ivanovbroo@gmai…

Цифровая кафедра. Веб-разработка 2022-2023

24 ноя 2022, 12:48:12
старт: 10 окт 2022, 11:03:58
начало: 25 сен 2022, 14:59:00

17. Классы 2. Кошельки

Ограничение времени	10 секунд
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Написать классы, которые будут использованы как счета в банке. Каждый счет - в своей валюте. Соответственно, у каждого объекта счета должны быть атрибуты с суммой денег, хранящихся на нём, и название кошелька. Каждый класс счета должен в себе хранить коэффициент отношения стоимости своей валюты к базовой валюте.
Нам понадобится один базовый класс BaseWallet, в котором будут реализованы общие для всех валютных счетов методы, и три класса для конкретных валютных счетов: RubbleWallet, DollarWallet, EuroWallet. Будем считать коэффициентами отношения валют к базовой валюте:

- Рубль: 1
- Доллар: 60
- Евро: 70

Протокол взаимодействия объектов следующий:

- RubbleWallet("Первый кошелек", 10), где "Первый кошелек" - это название кошелька, а 10 - сумма денег на нём.
- аналогичные конструкторы для других счетов
- RubbleWallet("X", 10) + 20 == RubbleWallet("X", 30) - при сложении с числом считаем, что это та же валюта.
- RubbleWallet("X", 10) += 20 - должен поддерживаться и такой синтаксис
- 20 + RubbleWallet("X", 10) == RubbleWallet("X", 30) - radd для чисел
- RubbleWallet("X", 20) + DollarWallet("D", 10) == RubbleWallet("X", 620) - конвертация валюты при сложении счетов.
- DollarWallet("D", 2) + RubbleWallet("X", 60) == DollarWallet("D", 3) - результат - в валюте первого слагаемого.
- DollarWallet("D", 2) += RubbleWallet("X", 60) - здесь тоже должен поддерживаться этот синтаксис.
- предыдущие 6 пунктов реализовать и для вычитания
- RubbleWallet("X", 10) * 20 == RubbleWallet("X", 200) - умножение на число
- RubbleWallet("X", 10) *= 20 - тоже с таким синтаксисом
- те же 2 пункта для деления
- 20 * RubbleWallet("X", 10) == RubbleWallet("X", 200) - умножение числа на кошелек
- DollarWallet("A", 15) == DollarWallet("B", 15): два объекта равны, если у них совпадает тип кошелька и сумма на счете.
- RubbleWallet("X", 100).spend_all() - для любого типа кошелька релизовать функцию, которая обнуляет баланс, если он положительный
- DollarWallet("X", 1).to_base() == 60 - эта функция должна возвращать число денег в кошельке в базовой валюте
- print(DollarWallet("Q", 150)) - должна выводить строку: 'Dollar Wallet Q 150' (и аналогично Rubble и Euro для остальных кошельков)

У каждого объекта должны быть доступны атрибуты:

- name - название кошелька
- amount - количество денег на счете
- exchange_rate - коэффициент стоимости валюты к базовой

Набрать здесь | Отправить файл

```
class BaseWallet:

    def _copy(self):
        return BaseWallet(self.name, self.amount)

    def __init__(self, name, amount):
        self.name =  name
        self.amount = amount

    def __add__(self, other):
        tmp = self._copy()
        if isinstance(other, BaseWallet):

            if(other._currency == "dol"):
                convert = other.amount * other.exchange_rate
                tmp.amount = self.amount + float(convert)

            if(other._currency == "euro"):
                convert = other.amount * other.exchange_rate
                tmp.amount = self.amount + float(convert)

            if(other._currency == "rub"):
                convert = other.amount / self.exchange_rate
                tmp.amount = self.amount + float(convert)
        else:
            tmp.amount = self.amount + float(other)

        return tmp

    def __iadd__(self, other):
        if isinstance(other, BaseWallet):

            if(other._currency == "dol"):
                convert = other.amount * other.exchange_rate
                self.amount += float(convert)

            if(other._currency == "euro"):
                convert = other.amount * other.exchange_rate
```

Отправить

Предыдущая Следующая