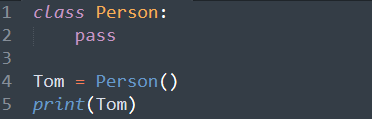
**Домашня робота №49 «Вступ до ООП. Класи»**

**Хід виконання роботи**

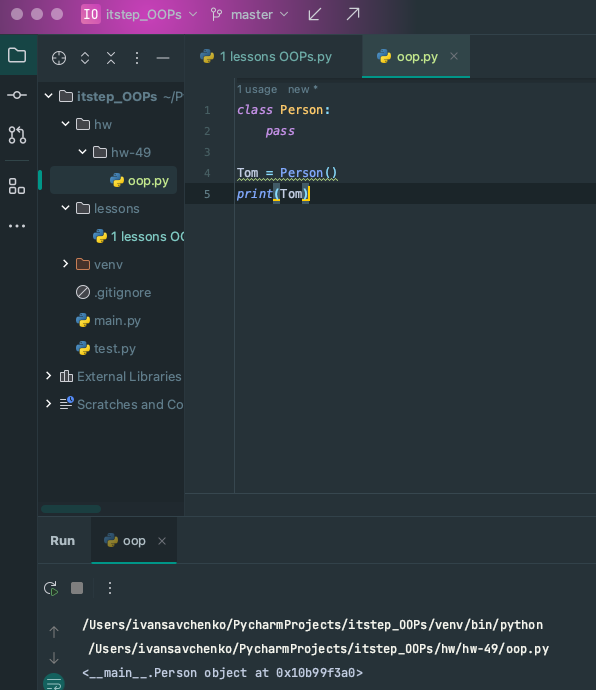
**ЧАСТИНА 1 (виконання оцінюється у 10 балів)**

1. Створіть новий модуль (файл) з назвою oop.py, а в ньому задайте пустий клас та створіть екземпляр Tom цього класу .

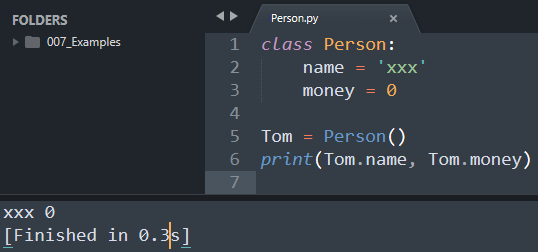


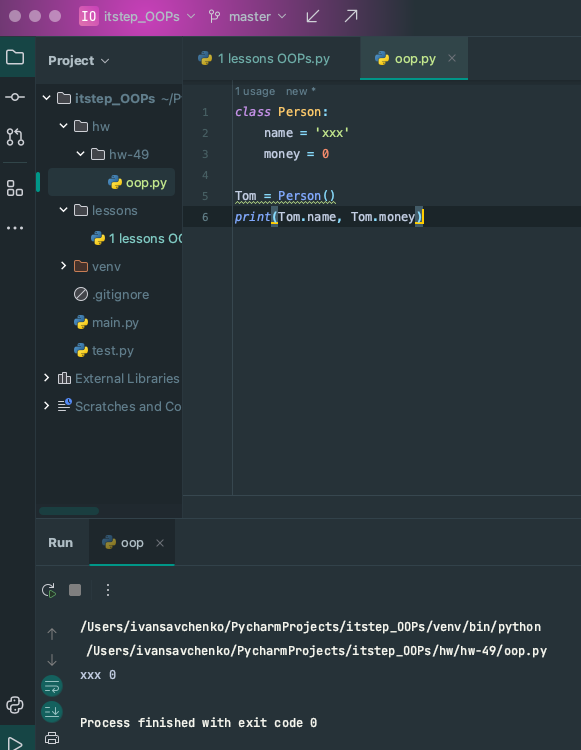
Виконайте даний код. Маєте приблизно отримати таке:



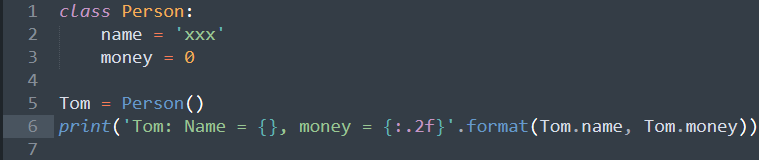


1. Додайте атрибути (поля або властивості) до нашого класу, відредагувавши попередній код:





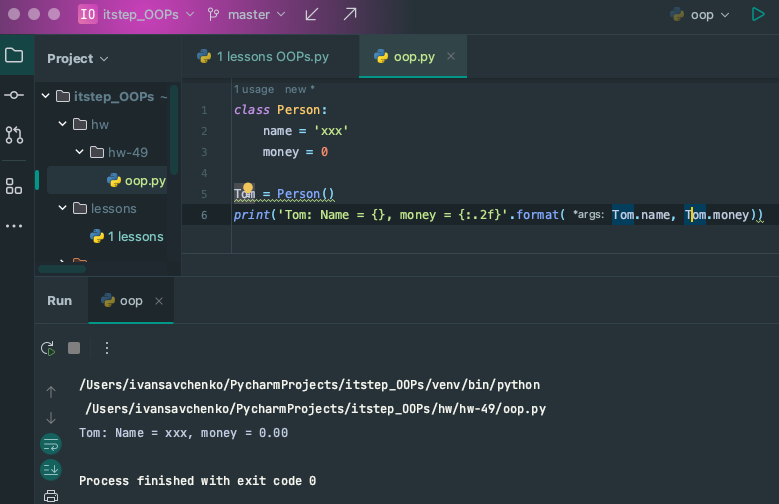
1. Зробимо покращення виводу атрибутів. Для команди print скористаємось функцією format:



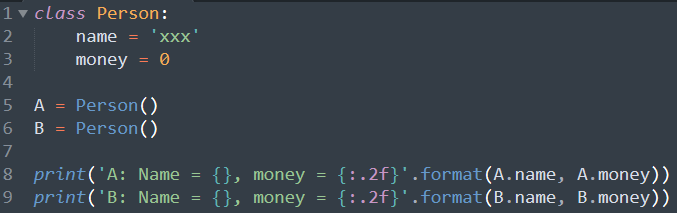
Результат:



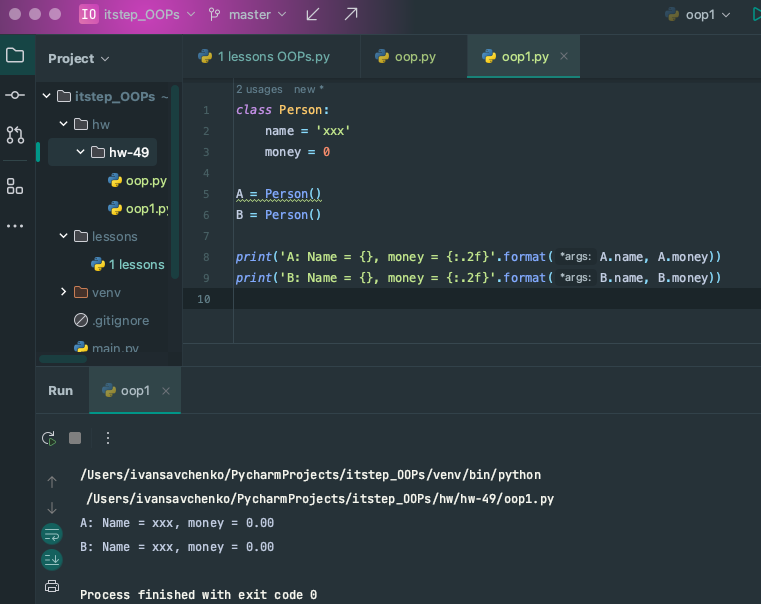
***Зауваження.*** Далі з*амість функції format можете використовувати f-рядок*



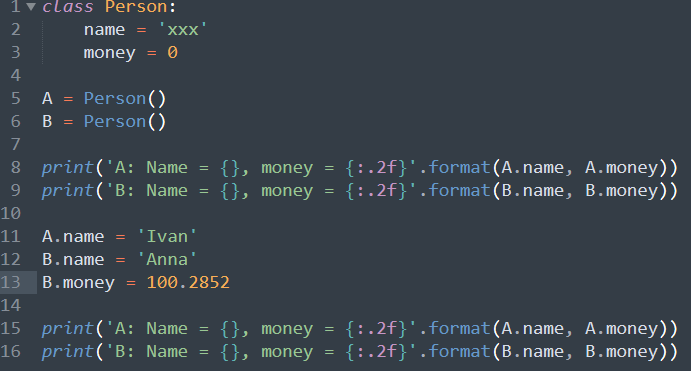
1. Відредагуємо далі код, створивши два екземпляра A і B класу Person:



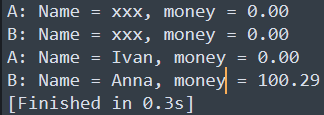
Як бачимо “людини” А і B мають однакове ім’я - xxx.

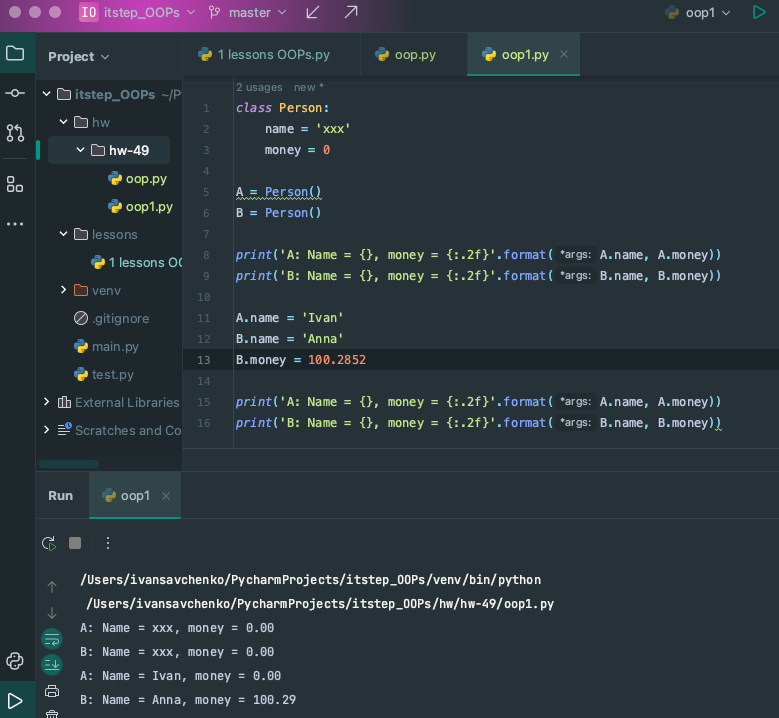


Виправимо це, задавши значення атрибуту name відповідно (рядки 11 і 12). Також задамо кількість грошей для Anna.



Результат:

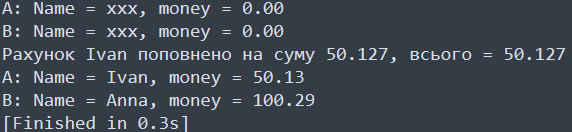


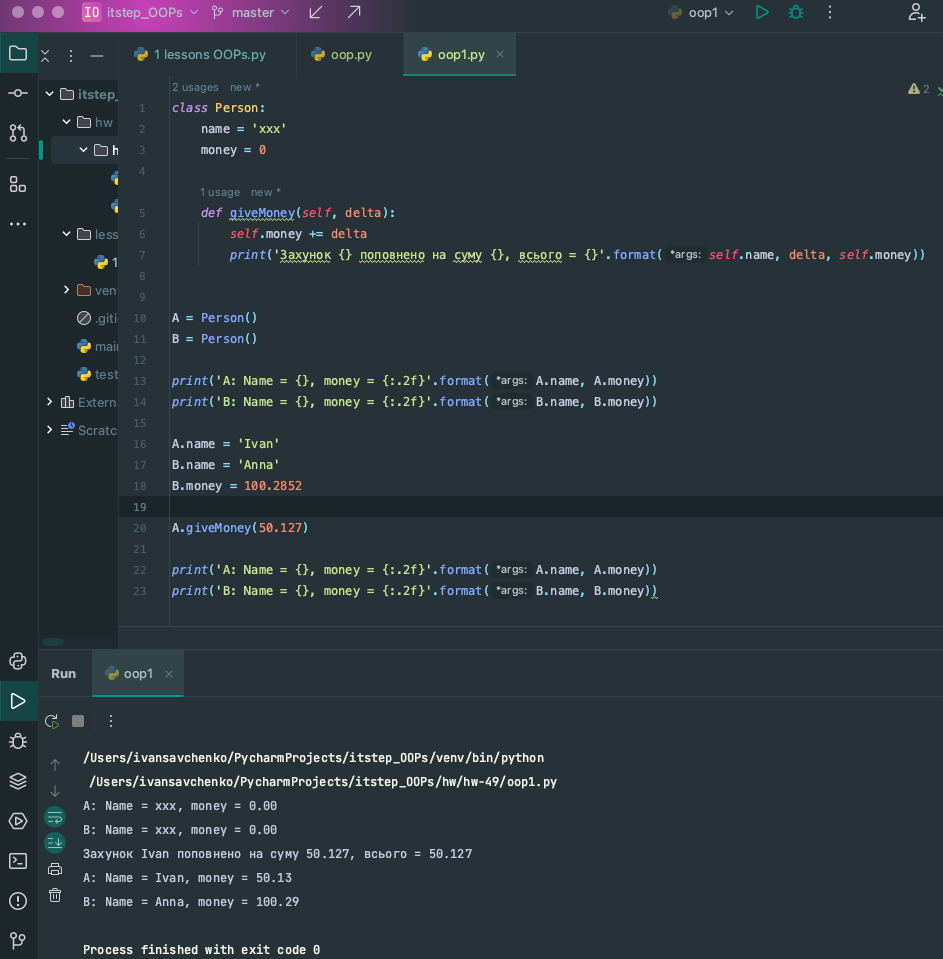


1. Реалізуємо метод giveMoney (дивись нижче рядки 5-8), який збільшує кількість грошей на рахунку для деякого екземпляру класу Person. У рядку 20 викличимо цей метод для екземпляру A.

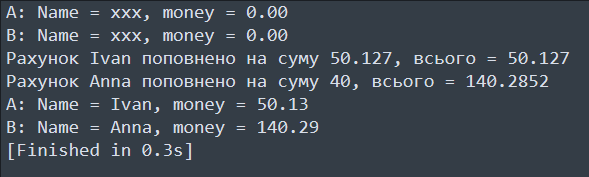


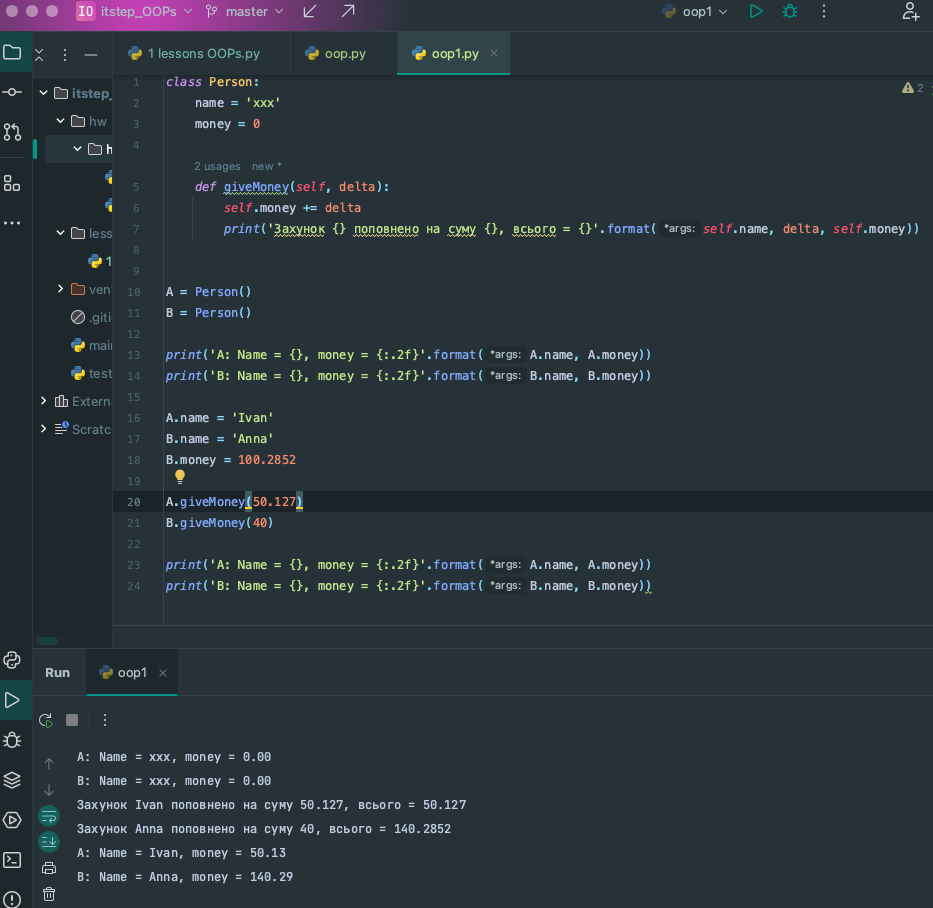
Результат:



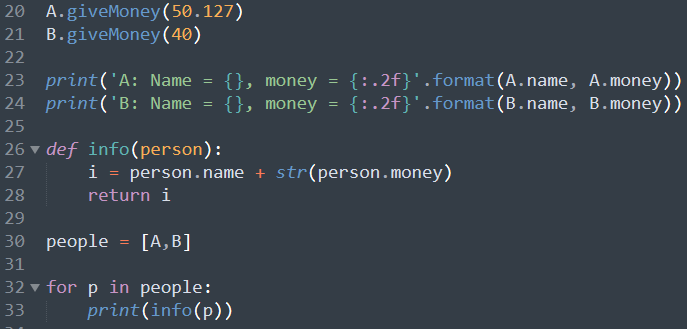
****

У рядку 21 кодом зробіть поповнення рахунку для Анни на 40. Рядок 22 і 23 перенести вниз. Маєте отримати такий результат:

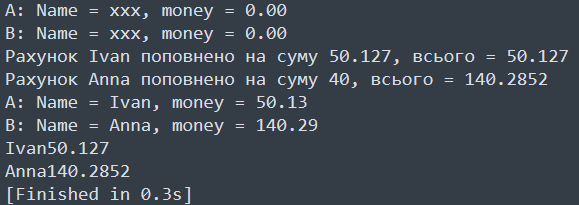


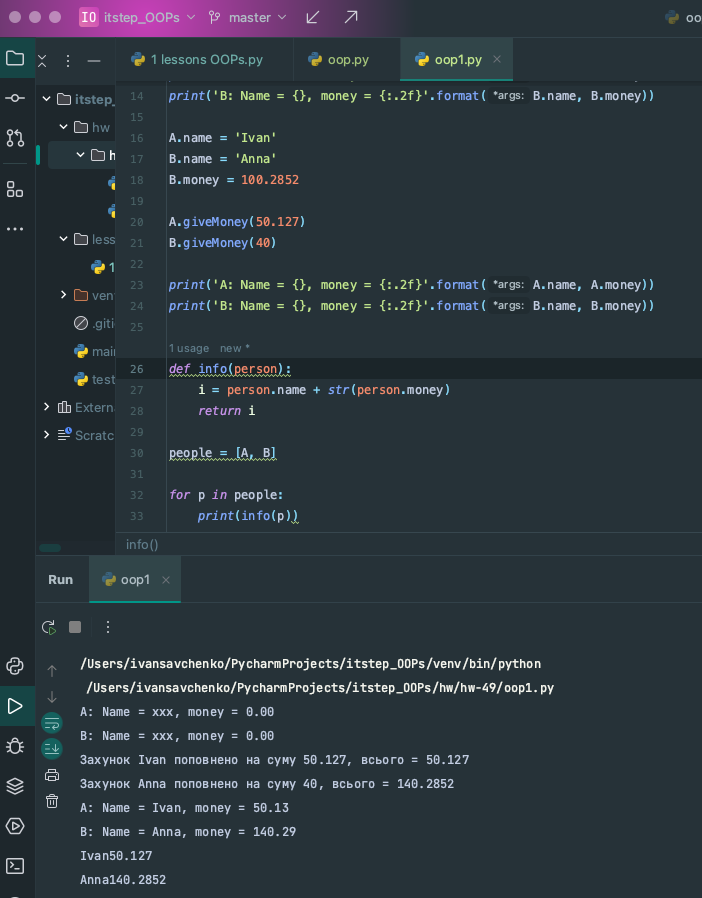


1. Реалізуємо функцію info, яка буде приймати параметром об’єкт person класу Person і повертати дані цього об’єкта у вигляді рядка.

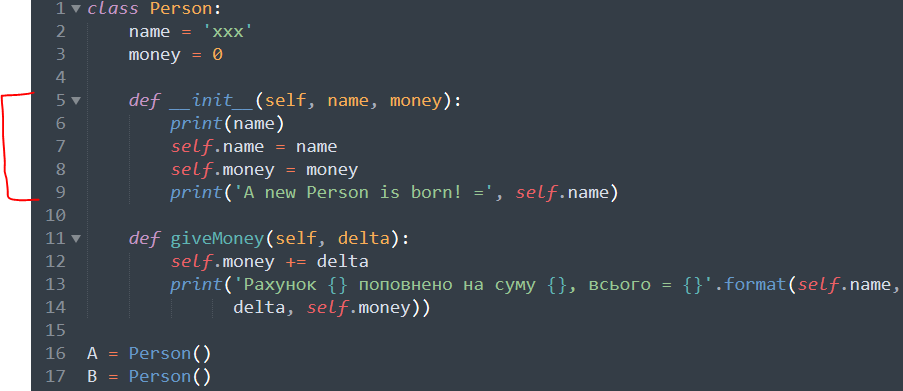


Отримаємо результат:

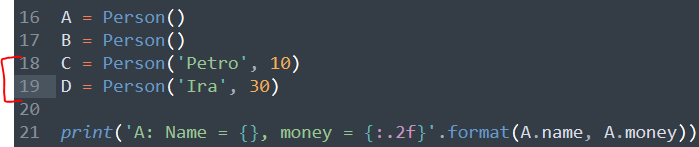




1. Починаючи з рядка 5, реалізуємо конструктор для нашого класу.



Створимо 2 екземпляра у рядку 18 і 19:



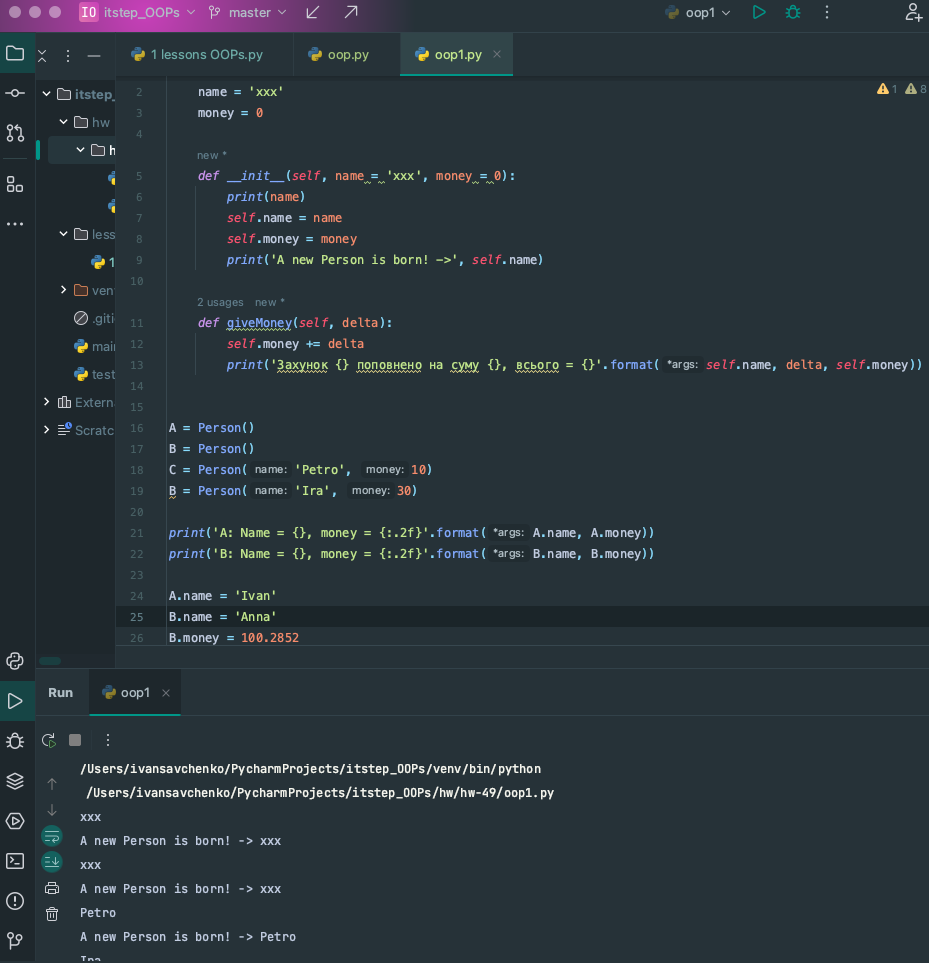
Спробуйте зараз програму. Однак виникне помилка



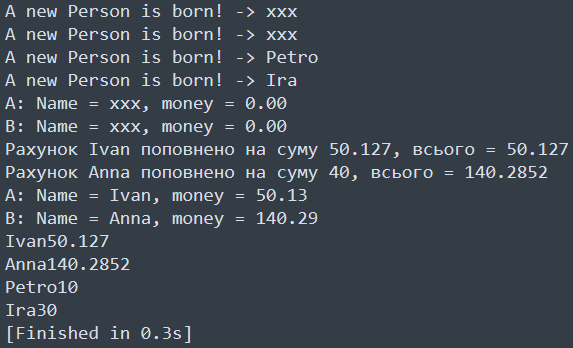
Виправимо це шляхом задання параметрів за замовчуванням.

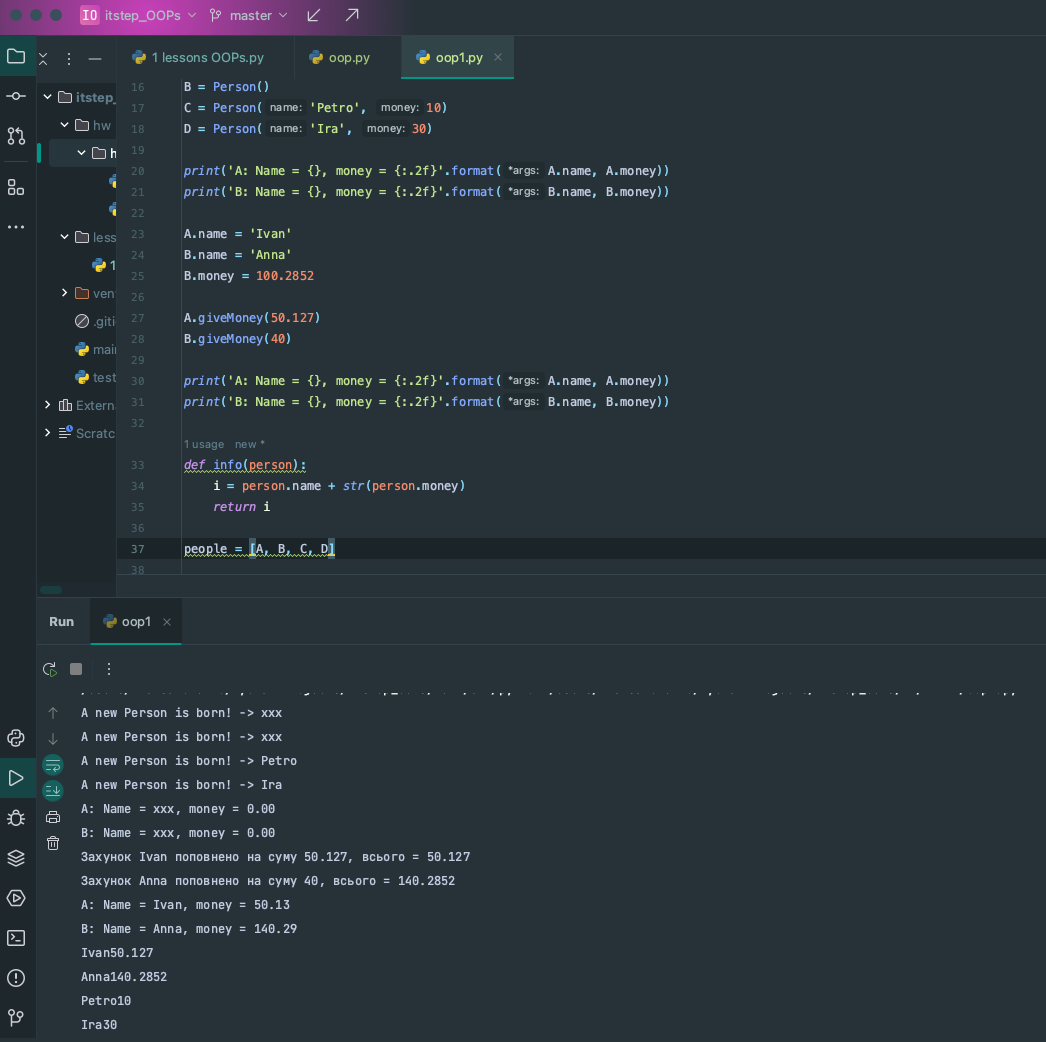


Запустіть код. Перегляньте результат.

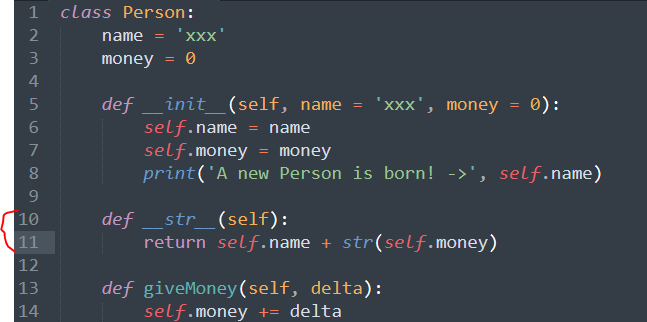


Видаліть рядок 6 (команду print(name)), а у масив people додайте ініціалізованих Петра і Іру і ще раз запустіть код. Маєте отримати наступне:

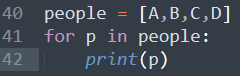




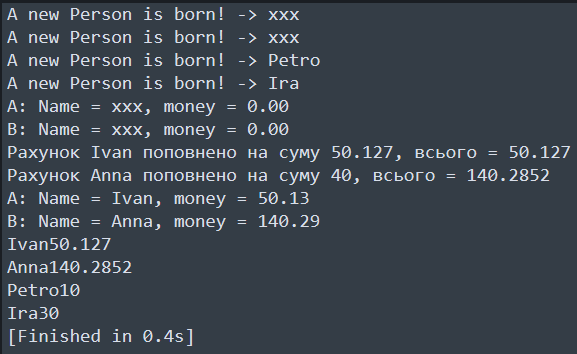
1. Зробимо так, щоб наші об’єкти класу Person можна було друкувати функцією print. Для цього реалізуємо магічний метод \_\_str\_\_ у рядках 10 і 11

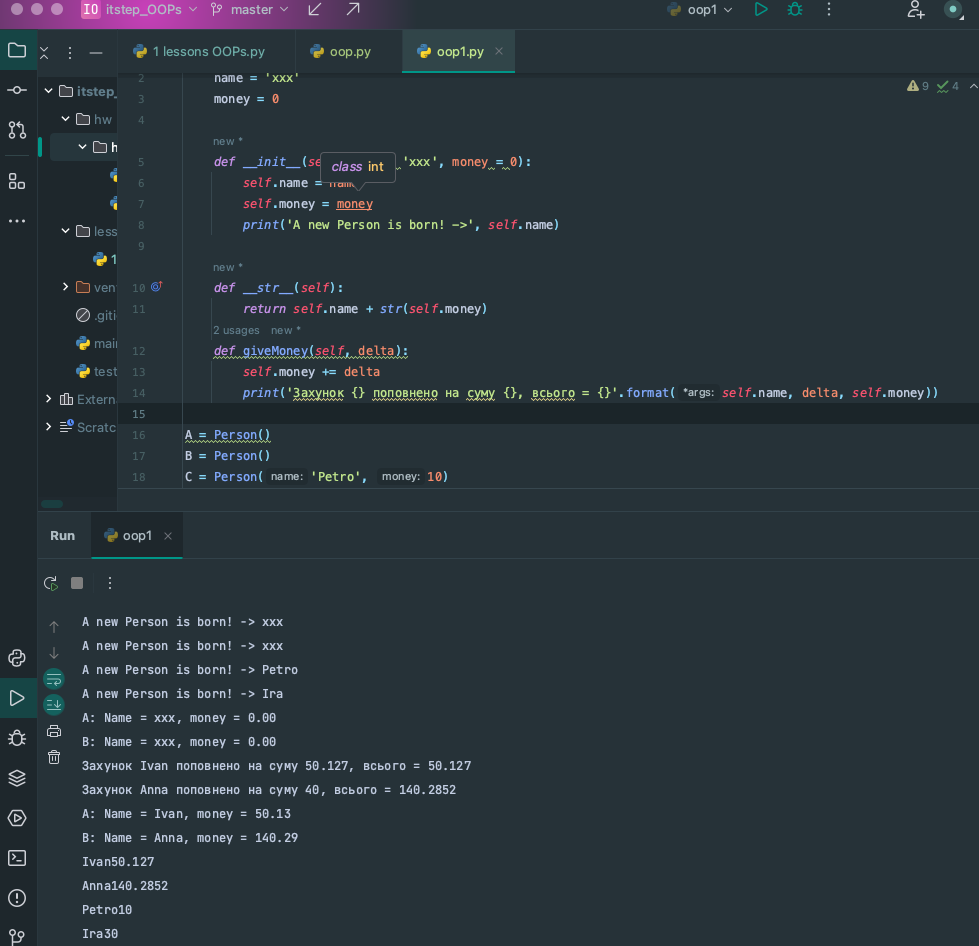


У циклі зробимо правки

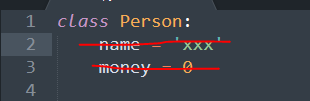


Отримаємо те саме:

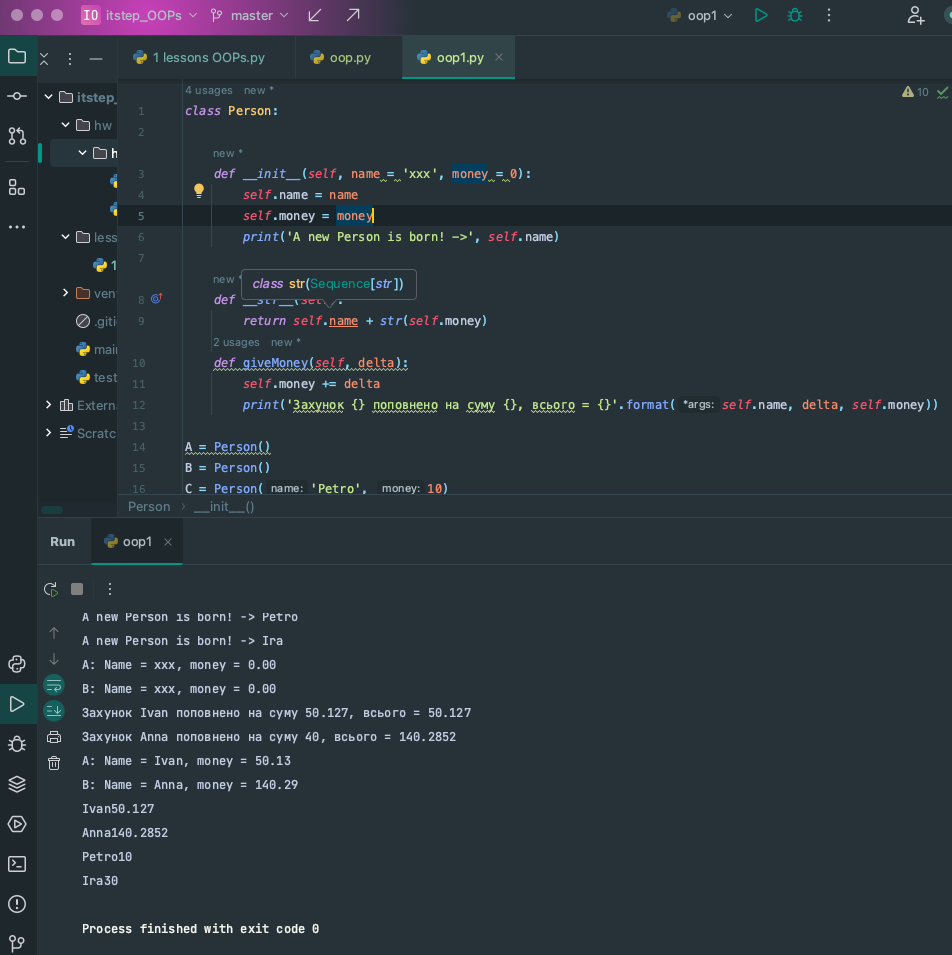




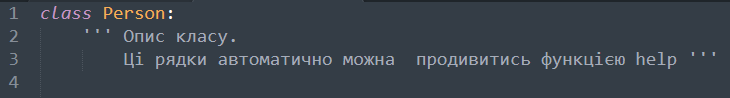
1. Видаліть рядки (це поля класу, які спільні для всіх об’єктів)



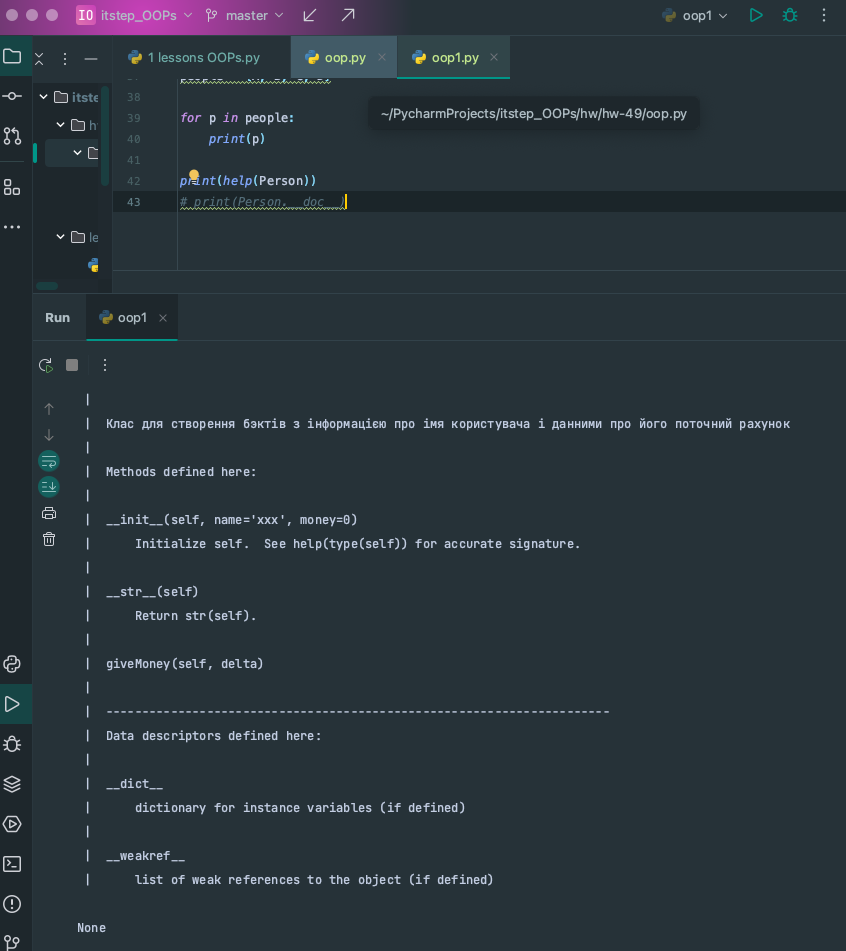
і переконайтесь у працездатності програми.

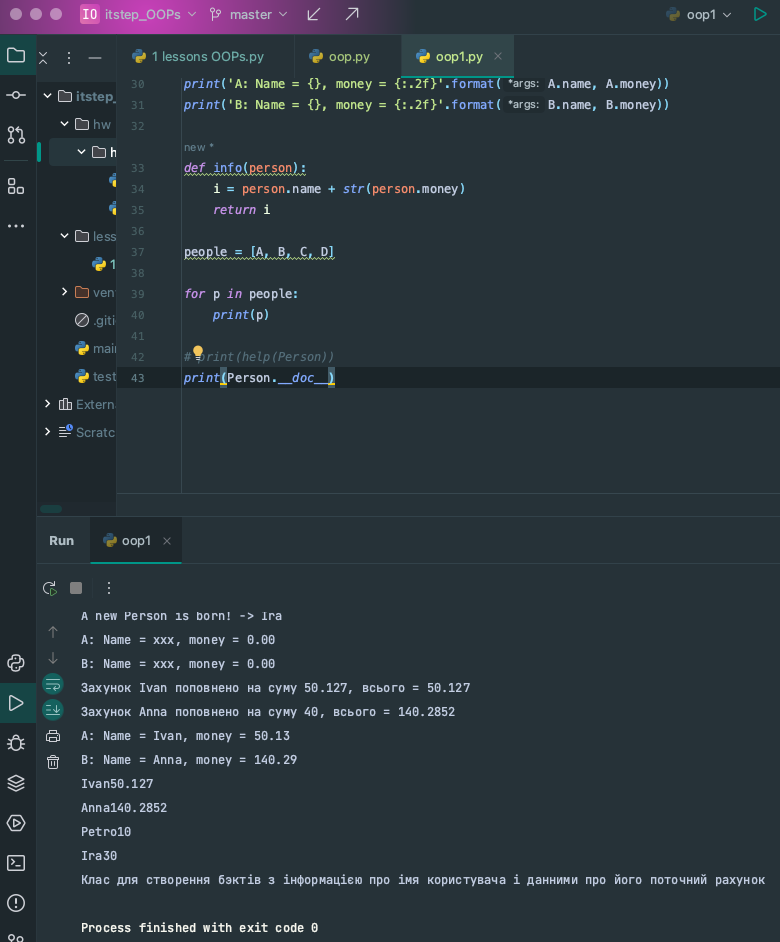


1. Добавте коментарій для класу



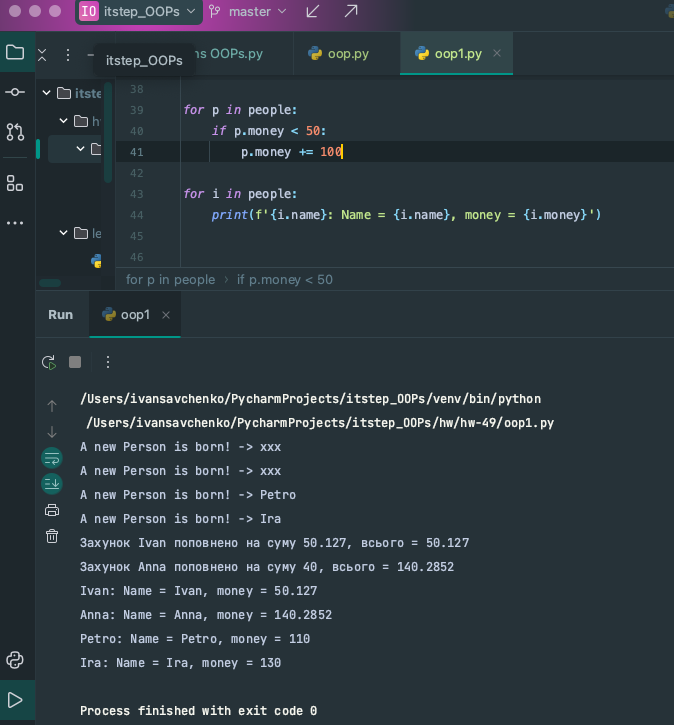
Викликати документацію можна так: help(Person) або Person.\_\_doc\_\_





**ЧАСТИНА 2 (самостійно, оцінюється у 2 бали)**

1. Усім тим, у кого на рахунку менше 50, зробіть поповнення рахунку на 100. Виведіть інформацію про стан рахунку кожного.



1. Для класу Person додайте атрибут об’єкта у вигляді списку, який буде містити знайомих. При ініціалізації об’єкту – цей список порожній. Далі реалізуйте два методи:

* know(person), який дозволяє додати іншу людину як об’єкт в список знайомих
* is\_known(person), який повертає чи знайомі дві людини.

