

Лабораторна робота № 9

Тема: Ланцюжок викликів

Мета: Рефакторінг коду з проблемою «Ланцюжок викликів».

Завдання:

- У прикладі вище, ви бачите як створюється ланцюжок викликів `bank.get_customer().get_account().get_balance()`, що може викликати проблеми з підтримкою та розширенням коду. Цей ланцюжок також збільшує залежність між класами. Реорганізуйте код, щоб уникнути прямого ланцюжка викликів через делегування обов'язків або введення методу, який інкапсулює цей ланцюжок.

```
class Customer:
    def __init__(self, account):
        self.account = account

    def get_account(self):
        return self.account

class Account:
    def __init__(self, balance):
        self.balance = balance

    def get_balance(self):
        return self.balance

class Bank:
    def __init__(self, customer):
        self.customer = customer

    def get_customer(self):
        return self.customer

# Користування ланцюжком викликів
bank = Bank(Customer(Account(1000)))
balance = bank.get_customer().get_account().get_balance()
print(f"The balance is: {balance}")
```

Звіт:

Звіт має включати в себе один файл:

1. **output_1.py** - код після рефакторінгу з п.1

За необхідності можна додати файл **README.md** для опису прийнятих рішень щодо рефакторінгу.

Робота може бути виконана на будь якій мові програмування або на псевдокодi.