jupyter shopping_cart.py a minute ago

Logout

File Edit View Language Python

```python
class ShoppingCart:

    def __init__(self, emp_discount=None):
        self._total = 0
        self._employee_discount = emp_discount
        self._items = []

    @property
    def items(self):
        return self._items

    @items.setter
    def items(self, list_of_items):
        self._items = list_of_items
        return self.items

    @property
    def employee_discount(self):
        return self._employee_discount

    @employee_discount.setter
    def employee_discount(self, new_employee_discount):
        self._employee_discount = new_employee_discount
        return self.employee_discount

    @property
    def total(self):
        return self._total

    @total.setter
    def total(self, new_total):
        self._total = new_total
        return self.total

    def add_item(self, name, price, quantity=1):
        for i in list(range(quantity)):
            self._items.append({"name": name, "price": price})
            self.total += price
        return self.total

    def mean_item_price(self):
        num_items = len(self.items)
        total = self.total
        mean = total/num_items
        return mean

    def median_item_price(self):
        prices = [self.get_attr(item, "price") for item in self.items]
        prices.sort()
        return self.find_median(prices)

    def find_median(self, list_of_prices):
        length = len(list_of_prices)
        if (length%2 == 0):
            mid_one = int(length/2)
            mid_two = mid_one - 1
            median = (list_of_prices[mid_one] + list_of_prices[mid_two])/2
            return median
        mid = int(length/2)
        return list_of_prices[mid]

    def apply_discount(self):
        if self.employee_discount:
            discount = self.employee_discount/100
            disc_total = self.total * (1 - discount)
            return disc_total
        else:
            return "Sorry, there is no discount to apply to your cart :("

    def get_attr(self, item, attr):
        return item[attr]

    def item_names(self):
        names = [self.get_attr(item, "name") for item in self.items]
        return names

    def void_last_item(self):
        if self.items:
            removed_item = self.items.pop()
        else:
            return "There are no items in your cart!"
        self.total -= removed_item['price']
```