

Практическое задание.

Теоретическая часть для выполнения.

1. Номер счёта должен состоять из 10 случайных цифр (например, 0578065418). Для генерации случайных чисел можно использовать библиотеку random, а именно функцию randint().

Можно использовать генератор списков, который создаст список случайных чисел. Затем можно объединить их в строку с помощью map() или join().

2. Передача other_account в метод transfer(). Метод transfer() предназначен для перевода денег другому пользователю. Использование isinstance(). Перед переводом можно добавить проверку.

Практическая часть.

Задание. Разработать класс BankAccount для управления банковским счётом.

Требования к классу.

Атрибуты:

- owner (владелец счёта, строка)
- balance (баланс, число с плавающей точкой)
- account_number (уникальный номер счёта, генерируется автоматически длиной в 10 цифр)

Методы:

- deposit(amount) — пополнение баланса на указанную сумму.
- withdraw(amount) — снятие указанной суммы, если хватает средств.
- get_balance() — возвращает текущий баланс.
- transfer(amount, other_account) — перевод денег на другой счёт, если хватает средств.

Пример входных данных:

```
owner_1 = BankAccount("N. Muller", 10_000)
owner_1.get_balance()
owner_1.deposit(5_000)
owner_1.get_balance()
owner_1.withdraw(20_000)
owner_1.withdraw(3_500)

owner_2 = BankAccount("O. Smith", 5_000)
owner_2.get_balance()

owner_1.transfer(1_000_000, "Kakoy-to Oleg")
owner_1.transfer(1_500, owner_2)
```

Выходные данные:

```
-----  
Владелец: N. Muller  
Номер счёта: 8524819593  
Баланс: 10000  
-----  
Пополнение 5000. Баланс: 15000  
-----  
Владелец: N. Muller  
Номер счёта: 8524819593  
Баланс: 15000  
-----  
Снятие невозможно! Не хватает средств.  
Снятие 3500. Баланс: 11500  
-----  
Владелец: O. Smith  
Номер счёта: 3010368679  
Баланс: 5000  
-----  
Ошибка! Получатель должен быть счётом банка.  
Перевод 1500 выполнен на счёт O. Smith.  
Баланс: 10000
```