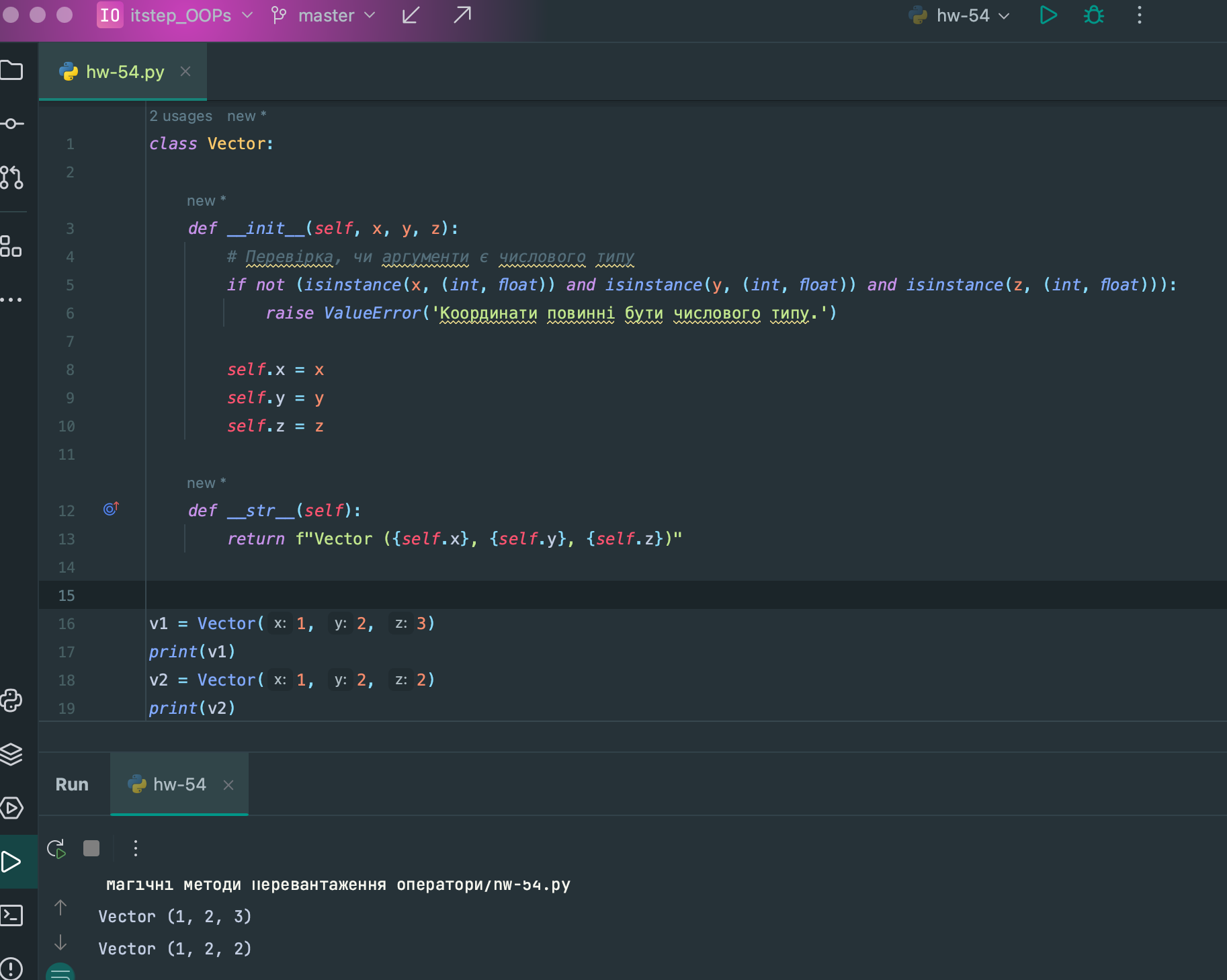
**Домашня робота №54 ООП. Магічні методи. Перевантаження операторів**

Реалізувати клас **Vector**, що представляє трьохвимірний вектор із координатами (x,y,z) у просторі.

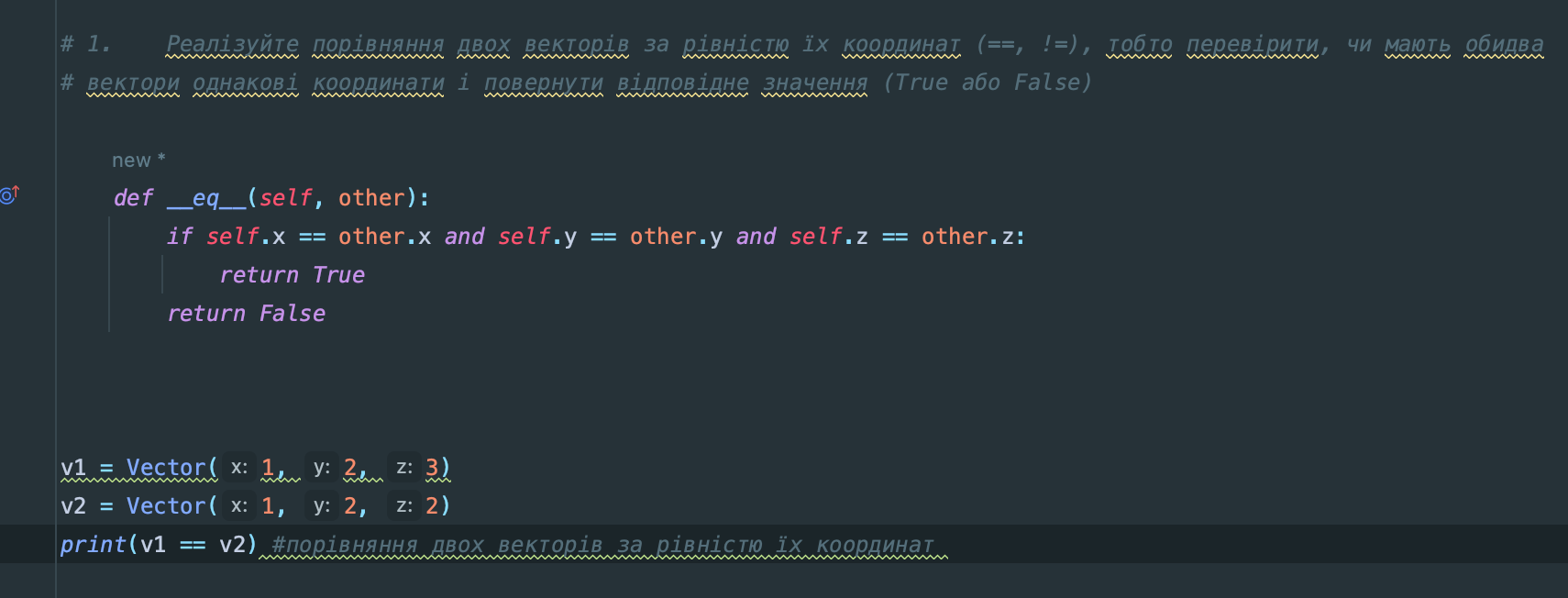
Виконати ініціалізацію методом \_\_init\_\_. В методі проконтролюйте значення аргументів на коректність (дані мають бути числового типу), використавши функцію isinstance(obj, Class).

Реалізувати \_\_str\_\_, що відображає екземпяр класу у вигляді рядкового літерала з подальшою можливістю виводу за допомогою функції print(). Наприклад, «Vector(1,2,3)».

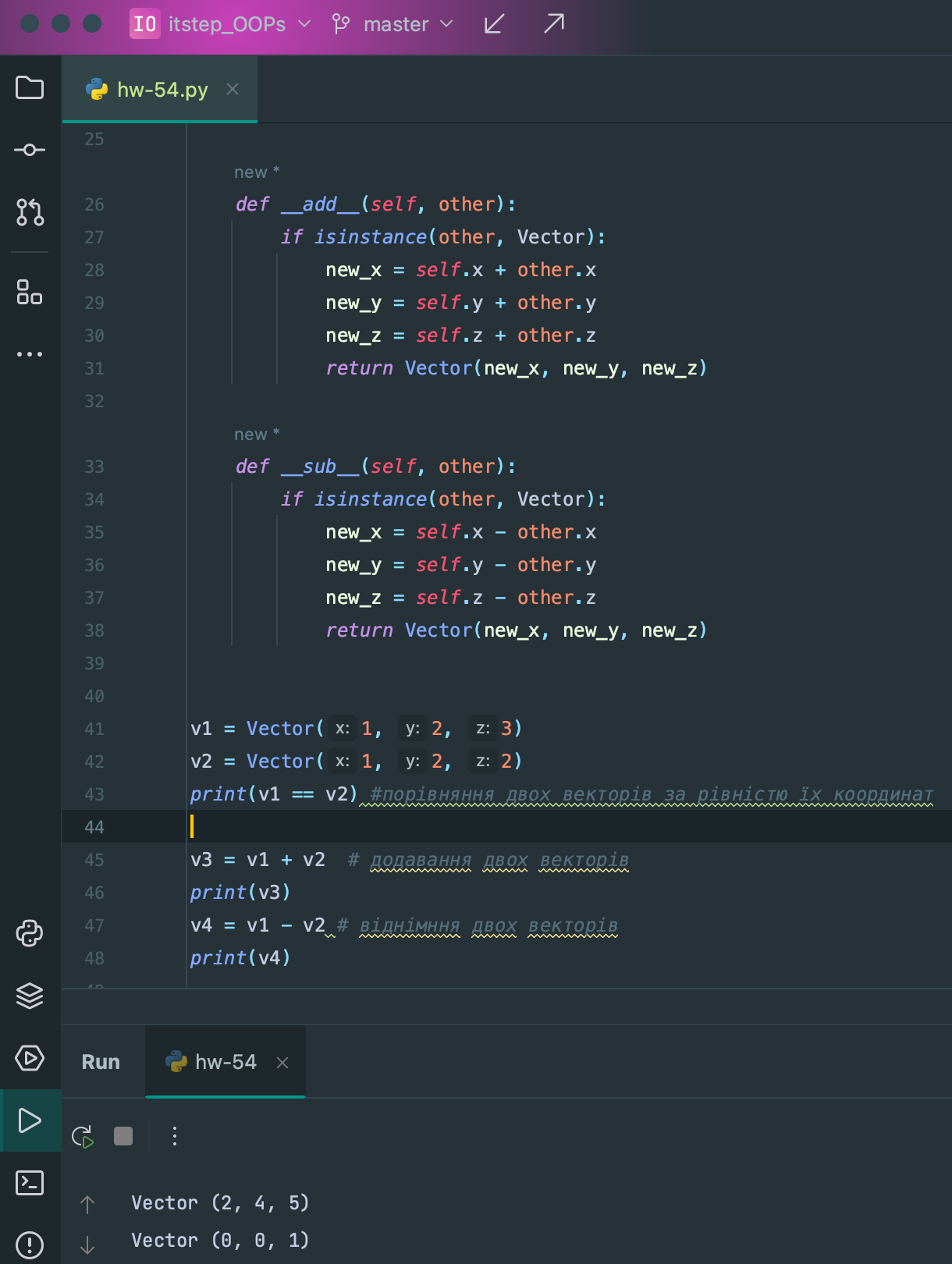


Реалізуйте наступні перевантажені оператори, щоб змінити та розширити його поведінку.

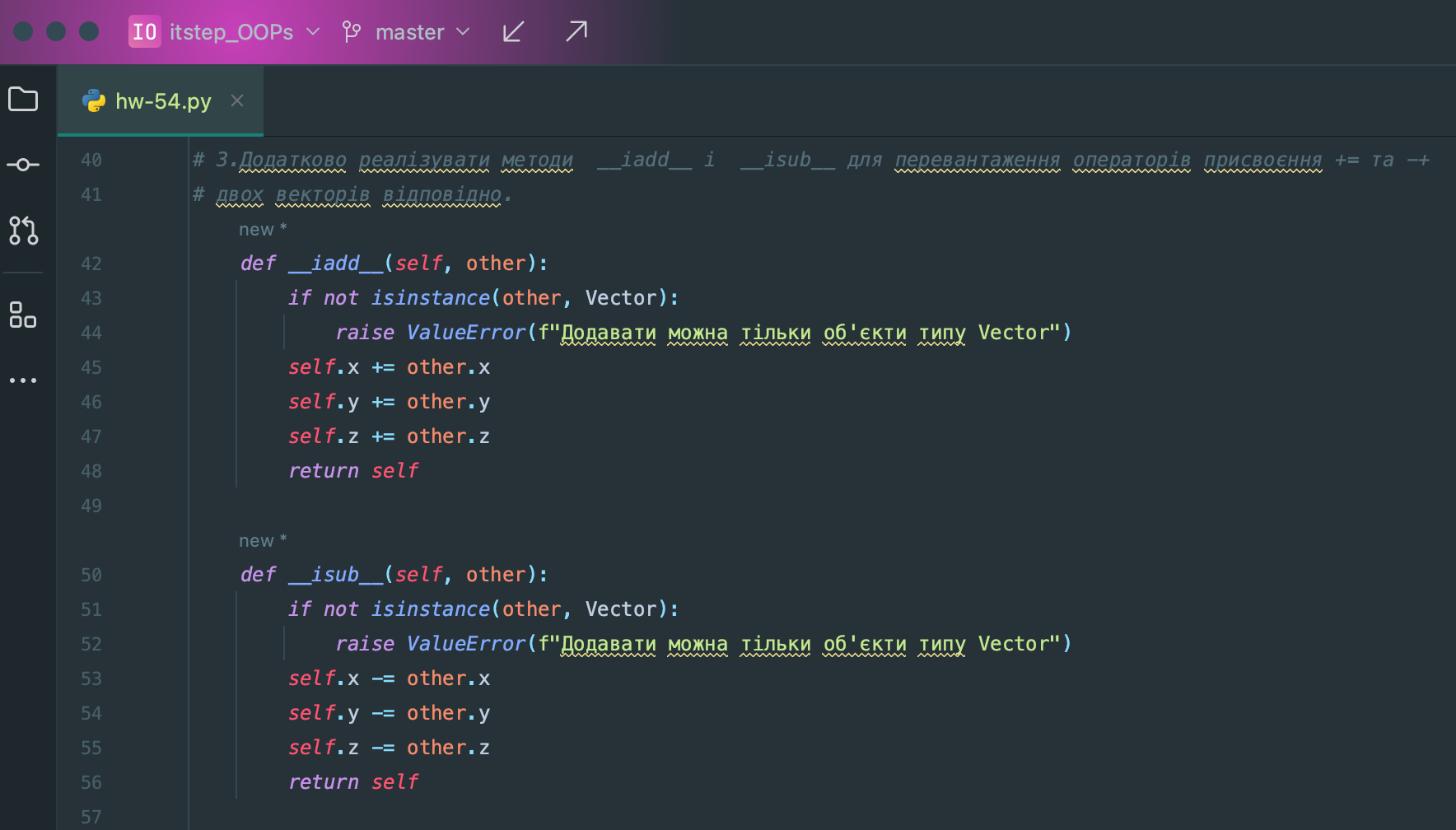
1. Реалізуйте порівняння двох векторів за рівністю їх координат (==, !=), тобто перевірити, чи мають обидва вектори однакові координати і повернути відповідне значення (True або False)



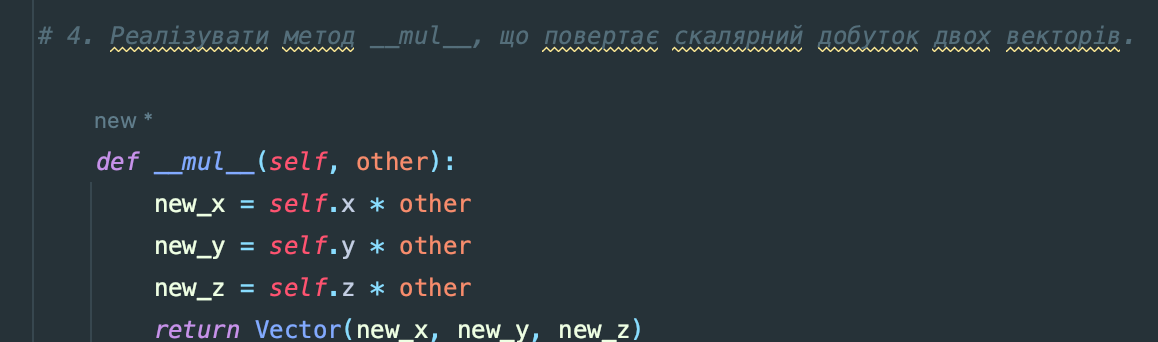
1. Реалізувати методи \_\_add\_\_ і \_\_sub\_\_ для додавання і віднімання двох векторів відповідно (повертає новий вектор з координатами).



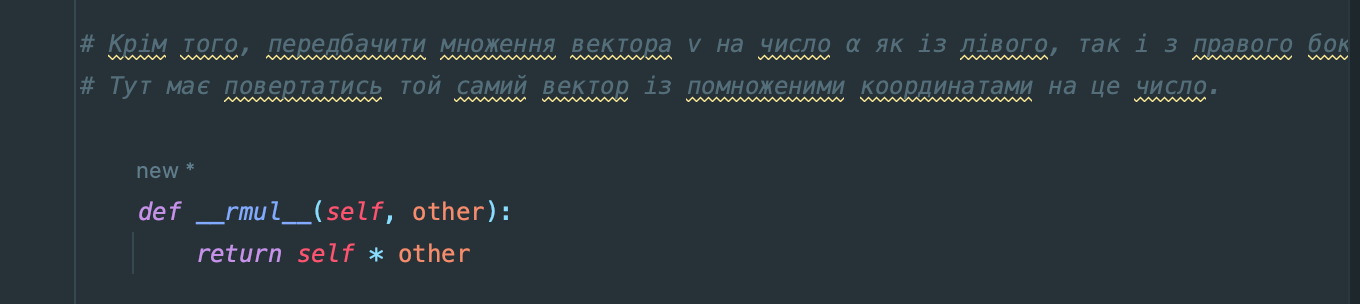
1. Додатково реалізувати методи \_\_iadd\_\_ і \_\_isub\_\_ для перевантаження операторів присвоєння += та -+ двох векторів відповідно.

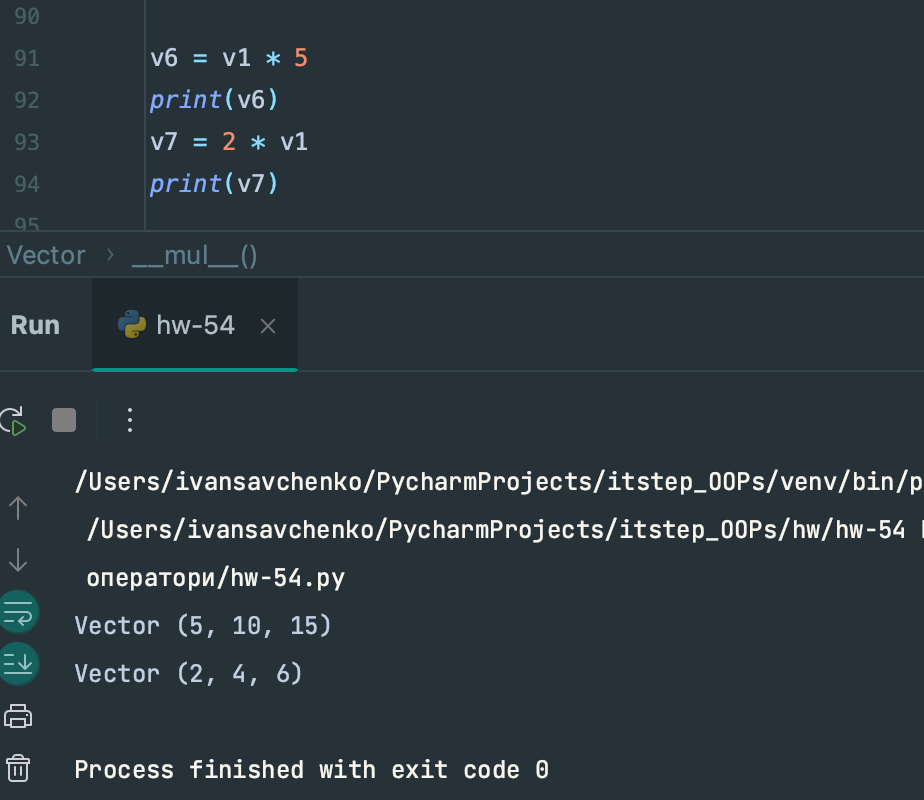


1. Реалізувати метод \_\_mul\_\_, що повертає скалярний добуток двох векторів.

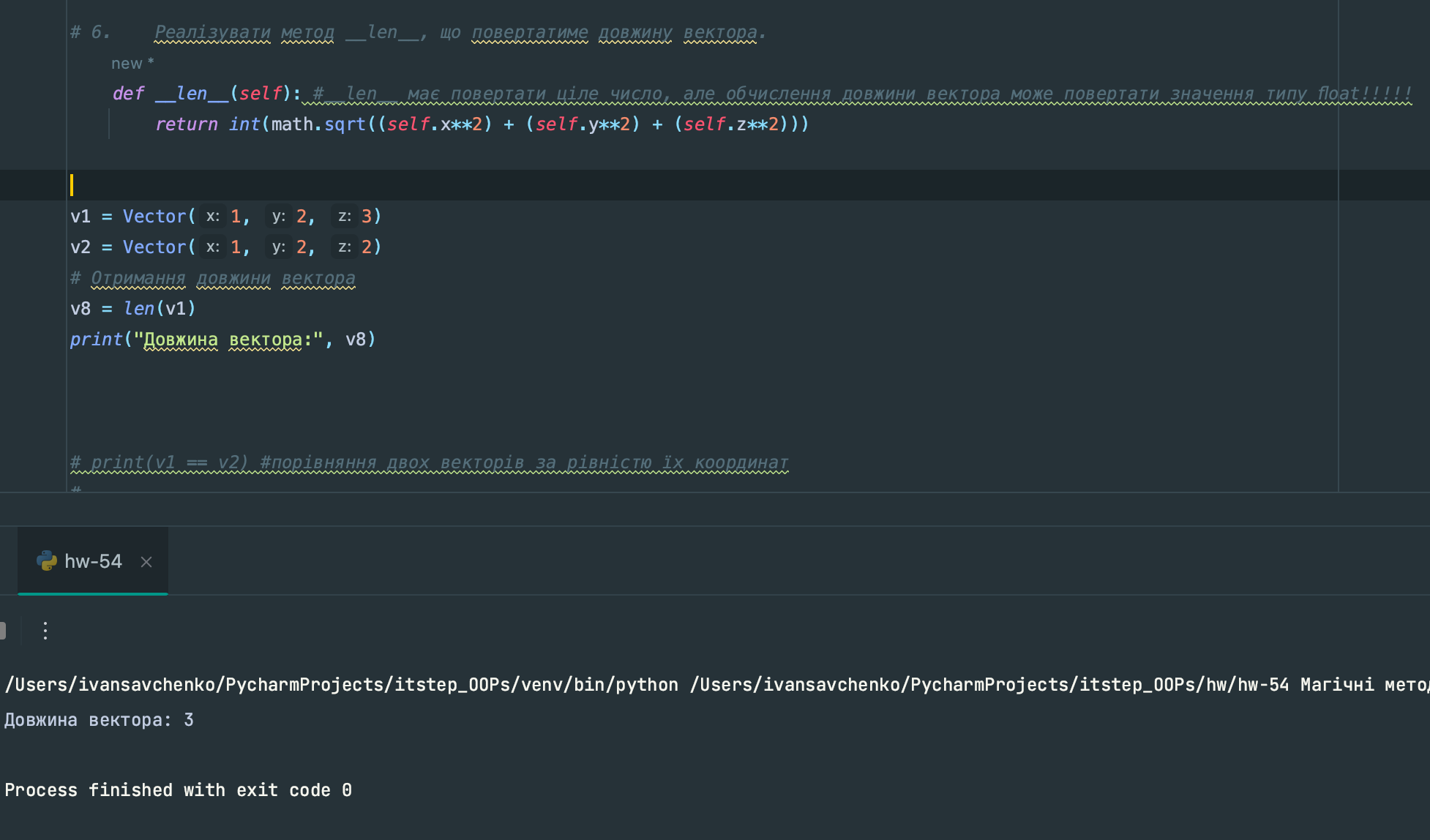


1. Крім того, передбачити множення вектора **v** на число як із лівого, так і з правого боку, тобто **v**\* та \***v.** Тут має повертатись той самий вектор із помноженими координатами на це число.

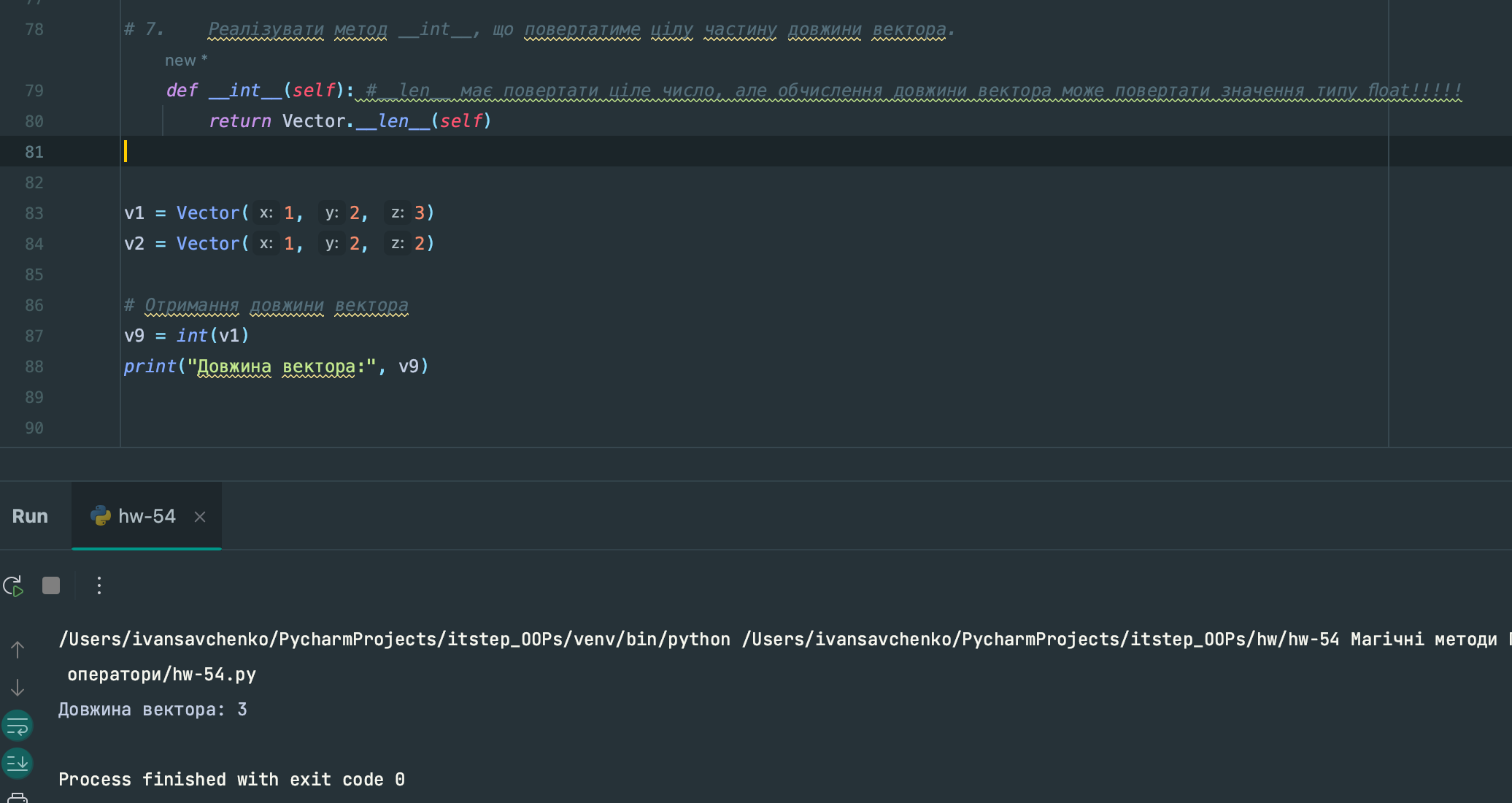
****

****

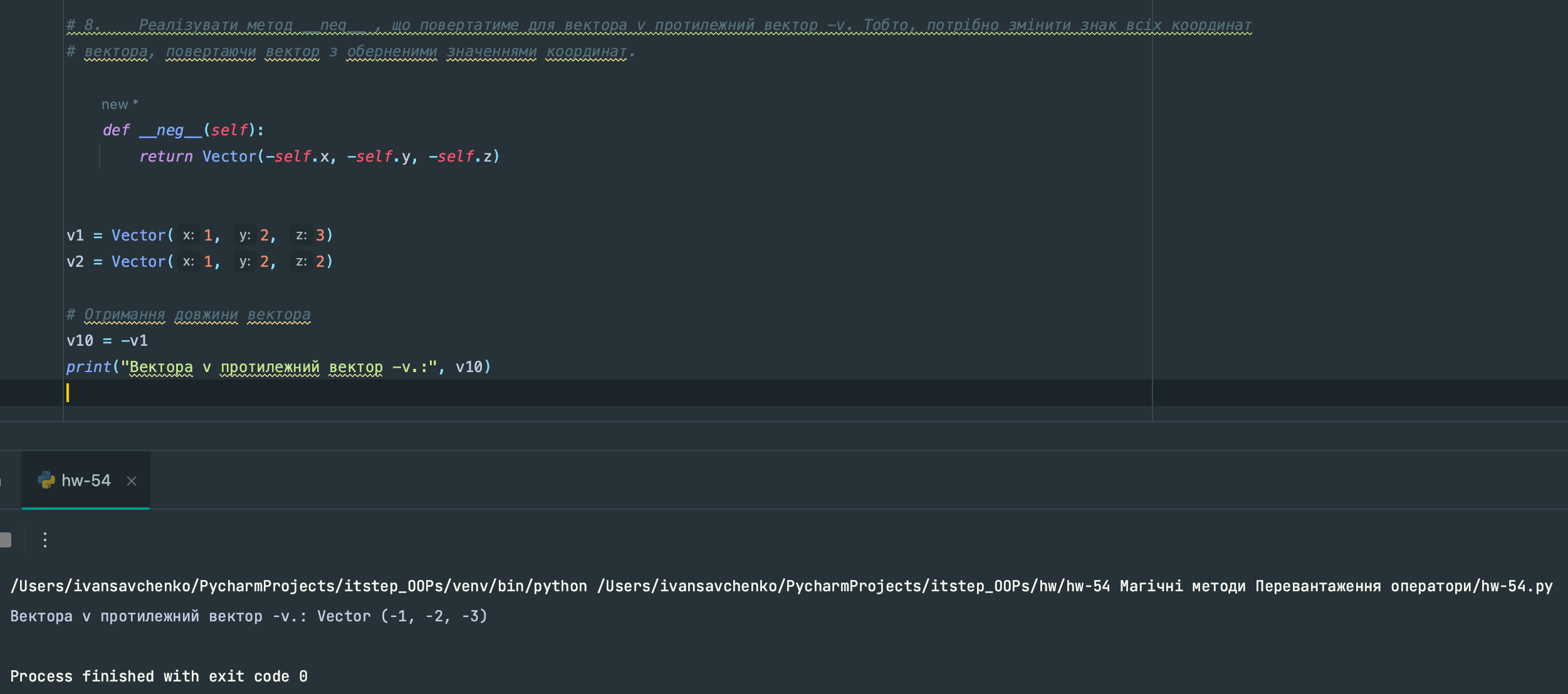
1. Реалізувати метод \_\_len\_\_, що повертатиме довжину вектора.



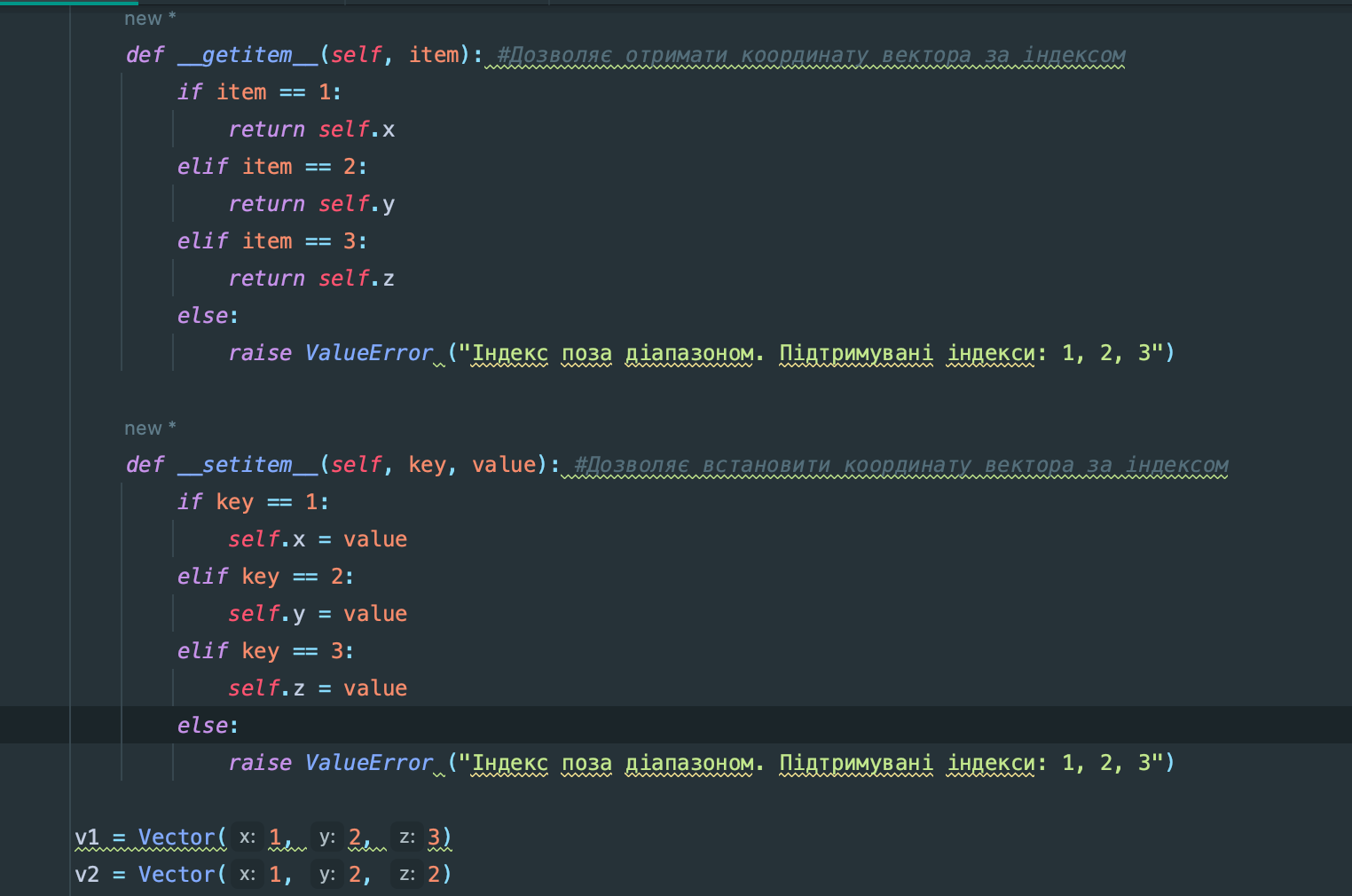
1. Реалізувати метод \_\_int\_\_, що повертатиме цілу частину довжини вектора.

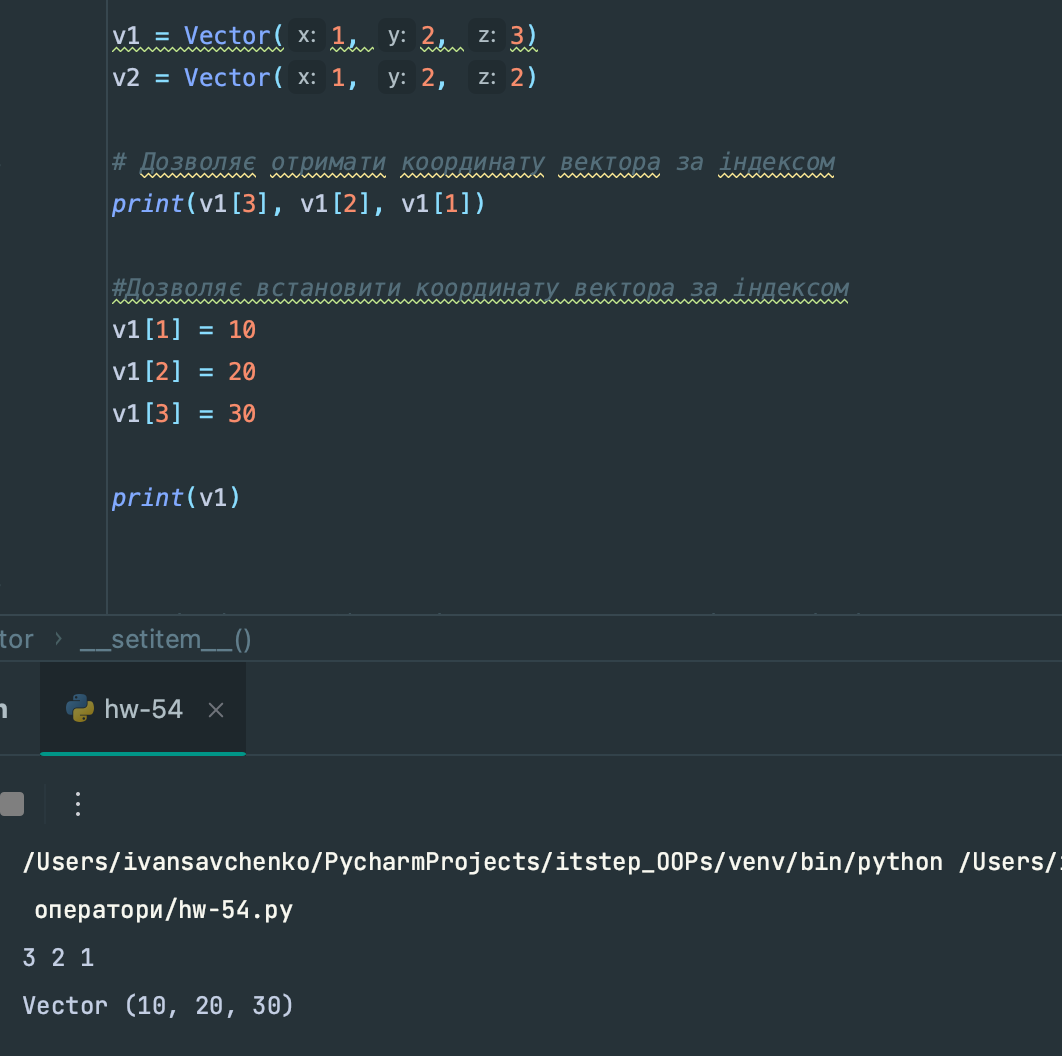


1. Реалізувати метод \_\_neq\_\_, що повертатиме для вектора **v** протилежний вектор **-v**. Тобто, потрібно змінити знак всіх координат вектора, повертаючи вектор з оберненими значеннями координат.

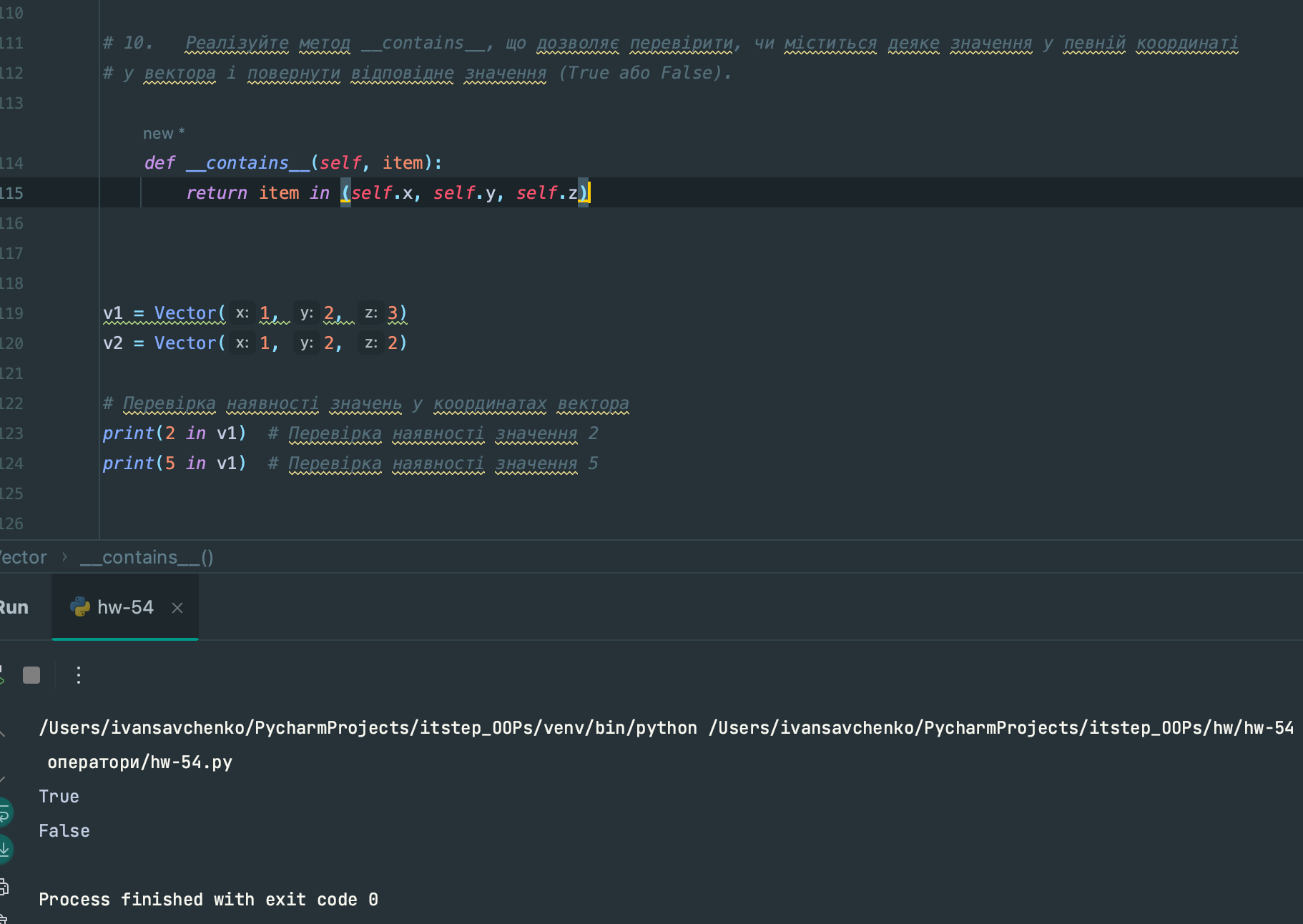


1. Реалізувати методи \_\_getitem\_\_ і \_\_setitem\_\_, що дозволяють достукатись до координат вектора за допомогою індексів. Наприклад, ви можете використовувати v[1] для доступу до першої координати вектора, v[2] - до другої, v[3] - до третьої.

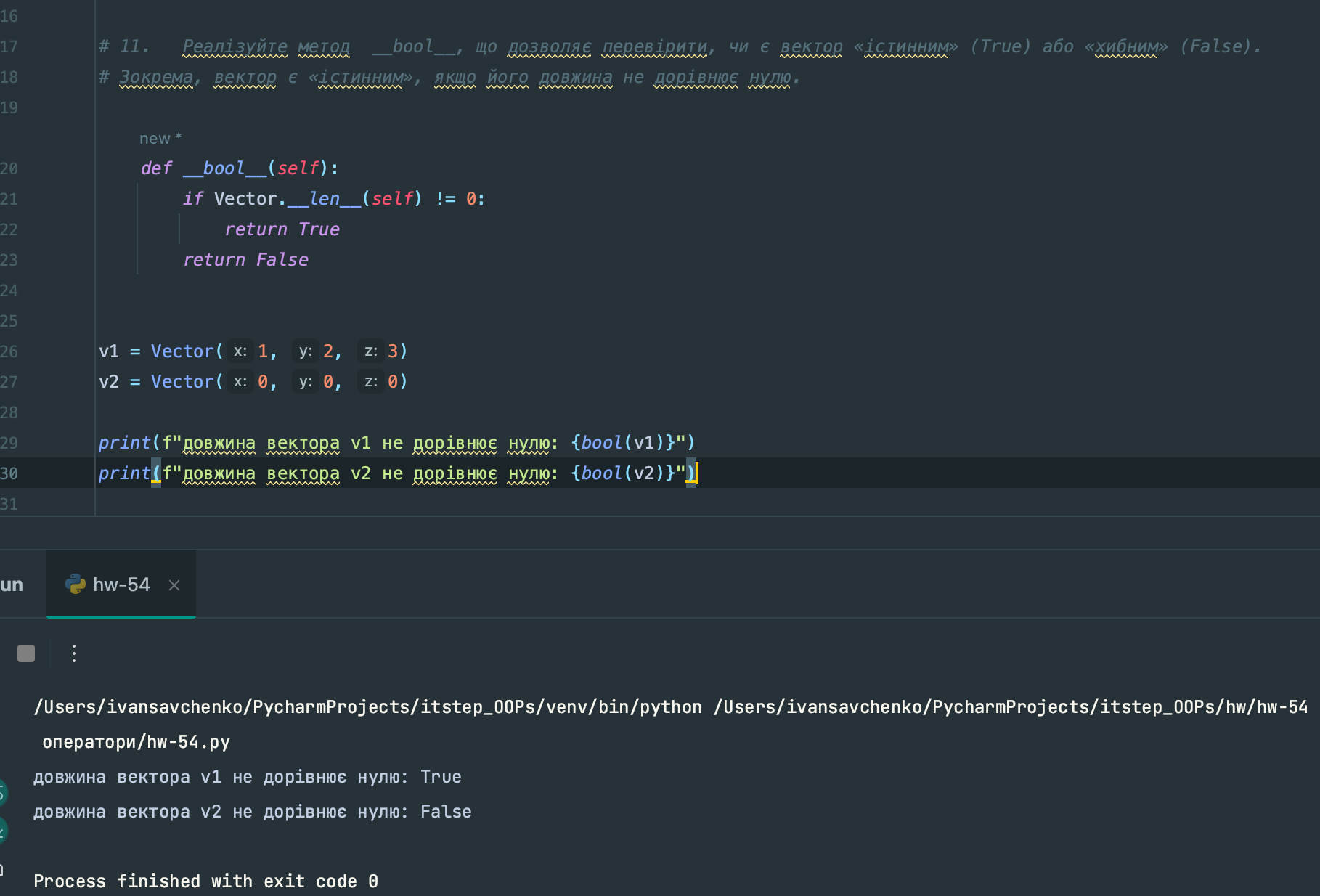


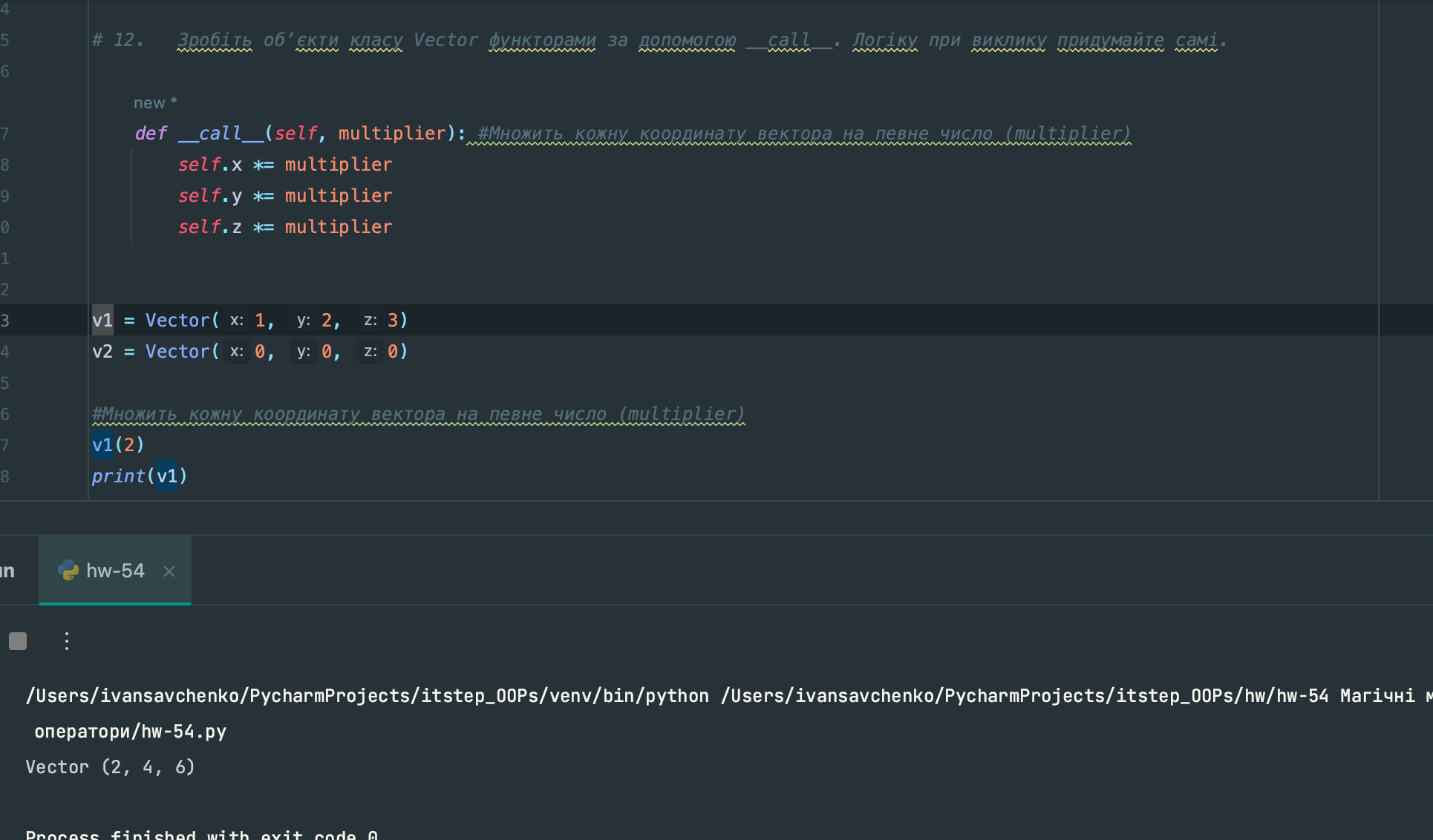


1. Реалізуйте метод \_\_contains\_\_, що дозволяє перевірити, чи міститься деяке значення у певній координаті у вектора і повернути відповідне значення (True або False).



1. Реалізуйте метод \_\_bool\_\_, що дозволяє перевірити, чи є вектор «істинним» (True) або «хибним» (False). Зокрема, вектор є «істинним», якщо його довжина не дорівнює нулю.



1. Зробіть об’єкти класу Vector функторами за допомогою \_\_сall\_\_. Логіку при виклику придумайте самі.
2. 

Протестуйте функціональність класу і роботу магічних методів і операторів.