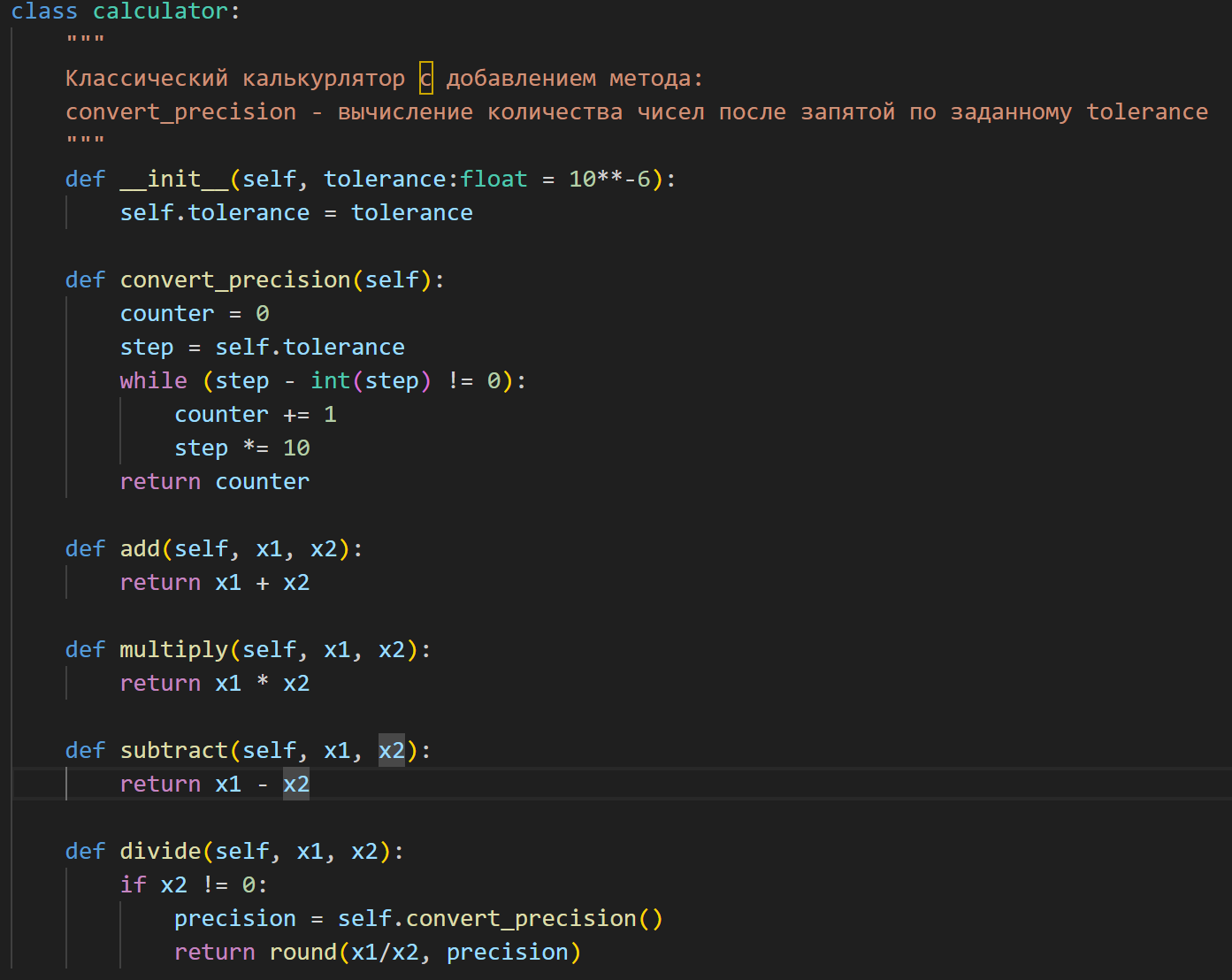
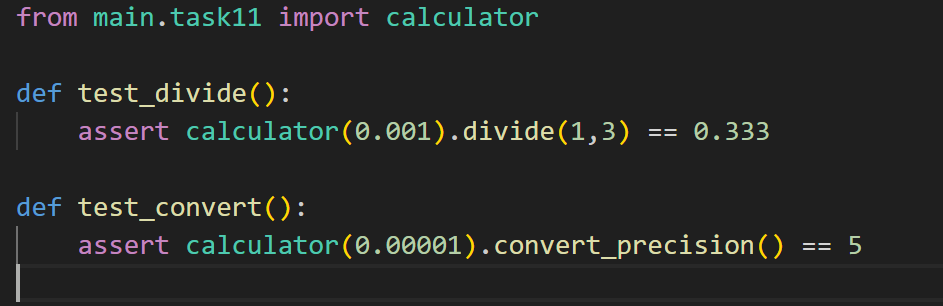
**Задача 1.**

****

****

**1. Назначение программы**

Программа реализует класс calculator с функционалом:

* Стандартные арифметические операции (сложение, вычитание, умножение, деление).
* Метод convert\_precision для вычисления точности округления на основе параметра tolerance.
* Автоматическое округление результатов деления до вычисленной точности.  
  Код протестирован с использованием **pytest** в виртуальном окружении venv.

**2. Описание работы кода**

**2.1. Класс calculator**

**Назначение:** Выполнение арифметических операций с контролем точности.

**Атрибуты класса:**

* tolerance (float): Параметр для определения точности округления (по умолчанию: 10^-6).

**Методы класса:**

1. **convert\_precision()**
   * Вычисляет количество знаков после запятой для округления на основе tolerance.
   * **Логика работы:**
     + Умножает tolerance на 10 до тех пор, пока дробная часть не станет равной нулю.
     + Пример: tolerance = 0.001 → точность = 3.
2. **divide(x1, x2)**
   * Возвращает результат деления x1 / x2, округленный до вычисленной точности.
   * При делении на ноль возвращает None.
3. Стандартные методы:
   * add(x1, x2): Сложение.
   * multiply(x1, x2): Умножение.
   * subtract(x1, x2): Вычитание.

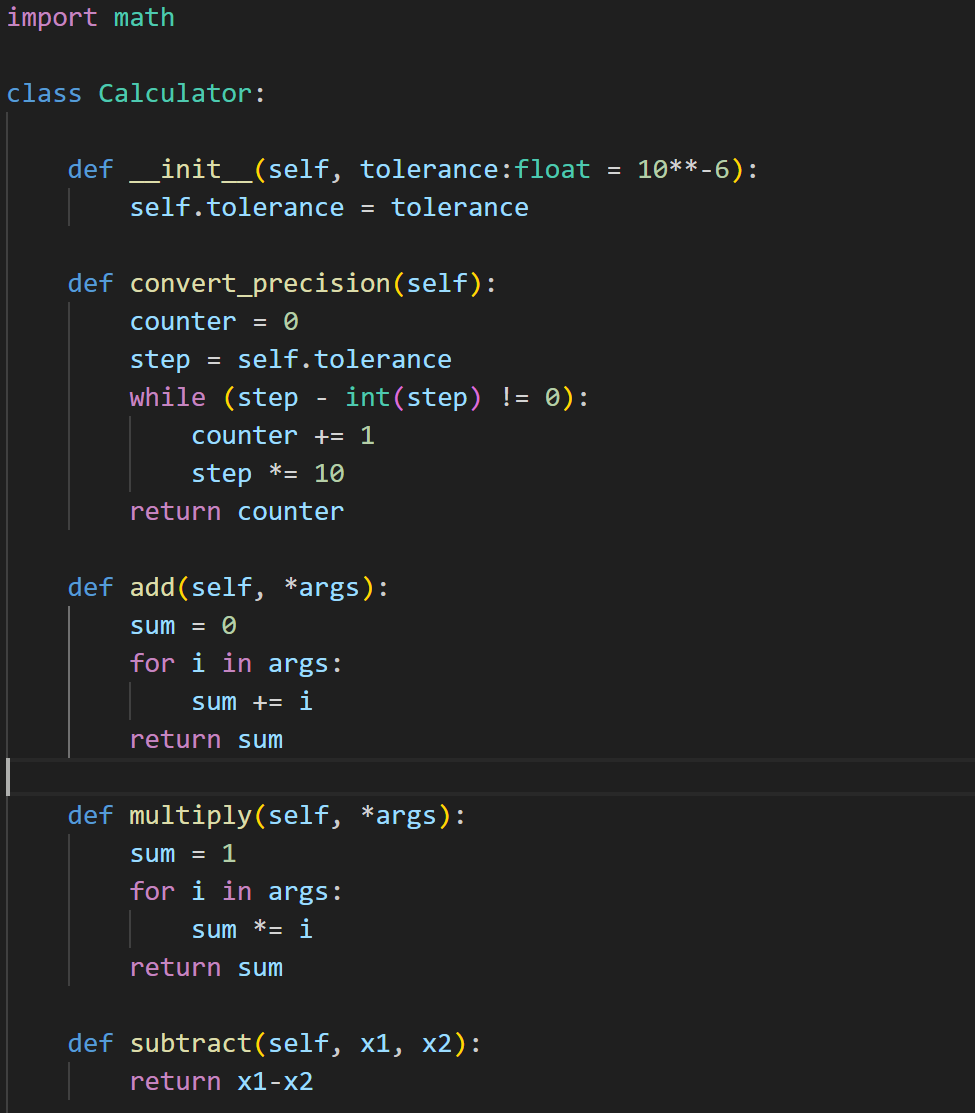
**2.2. Тестирование (файл тестов)**

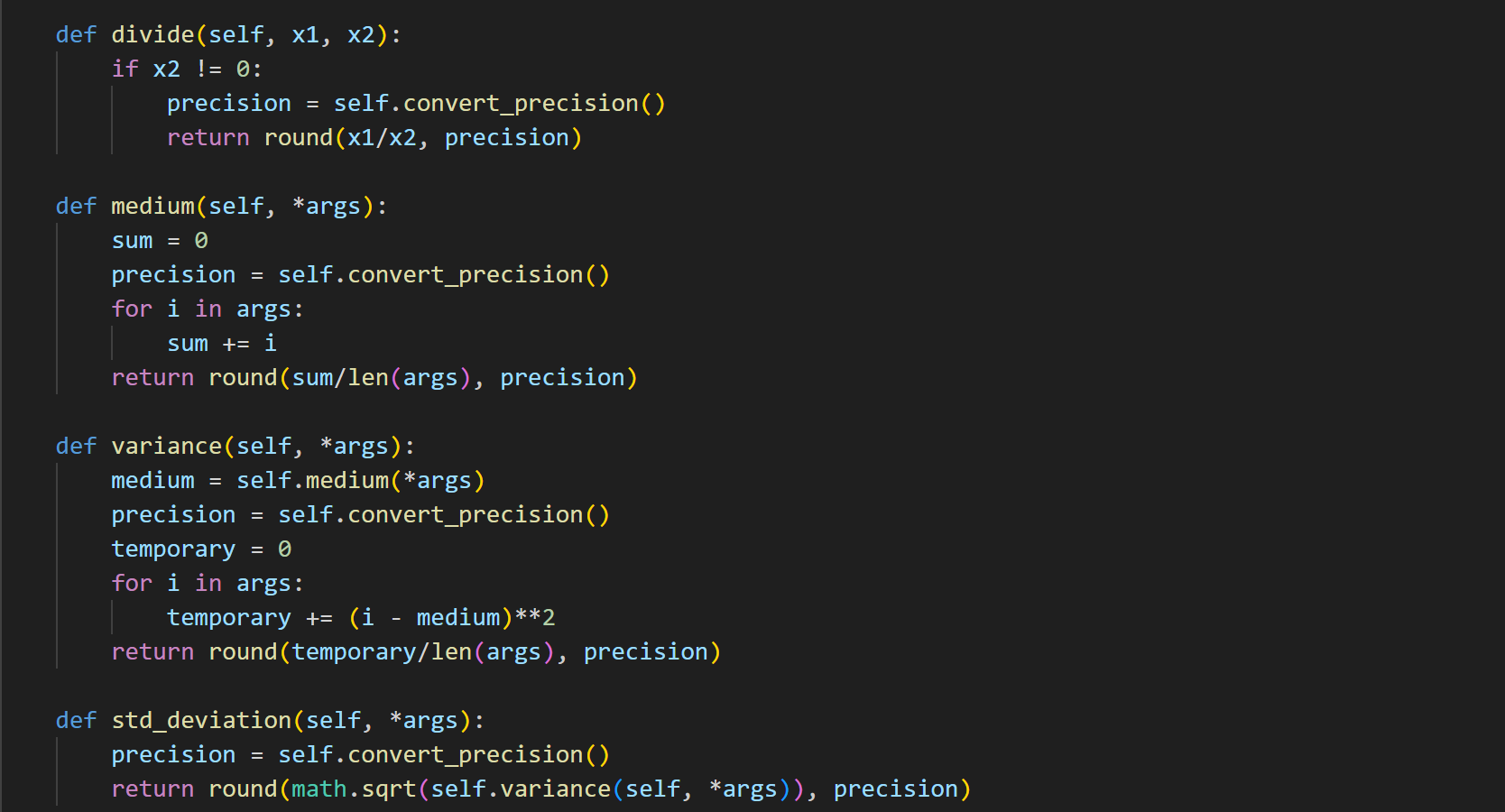
**Цель тестов:** Проверка корректности работы метода convert\_precision и округления в divide.

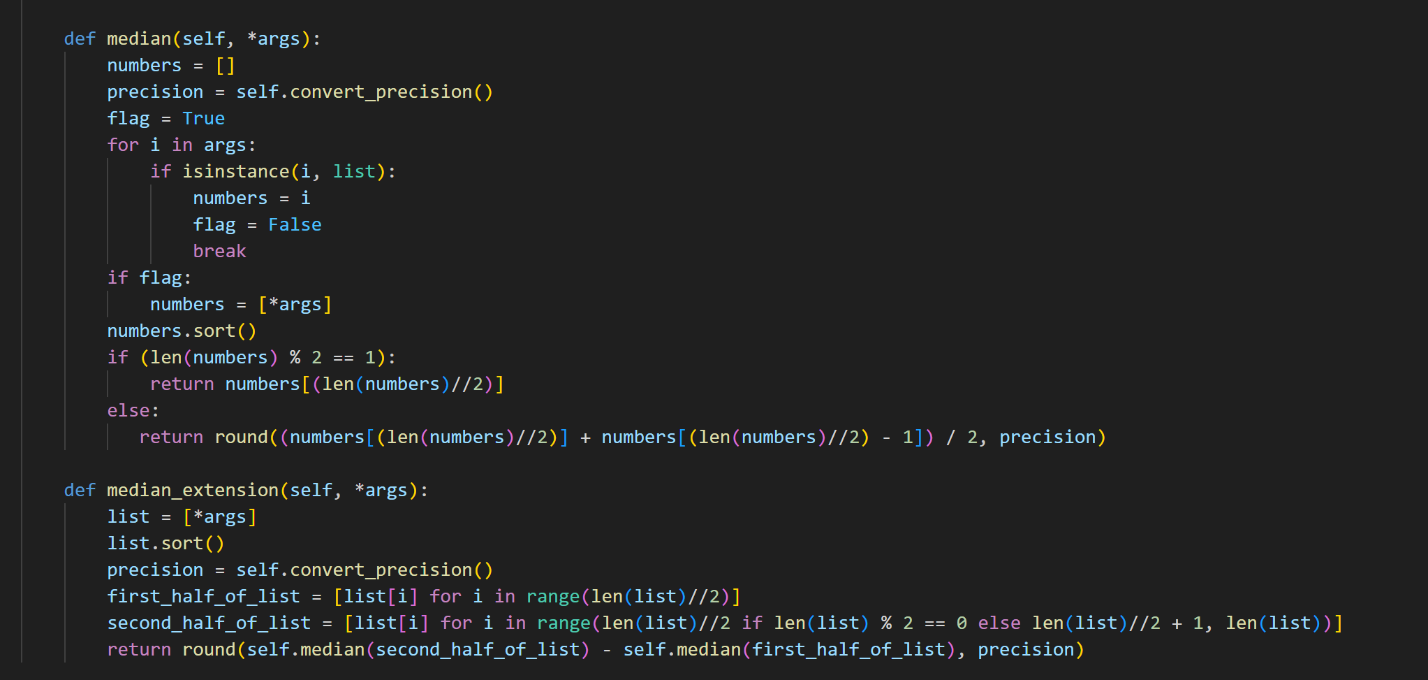
**Примеры тестов:**

1. **test\_convert()**
   * Проверяет, что для tolerance = 0.00001 метод convert\_precision возвращает 5.
2. **test\_divide()**
   * Проверяет, что divide(1, 3) с tolerance = 0.001 возвращает 0.333 (округление до 3 знаков).

**Задача 2.**

****

****

****

**1. Назначение программы**

Программа реализует расширенный класс Calculator для выполнения арифметических операций и статистических расчетов с контролем точности. Основные функции:

* Базовые операции (сложение, вычитание, умножение, деление).
* Статистические расчеты: среднее, дисперсия, стандартное отклонение, медиана, разница между медианами половин данных.
* Автоматическое округление результатов на основе параметра tolerance.

**2. Описание работы кода**

**2.1. Класс Calculator**

**Атрибуты:**

* tolerance (float): Параметр для определения точности округления (по умолчанию: 10^-6).

**Методы:**

1. **convert\_precision()**
   * Вычисляет количество знаков после запятой для округления.
   * Пример: tolerance = 0.001 → точность = 3.
2. **Базовые операции:**
   * add(\*args): Суммирует произвольное количество чисел.
   * multiply(\*args): Перемножает произвольное количество чисел.
   * subtract(x1, x2): Возвращает разность x1 - x2.
   * divide(x1, x2): Делит x1 на x2 с округлением (при x2 ≠ 0).
3. **Статистические методы:**
   * medium(\*args): Вычисляет среднее значение с округлением.
   * variance(\*args): Рассчитывает дисперсию.
   * std\_deviation(\*args): Возвращает стандартное отклонение.
   * median(\*args): Находит медиану данных (поддерживает вход в виде списка).
   * median\_extension(\*args): Вычисляет разницу между медианами двух половин данных.

**2.2. Особенности реализации**

* **Обработка произвольного числа аргументов** (\*args) в методах add, multiply, medium, variance, median.
* **Автоматическая сортировка данных** в методах median и median\_extension.
* **Проверка типа данных** в median: если передан список, он используется напрямую.