Гатин Дамир, 912 гр.

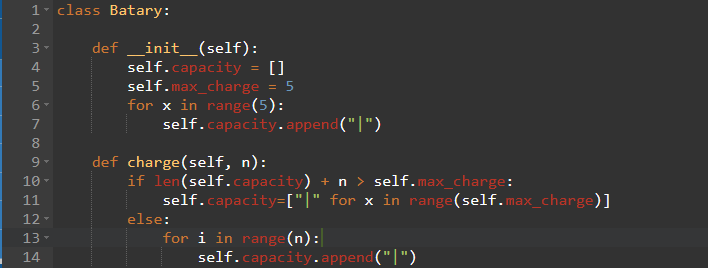
Урок Python №17 – ООП. ПОЛИМОРФИЗМ

Задание 1

Создайте класс Batary, у которой будет определен атрибут capacity = [ ] (емкость), max\_charge = 5 (максимальный заряд) по умолчанию, и методы:

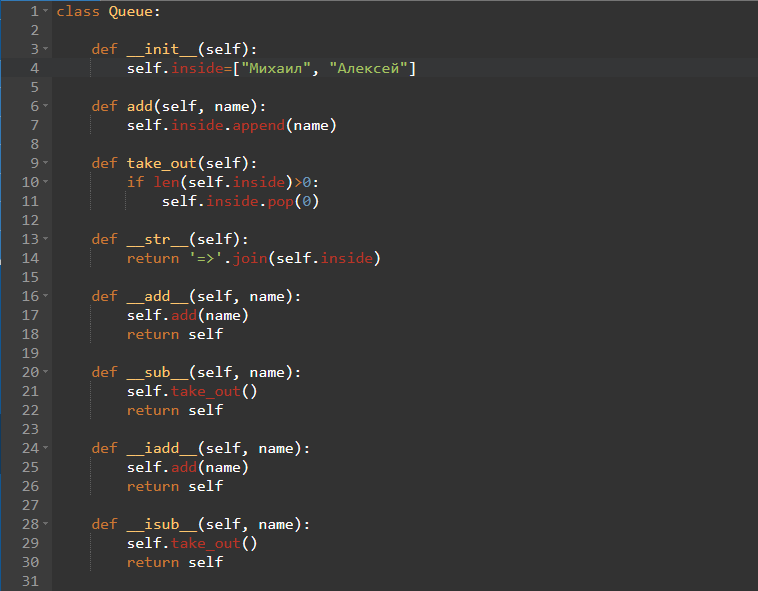
* charge - заряжает батарею
* discharge - разряжает батарею.
* Переопределите метод \_\_str\_\_, чтобы при вызове экземпляра он представлялся в виде: [)))))] - максимально заряженная батарея.

Подсказка: можете использовать методы очень похожего стандартного типа данных. Догадались какого?



Задание 2

* Реализовать класс Queue
* Определить атрибут inside, который будет хранить в себе имена людей в очереди.
* Переопределить метод \_\_str\_\_, чтобы преобразовать его к виду: Name1=>Name2=>...=>Name3
* Реализовать методы:
  + add - который добавляет имя в очередь
  + take\_out убирает первого человека из списка
* Переопределить методы \_\_add\_\_ , \_\_sub\_\_, \_\_iadd\_\_, \_\_isub\_\_ чтобы они соответствовали методам add и take\_out

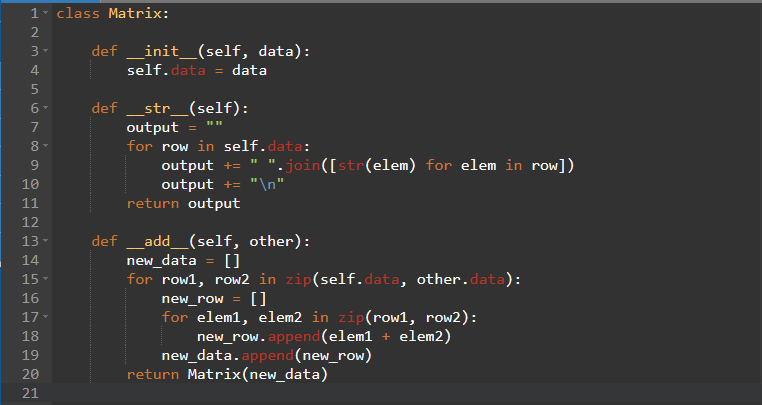


Задание 3

а) Создайте класс Matrix который должен принимать данные (список списков) для формирования матрицы.

b) Следующий шаг реализовать перегрузку метода \_\_str\_\_() для вывода матрицы в привычном виде.

c) Далее реализовать перегрузку метода \_\_add\_\_() для сложения двух объектов класса Matrix (двух матриц). Результатом сложения должна быть новая матрица.



Задание 4

Реализуйте класс Deque - двух стороннюю очередь или колоду. Идея та же, что и у очереди, только можно вставлять элементы в центр массива убирать как слева так и справа.

* Реализуйте класс Deque
* Переопределите метод \_\_str\_\_
* Реализуйте методы, который позволят вставлять элементы слева, справа, в в центр массива и аналогично удалять.

