

Магические методы.

Шпаргалка

```
class BigFuncClass:

    def __init__(self, attr1, attr2):
        """
        Магический метод-конструктор для инициализации(создания)
        нового объекта
        """
        self.attr1 = attr1
        self.attr2 = attr2

    def __repr__(self):
        """
        Магический метод для отображения информации об объекте в режиме отладки
        """
        return f'{self.__class__.__name__}({self.attr1}, {self.attr2})'

    def __str__(self):
        """
        Магический метод для строкового отображения объекта
        """
        return f'{self.attr1} - {self.attr2}'

    def __len__(self):
        """
        Магический метод, который вызывается при применении функции len
        """
        return len(f'{self.attr1}{self.attr2}')

    def __add__(self, other):
        """
        Магический метод, который вызывается при сложении двух объектов
        """
        self.attr1 += other.attr1
        self.attr2 += other.attr2

    def __call__(self, *args, **kwargs):
        """
        Магический метод, который делает созданный объект вызываемым (callable)
        """
        print(f'Был вызван объект {self}'')
```

```
def __iter__(self):
    """
    Магический метод, который создает объект для итераций
    """
    self.current_value = -1
    return self

def __next__(self):
    """
    Магический метод, который возвращает очередной элемент
    при выполнении итерации
    """
    if self.current_value + 1 < len(self):
        self.current_value += 1
        return str(self)[self.current_value]
    else:
        raise StopIteration

def __enter__(self):
    """
    Магический метод, который вызывается при описании менеджера контекста
    """
    self.fp = open(self.attr1, self.attr2)
    return self.fp

def __exit__(self, exc_type, exc_val, exc_tb):
    """
    Магический метод, который содержит инструкции для выполнения
    при выходе из блока менеджера контекста
    """
    self.fp.close()

bfc1 = BigFuncClass('1', '2')
bfc2 = BigFuncClass('3', '4')
```