

# Цифровая кафедра. Веб-разработка 2022-2023

20 ноя 2022, 20:30:29

старт: 10 окт 2022, 11:03:58

начало: 25 сен 2022, 14:59:00

## 12. Функции. Переиспользование кода

Ограничение времени	10 секунд
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этом задании вам нужно описать несколько функций, которые будут использовать друг друга в своем коде. Функции нужно описать подряд, в одном файле, и отправить в контест все функции сразу.

### Постановка задачи

Вы с друзьями периодически собираетесь на лавке у подъезда выпить чаю с баранками. Каждый раз собирается разное количество человек, которое должно скинуть по  $n$  рублей на мероприятие. Также у вас в компании есть общак, в который можно докинуть денег, а можно взять. Все транзакции в общак логируются, т.е. у вас есть информация о каждом переводе денег в/из общака для каждого из друзей. Ваша задача - посчитать, сколько денег должен каждый из собравшихся на мероприятие друзей на момент начала мероприятия. Будем считать, что все деньги проходят через общак, поэтому у каждого друга к началу мероприятия уже есть некоторый "баланс" в общаке.

### Формат хранения данных

Все операции с деньгами хранятся в базе данных и приходят к вам в виде списка словарей вида:

```
[{"name": "Василий", "amount": 500}, {"name": "Петя", "amount": 100}, {"name": "Василий", "amount": -300}, ]
```

где name - имя друга (считаем, что всех зовут по-разному), amount - сумма, которая добавлена в общак. Если сумма отрицательная - это значит, что друг взял деньги из общака. Также считаем, что скидываются суммы в рублях без копеек.

### Вам нужно описать две функции.

1. `get_balance(name, transactions) -> int`
  - функция, которая возвращает текущий баланс друга с именем name, исходя из списка транзакций transactions. Если имя name ни разу не встречается в списке transactions, считаем, что баланс этого друга в общаке равен 0 рублей.
2. `count_debts(names, amount, transactions) -> dict`
  - функция, которая принимает список имен присутствующих на мероприятии друзей names, стоимость баранок и чая на человека amount, а также список транзакций в общак transactions. Вернуть эта функция должна словарь вида {"имя\_друга": 100}, где 100 - это количество денег, которое он должен скинуть на мероприятие. Если на балансе друга больше денег, чем требуется на мероприятие, то он должен 0 рублей.

### Формат ввода

```
transactions = [ {"name": "Василий", "amount": 500}, {"name": "Петя", "amount": 100}, {"name": "Василий", "amount": -300}, ]
```

```
get_balance("Василий", transactions)
```

```
count_debts(["Василий", "Петя", "Вова"], 150, transactions)
```

### Формат вывода

```
get_balance("Василий", transactions) == 200
```

```
count_debts(["Василий", "Петя", "Вова"], 150, transactions) == {"Василий": 0, "Петя": 50, "Вова": 150}
```

```
1 def sort_listdict_for_dict(raw_dict):
2     sort_dict = {}
3
4     for i in raw_dict:
5         tmp_name = ""
6         for key, value in i.items():
7             if(key == "name"):
8                 if(sort_dict.get(value) == None):
9                     sort_dict[value] = 0
10                    tmp_name = value
11                if(key == "amount"):
12                    sort_dict[tmp_name] += value
13
14     return sort_dict
15
16
17 def get_balance(name, transactions):
18     name_amount_dict = sort_listdict_for_dict(transactions)
19
20     for key, value in name_amount_dict.items():
21         if(key == name):
22             return value
23
24     return 0
25
26
27 def count_debts(names, amount, transactions):
28     name_amount_dict = sort_listdict_for_dict(transactions)
29     party_dict = {}
30
31     for name in names:
32         if(name_amount_dict.get(name) == None):
33             party_dict[name] = amount
34             continue
35         if(name_amount_dict.get(name) >= amount):
36             party_dict[name] = 0
37             continue
38         if(name_amount_dict.get(name) < amount):
```