

# Цифровая кафедра. Веб-разработка 2022-2023

20 ноя 2022, 20:31:15  
старт: 10 окт 2022, 11:03:58  
начало: 25 сен 2022, 14:59:00

## 16. Классы 1. Калькулятор

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

Опишите класс Calculator, который будет реализовывать следующие методы и поля:

- `sum(self, a, b)` - сложение чисел `a` и `b`
- `sub(self, a, b)` - вычитание
- `mul(self, a, b)` - умножение
- `div(self, a, b, mod=False)` - деление. Если параметр `mod == True`, то метод должен возвращать остаток от деления вместо деления. По умолчанию `mod=False`.
- `history(self, n)` - этот метод должен возвращать строку с операцией по ее номеру относительно текущего момента (1 - последняя, 2 - предпоследняя). Формат вывода: `sum(5, 15) == 20`
- `last` - строка того же формата, что в предыдущем пункте, в которой содержится информация о последней операции по всем созданным объектам калькулятора. Т.е. это последняя операция последнего использованного объекта калькулятор. Если операций пока не было, то `None`.
- `clear(cls)` - метод, который очищает `last`, т.е. присваивает ему значение `None`.

### Формат вывода

При сохранении строк в `history` и `last` нужно выводить только один знак после запятой дробного числа. При выполнении деления с `mod` сам параметр `mod` не нужно записывать в лог.

```
1 class Calculator:
2
3     last = None
4
5     def __init__(self):
6         self._hist = list()
7
8     def sum(self, a, b):
9         result = a + b
10        output = f"sum({a}, {b}) == {round(result, 1)}"
11        Calculator._last_operation(output)
12        self._hist.append(output)
13        return result;
14
15    def sub(self, a, b):
16        result = a - b
17        output = f"sub({a}, {b}) == {round(result, 1)}"
18        Calculator._last_operation(output)
19        self._hist.append(output)
20        return result;
21
22    def mul(self, a, b):
23        result = a * b
24        output = f"mul({a}, {b}) == {round(result, 1)}"
25        Calculator._last_operation(output)
26        self._hist.append(output)
27        return result;
28
29    def div(self, a, b, mod=False):
30        if(mod == True):
31            result = a % b
32        else:
33            result = a / b
34
35        output = f"div({a}, {b}) == {round(result, 1)}"
36
37        Calculator._last_operation(output)
38        self._hist.append(output)
```

[Отправить](#)[Предыдущая](#)[Следующая](#)