Звіт

З лабораторної роботи №3

Студента групи МІТ-31

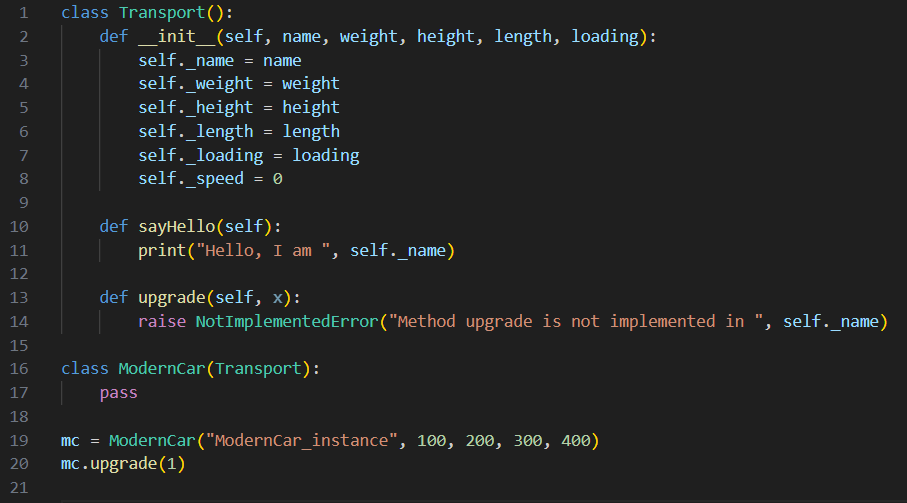
Добровольського Арсенія Михайловича

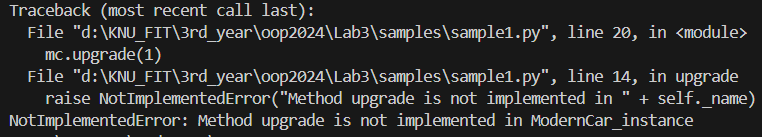
Варіант №9

**Тема роботи:** Абстрактні класи

**Мета роботи:** ознайомитися з поняттям абстрактного класу, навчитися створювати абстрактні класи у Python.

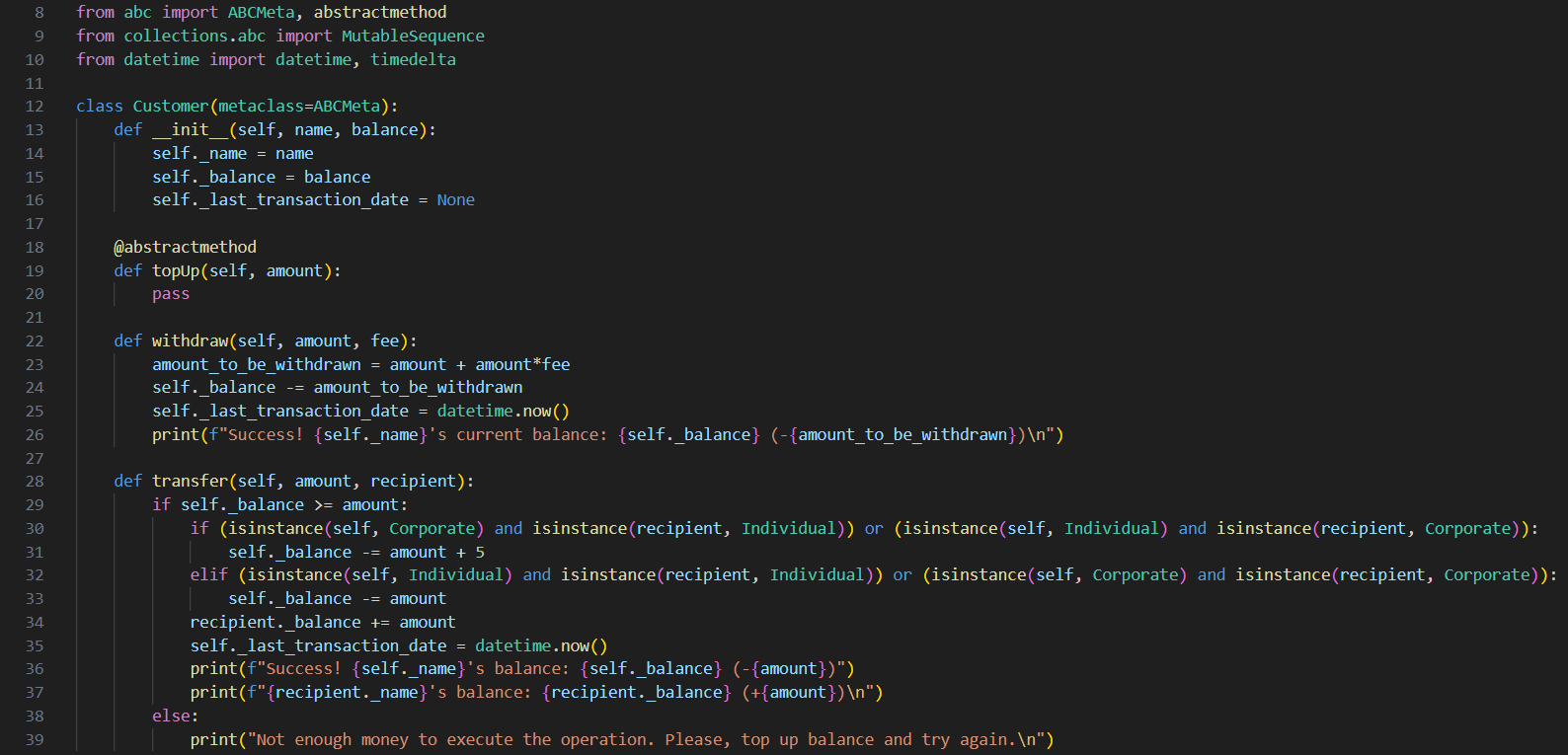
Програма-зразок, наведена у лабораторній роботі



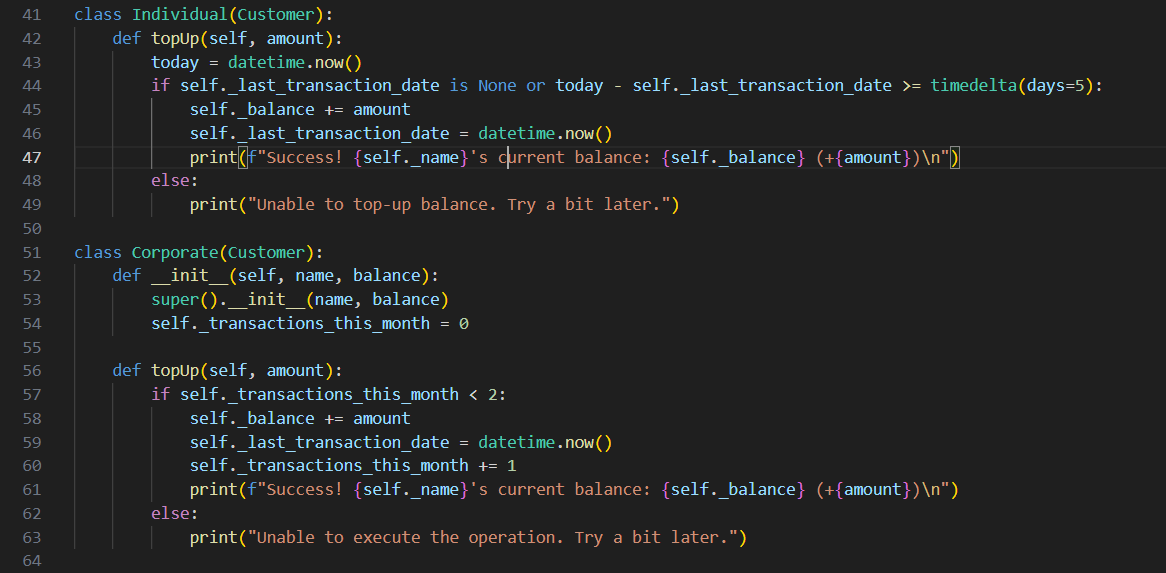


При виконанні даної програми виникає помилка, оскільки у класі **ModernCar** наразі немає реалізації методу **upgrade()**. Тому Python звертається до батьківського класу **Transport**, щоб знайти в ньому реалізацію відповідного методу. Оскільки батьківський клас **Transport** також не має реалізації методу **upgrade()**, виникає помилка **NotImplementedError**.

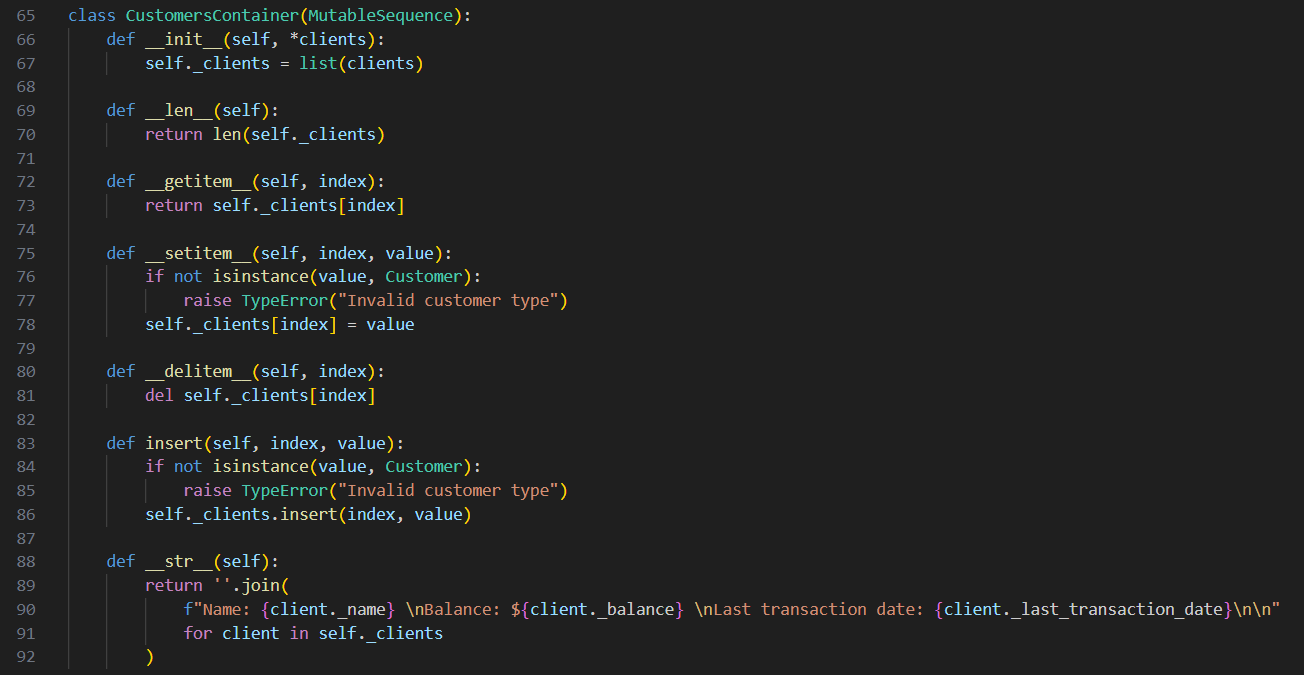
Індивідуальне завдання



Описую абстрактний клас **Customer**, який містить ім’я клієнта, поточний баланс його рахунку та дату останньої транзакції (за замовчуванням None). В цьому класі наявний абстрактний метод **topUp()** (поповнення рахунку, його буде перевизначено у класах-потомках), а також реалізовано методи **withdraw()** (зняття коштів) і **transfer()** (переказ коштів).

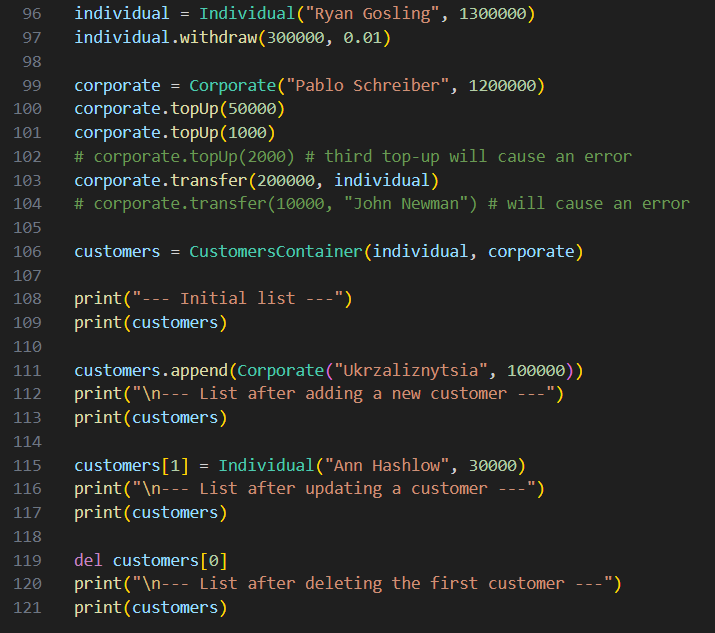


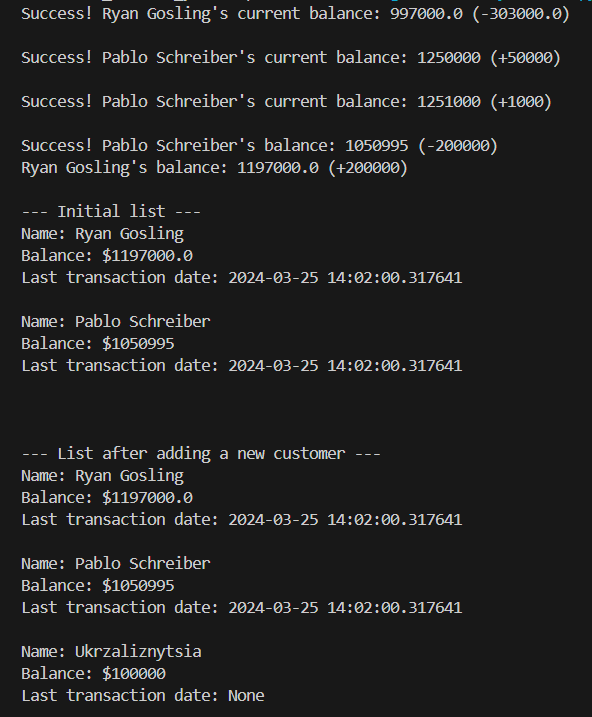
Далі представлено реалізацію класів **Individual** і **Customer**, які наслідують клас **Customer**, описаний вище. У класі **Individual** метод **topUp()** перевіряє дату останньої транзакції. Якщо вона None або з її моменту минуло щонайменше 5 днів, то клієнту дозволяється поповнити рахунок, інакше йому слід зачекати. Водночас у класі **Corporate** перевіряється кількість здійснених поповнень рахунку протягом місяця. Якщо це значення перевищує 2, то поповнення не вдасться здійснити до настання наступного місяця.

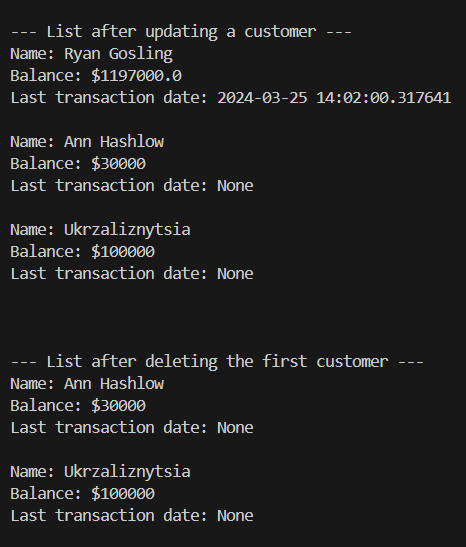


Клас-контейнер **CustomersContainer**, який зберігає інформацію про всіх клієнтів банку. Для його реалізації використано абстрактний базовий клас **MutableSequence** з модуля **collections.abc**. Цей клас надає методи, які повинен імплементувати підклас для того, щоб його можна було використовувати в ролі змінюваної послідовності (методи **\_\_getitem\_\_, \_\_setitem\_\_, \_\_delitem\_\_, \_\_len\_\_** та **insert**). Для виведення інформації про кожного клієнта використано магічний метод **\_\_str\_\_**.

Основна програма та результат її роботи мають вигляд:







**Висновок:** в ході виконання лабораторної роботи я детально ознайомився з поняттям абстрактних класів та класів-контейнерів у Python, дізнався якими засобами вони реалізуються, а також написав власну програму обігу коштів клієнтів банку, використовуючи здобуті знання.