Software Systems and Design - Report

by Elizaveta Zagurskih

April 15, 2024

Design Patterns

Singleton

Singleton is a creational design pattern that guarantees the existence of the only instance of the class and provides global point of access to that instance.

In my program, Singleton pattern is used in the Banking System class to ensure the uniqueness of the system instance. This ensures that there is only one instance of the banking system managing all bank accounts. This design choice provides a global point of access to the banking system instance, allowing all clients to interact with the system seamlessly while maintaining data consistency and integrity.

State

State is a behavioral design pattern, which allows an object to alter its behavior when its internal state changes.

In my program, State pattern is used to represent state of the bank account: either "Active" or "Inactive". Each type of bank account in the system implements the "AccountState" interface, which defines methods for state transition and behavior modification. For instance, when an account is activated or deactivated, its behavior changes accordingly. By implementing state-specific functionality for the activate() and deactivate() methods in each bank account class, we ensure clear representation of state transitions and enable customized behavior for different account types.

Proxy

Proxy is a structural design patter that provides a surrogate or placeholder for another object to control access to it.

In my program, Proxy pattern is implemented through the "BankingSystemProxy" interface, which defines the methods for managing bank accounts and facilitating transactions. The BankingSystem class acts as a proxy for accessing and controlling the behavior of the actual bank account objects (SavingsAccount, CheckingAccount, BusinessAccount). Each method in the BankingSystem class delegates the execution to the corresponding method of the actual account object.

This pattern allows to control access to bank account objects, enforcing security measures (pattern encapsulates the bank account functionality, ensuring that it cannot be directly altered).

UML Diagram

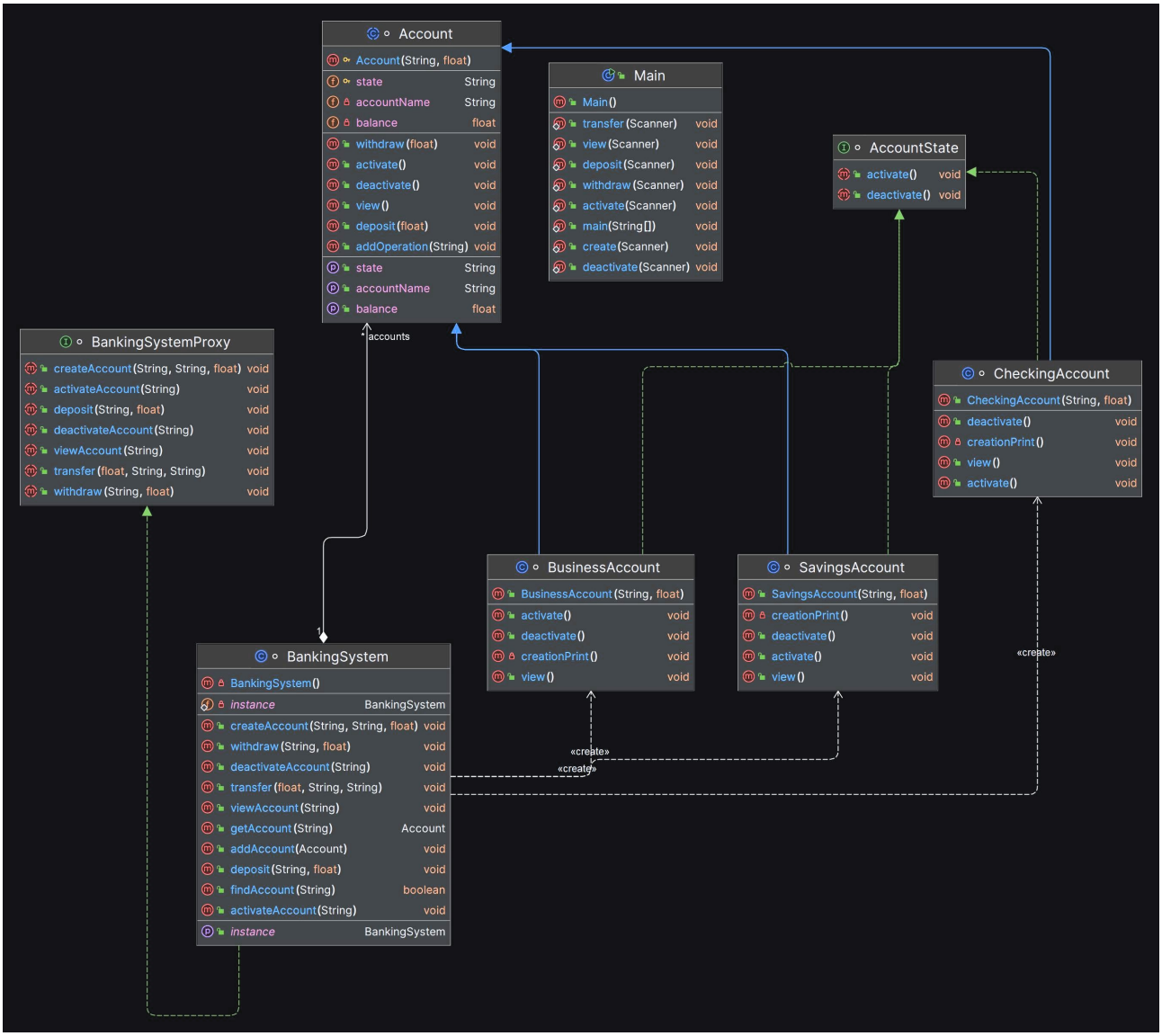

Figure 1: Assignment 3. UML diagram