**Домашня робота №51 ООП. Створення класів.**

**Завдання. «Банківський рахунок»**

1. Створіть клас BankAccount, який має атрибути "номер рахунку" account\_number і "баланс" balance. Реалізуйте методи \_\_init\_\_ та \_\_str\_\_. Протестуйте: ініціалізуйте деякий об'єкт зі номером рахунку та початковим балансом. Протестуйте \_\_str\_\_.
2. Реалізуйте і додайте до класу методи для додавання грошей на рахунок deposit і зняття грошей з рахунку withdraw для об’єкту. Створіть кілька об'єктів класу і виконайте операції з ними, щоб перевірити їх працездатність: використайте методи `deposit` та `withdraw` для здійснення операцій з балансом.
3. Розширимо клас "BankAccount" додатковими полями і методами. Додамо нове поле "owner\_name" для зберігання імені власника рахунку та метод "change\_owner\_name" для зміни цього поля. Додамо також метод "display\_account\_info", який виводитиме повну інформацію про рахунок. Протестуйте виконання даного функціоналу.
4. Для реалізації взаємодії між двома рахунками, додамо до класу "BankAccount" новий (instance) метод "transfer", який дозволить переказувати кошти з одного рахунку на інший, при цьому передбачити зняття коштів у поточного об’єкта і зарахування на інший рахунок. Обробити виняткові ситуації, які можуть трапитись (наприклад, від’ємний переказ, некоректна передача в аргументи отримувача тощо). Протестувати виконання і перевірити поточний баланс перед переказом і після переказу між двома банківськими рахунками.
5. Додати статичний метод "check\_account\_number" до класу "BankAccount", який перевіряє, чи є вказаний номер рахунку валідним (наприклад, вважаємо, що account\_number складається рівно із 5 цифр). Протестувати виконання статичного методу на самому класі і на деякому з його екземплярі.
6. Захистіть атрибути balance і account\_number у даному класі, щоб доступ до нього як до атрибуту ззовні класу був захищений (protected) і приватний (private) відповідно.
7. Отримайте поточний номер об’єкта за допомогою методу `get\_ account\_number ` та виведіть його (getter)
8. Реалізуйте зміну поточного номеру об’єкта за допомогою метода `set\_account\_number ` (setter)
9. За допомогою декораторів @property та @ account\_number.setter забезпечити доступ до приватного атрибуту (поля) "account\_number" і можливість присвоєння нових значень для цього атрибуту. Протестувати виконання.