

Лабораторная работа №3

```
class SimpleCalculator:
    """
    Класс для выполнения базовых математических операций с заданной точностью.
    """

    def __init__(self, precision: float = 10**-6):
        """
        Конструктор класса SimpleCalculator.

        Аргументы:
        precision -- точность вычислений (по умолчанию 1e-6)
        """
        self.precision = precision

    def get_precision_order(self):
        """
        Получает порядок точности на основе заданного значения precision.

        Возвращает:
        int -- порядок точности (количество цифр после запятой)
        """
        order = 0 # Инициализируем счетчик порядка
        current_precision = self.precision # Сохраняем текущее значение точности
        # Пока текущее значение precision не является целым числом
        while (current_precision - int(current_precision) != 0):
            order += 1 # Увеличиваем порядок
            current_precision *= 10 # Умножаем на 10 для перехода к следующей цифре
        return order

    def add(self, operand1, operand2):
        """
        Складывает два числа.

        Аргументы:
        operand1 -- первое слагаемое
        operand2 -- второе слагаемое

        Возвращает:
        float -- сумма operand1 и operand2
        """
        return operand1 + operand2

    def multiply(self, operand1, operand2):
        """
        Умножает два числа.

        Аргументы:
        operand1 -- первое множитель
        operand2 -- второе множитель

        Возвращает:
        """
```

```

float -- произведение operand1 и operand2
"""

return operand1 * operand2

def subtract(self, operand1, operand2):
    """
    Вычитает одно число из другого.

    Аргументы:
    operand1 -- уменьшаемое
    operand2 -- вычитаемое

    Возвращает:
    float -- разность operand1 и operand2
    """
    return operand1 - operand2

def divide(self, operand1, operand2):
    """
    Делит одно число на другое.

    Аргументы:
    operand1 -- делимое
    operand2 -- делитель

    Возвращает:
    float -- результат деления, округленный до указанной точности

    Исключения:
    ValueError -- если operand2 равен нулю
    """
    if operand2 != 0:
        order = self.get_precision_order() # Получаем порядок точности
        return round(operand1 / operand2, order) # Делим и округляем результат
    else:
        raise ValueError("Делить на 0 нельзя") # Исключение при делении на ноль

# Пример использования
result = SimpleCalculator(0.001).divide(3, 5) # Создаем объект калькулятора и делим 1
на 3
print(result) # Выводим результат деления

```