*"""1. Создайте класс, который называется Thing,  
не имеющий содержимого, и выведите его на экран.  
Затем создайте объект example этого класса и также  
выведите его. Совпадают ли выведенные значения?"""*class Thing:  
 pass  
  
print(Thing())  
  
example = Thing()  
print(example)  
  
"""  
<\_\_main\_\_.Thing object at 0x100f464a0>  
<\_\_main\_\_.Thing object at 0x100f464a0>  
Совпадают  
"""  
  
"""  
2. Создайте новый класс с именем Thing2 и присвойте его   
атрибуту letters значение 'abc'. Выведите на экран   
значение атрибута letters.  
"""  
class Thing2:  
 letters = 'abc'  
  
print(Thing2.letters)  
  
  
"""  
3.Создайте еще один класс, который, конечно же,   
называется Thing3. В этот раз присвойте значение   
'xyz' атрибуту объекта, который называется letters.   
Выведите на экран значение атрибута letters.   
Понадобилось ли вам создавать объект класса, чтобы   
сделать это?"""  
class Thing3:  
 def \_\_init\_\_(self):  
 self.letters = 'xyz'  
  
example = Thing3()  
print(example.letters)  
# Для вывода значения атрибута letters необходимо создать объект класса Thing3.  
  
"""  
4. Создайте класс, который называетсяElement,   
имеющий атрибуты объектаname, symbol и number.   
Создайте объект этого класса со значениями 'Hydrogen',   
'H' и 1.  
"""  
class Element:  
 def \_\_init\_\_(self, name, symbol, number):  
 self.name = name  
 self.symbol = symbol  
 self.number = number  
  
hydrogen = Element('Hydrogen', 'H', 1)  
  
  
"""  
5. Создайте словарь со следующими ключами и значениями: 'name': 'Hydrogen', 'symbol': 'H', 'number': 1.   
Далее создайте объект с именем hydrogen класса Element с помощью этого словаря.  
"""  
hydrogen\_dict = {'name': 'Hydrogen', 'symbol': 'H', 'number': 1}  
hydrogen = Element(\*\*hydrogen\_dict)  
  
  
"""6. Для класса Element определите метод с именем dump(), который выводит на экран значения   
атрибутов объекта (name, symbol и number). Создайте объект hydrogen из этого нового определения и используйте   
метод dump(), чтобы вывести на экран его атрибуты.  
"""  
  
  
class Element:  
 def \_\_init\_\_(self, name, symbol, number):  
 self.name = name  
 self.symbol = symbol  
 self.number = number  
  
 def dump(self):  
 print('Name:', self.name)  
 print('Symbol:', self.symbol)  
 print('Number:', self.number)  
  
  
hydrogen = Element('Hydrogen', 'H', 1)  
hydrogen.dump()  
  
"""  
7. Вызовите функцию print(hydrogen). В определении класса Element измените имя метода dump на str,   
создайте новый объект hydrogen и затем снова вызовите метод print(hydrogen).  
"""  
class Element:  
 def \_\_init\_\_(self, name, symbol, number):  
 self.name = name  
 self.symbol = symbol  
 self.number = number  
  
 def \_\_str\_\_(self):  
 return f'Name: {self.name}, Symbol: {self.symbol}, Number: {self.number}'  
  
  
hydrogen = Element('Hydrogen', 'H', 1)  
print(hydrogen)  
  
  
"""  
8. Модифицируйте класс Element, сделав атрибуты name, symbol и number закрытыми.   
Определите для каждого атрибута свойство получателя, возвращающее значение соответствующего атрибута.  
"""  
  
class Element:  
 def \_\_init\_\_(self, name, symbol, number):  
 self.\_\_name = name  
 self.\_\_symbol = symbol  
 self.\_\_number = number  
  
 @property  
 def name(self):  
 return self.\_\_name  
  
 @property  
 def symbol(self):  
 return self.\_\_symbol  
  
 @property  
 def number(self):  
 return self.\_\_number  
  
  
hydrogen = Element('Hydrogen', 'H', 1)  
print(hydrogen.name)  
print(hydrogen.symbol)  
print(hydrogen.number)  
  
"""  
9. Определите три класса: Bear, Rabbit и Octothorpe.   
Для каждого из них определите всего один метод — eats(). Он должен возвращать значения 'berries'   
(для Bear), 'clover' (для Rabbit) или 'campers' (для Octothorpe). Создайте по одному объекту каждого   
класса и выведите на экран то, что ест указанное животное.  
"""  
  
  
class Bear:  
 def eats(self):  
 return 'berries'  
  
class Rabbit:  
 def eats(self):  
 return 'clover'  
  
class Octothorpe:  
 def eats(self):  
 return 'campers'  
  
bear = Bear()  
rabbit = Rabbit()  
octothorpe = Octothorpe()  
  
print(bear.eats())  
print(rabbit.eats())  
print(octothorpe.eats())  
  
"""  
10. Определите три класса: Laser, Claw и SmartPhone. Каждый из них имеет только один метод — does().   
Он возвращает значения 'disintegrate' (для Laser), 'crush' (для Claw) или 'ring' (для SmartPhone).   
Далее определите класс Robot, который содержит по одному объекту каждого из этих классов.   
Определите метод does() для класса Robot, который выводит на экран все, что делают его компоненты  
"""  
  
  
class Laser:  
 def does(self):  
 return 'disintegrate'  
  
  
class Claw:  
 def does(self):  
 return 'crush'  
  
  
class SmartPhone:  
 def does(self):  
 return 'ring'  
  
  
class Robot:  
 def \_\_init\_\_(self):  
 self.laser = Laser()  
 self.claw = Claw()  
 self.smartphone = SmartPhone()  
  
 def does(self):  
 print('Laser does', self.laser.does())  
 print('Claw does', self.claw.does())  
 print('SmartPhone does', self.smartphone.does())  
  
  
robot = Robot()  
robot.does()