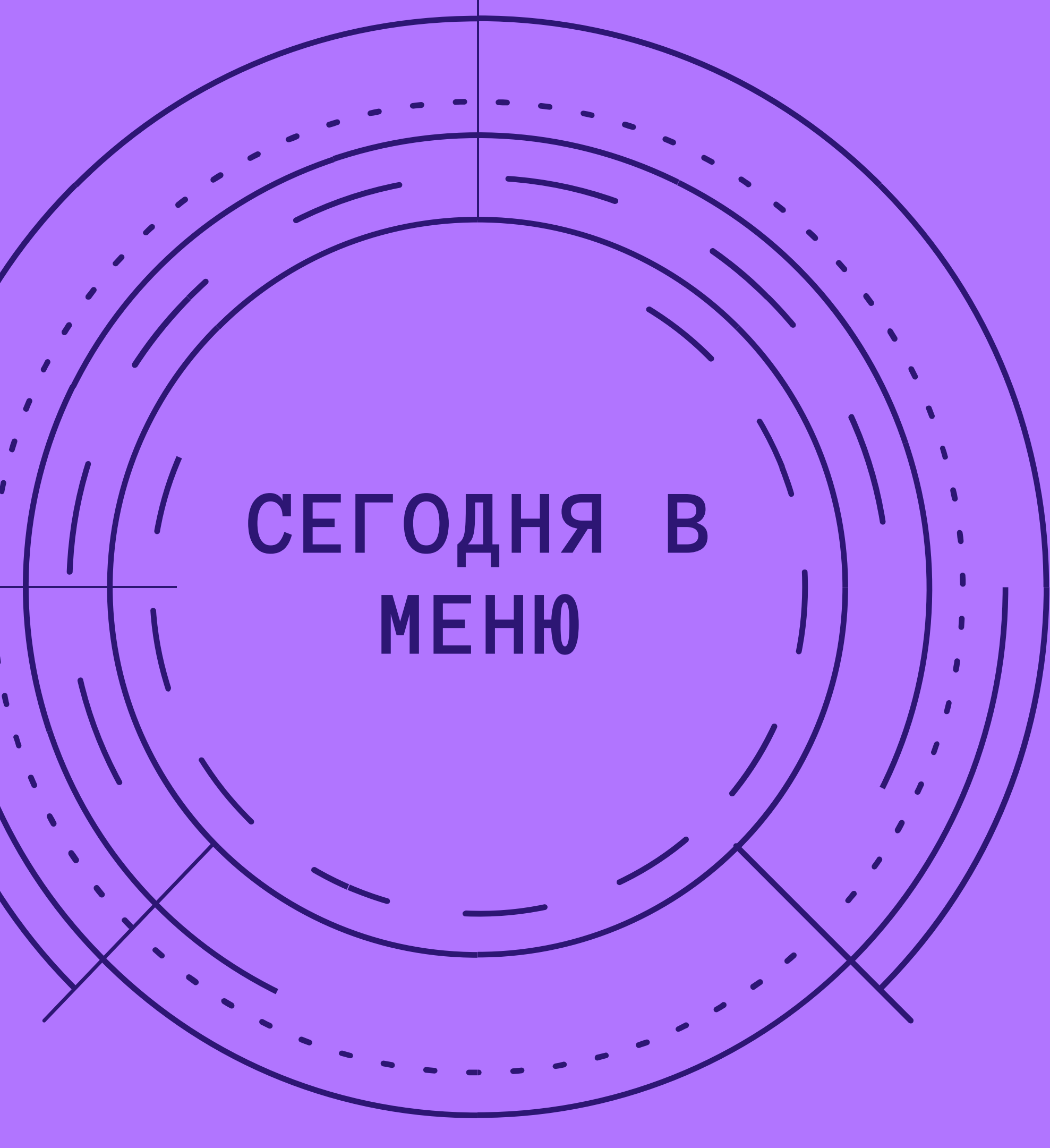


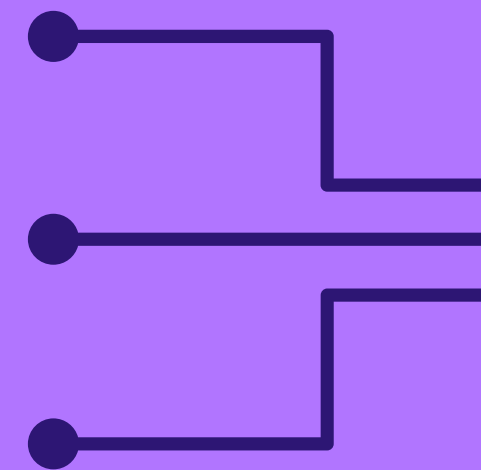
НЕДЕЛЯ_3: КЛАССЫ ЧАСТЬ 2



СЕГОДНЯ В МЕНЮ

СОДЕРЖАНИЕ :

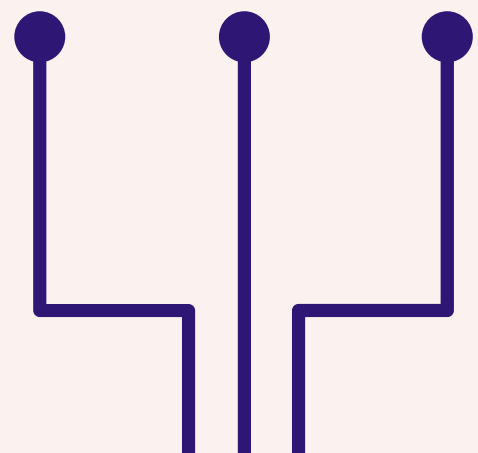
Наследование методов из других классов.
Атрибуты класса.
Приватные и Публичные методы.
Импортирование классов



НАСЛЕДОВАНИЕ ОДНА ИЗ 4 КОНЦЕПЦИЙ КЛАССА

Наследование – это возможность
класса пользоваться методами
другого класса.

Вам не нужно снова создавать
методы если они уже есть в
другом классе – просто
унаследуйте их.



```
class Java:
    def __init__(self):
        self.name = 'Java'
        self.paradigms = 'OOP'
        self.language_type = 'Compiled'

    def create_desktop(self):
        return 'Я могу создавать Desktop программы.'

    def create_game(self):
        return 'Я могу создавать Игры.'

    def create_site(self):
        return 'Я могу создавать Сайты.'
```

РОДИТЕЛЬСКИЙ КЛАСС

Каждый класс может быть
родительским для любого другого.

```
class Python(Java):
    def __init__(self):
        Java.__init__(self)
        self.name = 'Python'
        self.paradigms = ('OOP',)

p = Python()

print(p.create_site())
```

SUB КЛАСС

Каждый класс может наследовать любой
другой класс. Класс не может
наследовать и быть родителем для
одного и того же класса.

АТТРИБУТ – ЭТО ПЕРЕМЕННАЯ КЛАССА

Как и метод – атрибут можно
вызвать из класса и узнать его
значение.

Атрибуты можно изменять после
вызова. Благодаря атрибутам из
класса можно сделать обычный
лист.

```
1 class Python:
2     def __init__(self):
3         self.i_am_groot = 'I am Groot!'
4         self.i_am_attribute =
5         'I am attribute...'
6
7     def create_game(self):
8         return 'Я могу создавать Игры.'
9
10    def create_site(self):
11        return 'Я могу создавать Сайты.'
12
13 python = Python()
14 print(python.i_am_groot)
15 print(python.i_am_attribute)
```



```
self.name = 'Python'  
self.paradigms = ('OOP', 'Functional')  
self.language_type = 'Interpreted'
```

PUBLIC ATTRIBUTES

Публичные атрибуты можно менять
переписывать и вызывать.

```
self.__var = 'Я спрятался!'  
self.__year = '1991'  
self.__country = 'America'
```

PRIVATE ATTRIBUTES

Приватные атрибуты доступны
только и только классу.

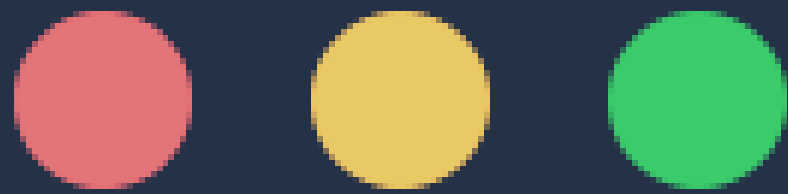


```
1  class SomeClass:
2      def __init__(self):
3          self.i_am_groot = 'I am Groot!'
4          self.i_am_attribute = 'I am attribute...'
5
6  some = SomeClass()
7  some.new_attr = 'New Value'
8  some.film = 'Avengers'
9  some.category = 'Fantastic'
10
11 print(some.i_am_groot)
12 print(some.i_am_attribute)
13 print(some.film)
14 print(some.category)
```

АТРИБУТЫ МОЖНО СОЗДАВАТЬ КАК В САМОМ КЛАССЕ ТАК И ПОСЛЕ ЕГО ИНИЦИАЛИЗАЦИИ

Класс можно рассматривать как
Dictionary.

Где ключ это атрибут а
значение – это объект справа
от равно.



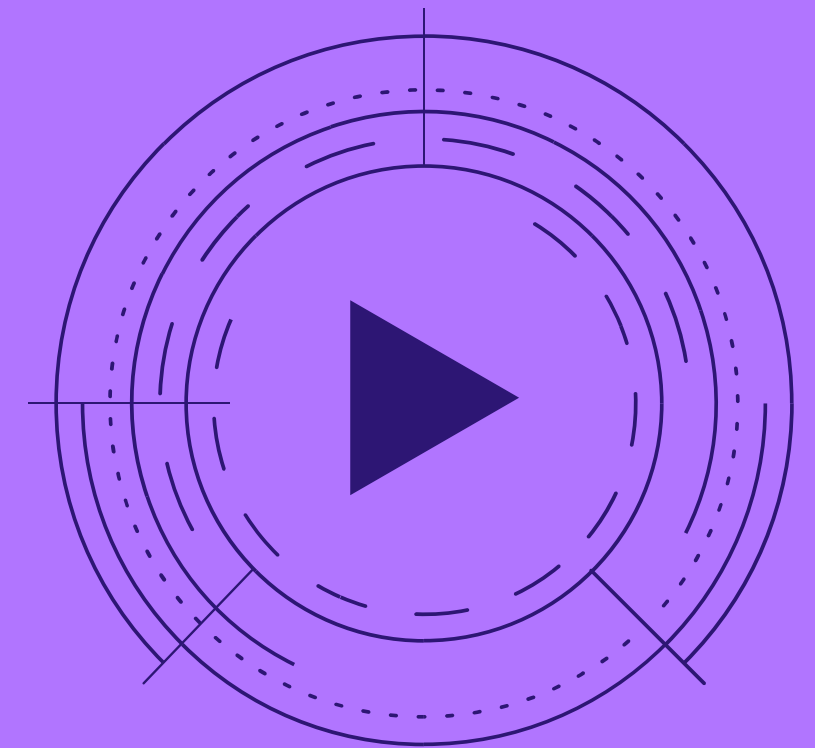
```
1  import os
2  import sys
3  from main import Python
```

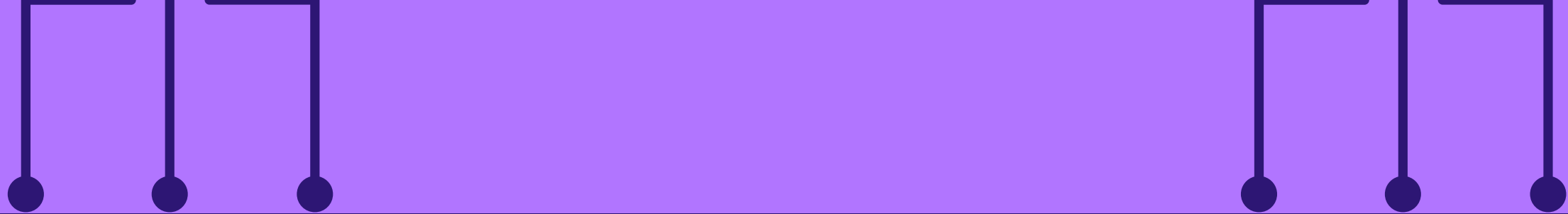

Классы можно импортировать как и модули

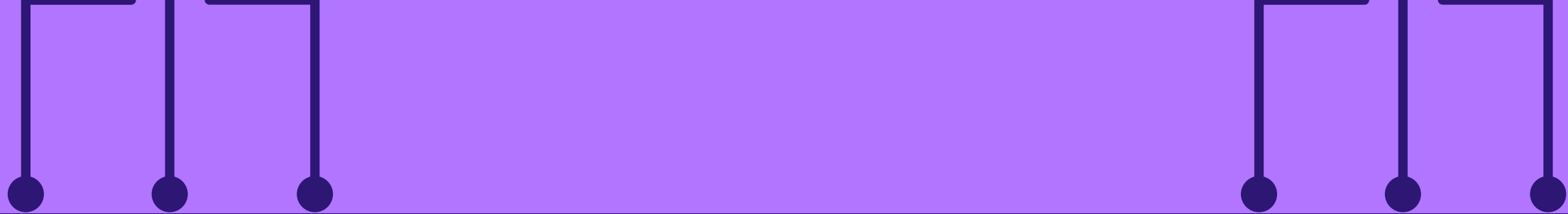
Для того чтобы импортировать класс укажите:

1. Модуль в котором он лежит
2. Имя Класса

```
from main import MyClass
```



- 
1. Создайте класс `Factory` и внутри создайте 2 метода:
 - a. Метод `engine` который возвращает строку "Двигатель создан"
 - b. Метод `bridge` который возвращает строку "Ходовая часть создана"
 2. Создайте класс `Toyota` который будет НАСЛЕДОВАТЬ класс `Factory`. В классе `Toyota` создайте методы `wheels` и `windows`.
 - a. Метод `wheels` возвращает строку "Колёса готовы"
 - b. Метод `windows` возвращает строку "Стёкла готовы"
- Из класса `Toyota` вызовите все методы, методы вернут вам строки(объекты)
Результат каждого метода вставьте в лист
- 

- 
1. Создайте Класс Zoo.
 2. Инициализируйте класс в объект.
 3. К объекту Zoo добавьте атрибут `animal_1` и присвойте ему значение "Тигр"
 4. К объекту Zoo добавьте атрибут `animal_2` и присвойте ему значение "Бегемот"
 5. К объекту Zoo добавьте атрибут `animal_3` и присвойте ему значение "Жираф"
 6. В объекте Zoo измените значение атрибута `animal_1` и присвойте ему значение "Лев".
 7. К объекту Zoo добавьте атрибут `animal_4` и присвойте ему list состоящий из `animal_2` и `animal_3`
 8. В объекте Zoo измените значение атрибута `animal_3` и присвойте ему значение "Змея".
- 