James Zafiri

09/16/2023

CSC6301 Module 3

Project 3 – Encapsulation & Modularity Guidelines Code Analysis

**Encapsulation:**

The Person class makes sure to encapsulate the person’s data as well as their bank account. These variables (name, age, and account) are declared as private. It is key to make sure that data fields are private. The BankAccount class also does the same with making sure to encapsulate the accounts data. These variables (accountNumber, balance) are declared as private. One thing to take note of is the setter method, which we know should try to be avoided. However, encapsulation is followed nicely here, and the internal data of the object is not exposed, but still able to be accessed by the needed methods.

**Modularity:**

This program shows modularity by dividing the functionality into separate classes (Person, BankAccount, Bank). Each one of these classes has a specific responsibility. Person is meant to hold the information of the individual, BankAccount is for representing the account itself, and Bank is meant for the main method to execute. These classes are independent, which promotes loose coupling since they can be edited/extended without affecting the others. This code is also structured nicely which makes it clear to see what each part’s role is.

All in all, I would say that this code follows the guidelines for encapsulation and modularity, and it is easy to read/work with. We also saw that it follows the Single Responsibility Principle from SOLID programming since each class is independent.