1. Создайте класс Calculator который поддерживает:

* сложение двух чисел
* вычисление разницы между двумя числами
* умножение двух чисел
* деление одного числа на другое
* возведение числа в степень

Примеры:

calculator = Calculator()

calculator.add(10, 5) ➞ 15

calculator.subtract(10, 5) ➞ 5

calculator.multiply(10, 5) ➞ 50

calculator.divide(10, 5) ➞ 2

# calculator.exponent(3, 2) ➞ 9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Создайте класс Pizza, который принимает список ингредиентов. Класс поддерживает:

* атрибут order\_number, который возвращает текущий номер заказа
* атрибут ingredients, который возвращает список, принятый в конструкторе
* функции (pepperoni, hawaiian, margherita) создания видов пицц, ингредиенты которых заранее известны (см. таблицу).

|  |  |
| --- | --- |
| **Name** | **Ingredients** |
| hawaiian | ham, pineapple |
| pepperoni | bacon, mozzarella, oregano |
| margherita | mozzarella, olives, tomatoes |

Примеры:

p1 = Pizza(['bacon', 'parmesan', 'ham']) # order 1

p2 = Pizza.pepperoni() # order 2

p1.ingredients ➞ ['bacon', 'parmesan', 'ham']

p2.ingredients ➞ ['bacon', 'mozzarella', 'oregano']

p1.order\_number ➞ 1

p2.order\_number ➞ 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Создайте класс Employee, который принимает имя, фамилию и зарплату в качестве аргументов при конструировании. При этом Employee является дочерним классом Human. Класс Human принимает first\_name и last\_name в качестве аргументов и реализует метод full\_name.

Класс Employee должен поддерживать:

* атрибут first\_name, возвращающий имя
* атрибут last\_name, возвращающий фамилию
* атрибут salary, возвращающий зарплату
* метод full\_name, который возвращает полное имя сотрудника.
* метод from\_string, которая принимает имя, фамилию и зарплату в формате 'first\_name-last\_name-salary', парсит строку и возвращает экземпляр Employee

*Также обратите внимание, что в методах/атрибутах, которые возвращают имя, фамилию или полное имя необходимо, чтобы эти данные возвращались в виде строки, которая начинается с заглавной буквы*

Примеры:

emp1 = Employee('JOAN', 'Smith', 85000)

emp2 = Employee.from\_string('John-doe-73000')

emp1.first\_name ➞ 'Joan'

emp1.full\_name ➞ 'Joan Smith'

emp1.salary ➞ 85000

emp2.first\_name ➞ 'John'

emp2.full\_name ➞ 'John Doe'

emp2.salary ➞ 73000