

Banking Application Report One

Abstract Classes

An abstract class cannot be instantiated and is utilized as a base class for other classes. The class can have abstract methods. In the UML diagram, two classes; User and Account are defined as abstract classes, providing the structure for BankEmployee, Customer, SavingsAccount, and CurrentAccount classes.

Concrete Classes

These are classes that can be instantiated. In the UML diagram, BankSystem, BankEmployee, Customer, SavingsAccount, CurrentAccount, Transaction, File Handler, Database Handler, and Login are all concrete classes.

Inheritance

Inheritance is a key concept in OOP. Through inheritance, a new class (derived class) inherits from an existing class. The derived class inherits all the properties and methods of the base class, besides adding other capabilities. BankEmployee and Customer classes inherit from the User class, while SavingsAccount and CurrentAccount classes inherit from the Account class in the UML diagram, .

Polymorphism

Polymorphism provides the ability to perform one action in many other different ways. In the UML diagram, polymorphism is utilized in the Account class methods addMoney() and subtractMoney() whereby they will possess different implementations in the SavingsAccount and CurrentAccount classes.

Public, Private, and Protected Methods

These methods are access modifiers and they set the visibility of methods and attributes. Public methods are accessible from any class, while private methods are accessible only within the class where they are defined. Protected methods on the other hand are accessible within the class they are defined and from any subclass.

In the UML diagram, methods prefixed with “+” are public, “-” are private, and “#” are protected. These methods are used to control the visibility and accessibility of methods in the banking application system.