// Book.java

public class Book {

// Private attributes

private String title;

private String author;

private String isbn;

// Parameterized constructor to initialize all attributes

public Book(String title, String author, String isbn) {

this.title = title;

this.author = author;

this.isbn = isbn;

}

// Method to show details of the book

public void showDetails() {

System.out.println("Title: " + title + ", Author: " + author + ", ISBN: " + isbn);

}

// Getter method to return the book title

public String getTitle() {

return title;

}

}

// Library.java

import java.util.\*;

public class Library {

// Private attributes

private String name;

private List<Book> books;

// Constructor to initialize the library name and list

public Library(String name) {

this.name = name;

this.books = new ArrayList<>();

}

// Add a book to the library

public void addBook(Book book) {

books.add(book);

System.out.println("Added book '" + book.getTitle() + "' to " + name + " Library");

}

// Display all books in the library

public void showBooks() {

System.out.println("\nBooks in " + name + " Library:");

for (Book book : books) {

book.showDetails();

}

}

}

// Member.java

import java.util.\*;

public class Member {

// Private attributes

private String name;

private List<Book> borrowedBooks;

// Constructor to initialize the member name and list

public Member(String name) {

this.name = name;

this.borrowedBooks = new ArrayList<>();

}

// Borrow a book from the library

public void borrowBook(Book book) {

borrowedBooks.add(book);

System.out.println(name + " borrowed book: " + book.getTitle());

}

// Show all borrowed books

public void showBorrowedBooks() {

System.out.println("\nBooks borrowed by " + name + ":");

for (Book book : borrowedBooks) {

book.showDetails();

}

}

}

// LibraryDemo.java

public class LibraryDemo {

public static void main(String[] args) {

// Step 1 - Create a Library object

Library lib = new Library("Central City");

// Step 2 - Create 3 Book objects with sample data

Book book1 = new Book("Java Programming", "James Gosling", "ISBN-001");

Book book2 = new Book("Data Structures", "Robert Sedgewick", "ISBN-002");

Book book3 = new Book("Design Patterns", "Gang of Four", "ISBN-003");

// Step 3 - Add books to library using addBook()

lib.addBook(book1);

lib.addBook(book2);

lib.addBook(book3);

// Step 4 - Display all books in the library using showBooks()

lib.showBooks();

// Step 5 - Create a Member object

Member member = new Member("Ravi");

// Step 6 - Borrow 2 books using borrowBook()

System.out.println();

member.borrowBook(book1);

member.borrowBook(book3);

// Step 7 - Display borrowed books using showBorrowedBooks()

member.showBorrowedBooks();

}

}

2.

// Product.java

public class Product {

// Private attributes

private String name;

private double price;

// Parameterized constructor to initialize all attributes

public Product(String name, double price) {

this.name = name;

this.price = price;

}

// Method to display product info

public void showDetails() {

System.out.println("Product: " + name + ", Price: ₹" + price);

}

// Getter for product name

public String getName() {

return name;

}

}

// Order.java

import java.util.\*;

public class Order {

// Private attributes

private String orderId;

private List<Product> products;

// Constructor to initialize orderId and list

public Order(String orderId) {

this.orderId = orderId;

this.products = new ArrayList<>();

}

// Add a product to order

public void addProduct(Product product) {

products.add(product);

System.out.println("Added product '" + product.getName() + "' to Order " + orderId);

}

// Show order details

public void showOrderDetails() {

System.out.println("\nOrder " + orderId + " contains:");

for (Product product : products) {

product.showDetails();

}

}

// Getter for orderId

public String getOrderId() {

return orderId;

}

}

// Customer.java

import java.util.\*;

public class Customer {

// Private attributes

private String name;

private String email;

private List<Order> orders;

// Constructor to initialize customer info

public Customer(String name, String email) {

this.name = name;

this.email = email;

this.orders = new ArrayList<>();

}

// Place an order

public void placeOrder(Order order) {

orders.add(order);

System.out.println(name + " placed Order " + order.getOrderId());

}

// Display all orders for this customer

public void showCustomerOrders() {

System.out.println("\n========================================");

System.out.println("Orders placed by " + name + " (" + email + "):");

System.out.println("========================================");

for (Order order : orders) {

order.showOrderDetails();

}

}

}

// ShoppingDemo.java

public class ShoppingDemo {

public static void main(String[] args) {

System.out.println("=== Online Shopping System Demo ===\n");

// Step 1 - Create Customer object

Customer customer1 = new Customer("Amit", "amit@gmail.com");

// Step 2 - Create Product objects

Product product1 = new Product("Laptop", 55000);

Product product2 = new Product("Mobile", 25000);

Product product3 = new Product("Mouse", 500);

System.out.println("Creating Orders and Adding Products:\n");

// Step 3 - Create 2 Order objects and add different products to each

Order order1 = new Order("ORD001");

order1.addProduct(product1); // Laptop to Order 1

order1.addProduct(product2); // Mobile to Order 1

System.out.println();

Order order2 = new Order("ORD002");

order2.addProduct(product2); // Mobile to Order 2

order2.addProduct(product3); // Mouse to Order 2

System.out.println("\nAssociating Orders with Customer:\n");

// Step 4 - Associate orders with customer using placeOrder()

customer1.placeOrder(order1);

customer1.placeOrder(order2);

// Step 5 - Display all orders and their products using showCustomerOrders()

customer1.showCustomerOrders();

System.out.println("\n========================================");

System.out.println("NOTE: This demo represents Object Diagram");

System.out.println("runtime objects and their links.");

System.out.println("========================================");

}

}

3.

// BankAccount.java

public class BankAccount {

// Private attributes

private String accountNumber;

private double balance;

private int pin;

// Constructor to initialize all fields

public BankAccount(String accountNumber, double balance, int pin) {

this.accountNumber = accountNumber;

this.balance = balance;

this.pin = pin;

}

// Validate PIN

public boolean validatePin(int enteredPin) {

return enteredPin == pin;

}

// Debit amount from account

public void debit(double amount) {

balance -= amount;

System.out.println("₹" + amount + " withdrawn. Remaining balance: ₹" + balance);

}

// Getter for account number (optional, for display)

public String getAccountNumber() {

return accountNumber;

}

// Getter for balance (optional, for display)

public double getBalance() {

return balance;

}

}

// ATM.java

public class ATM {

// Attribute: linked BankAccount

private BankAccount linkedAccount;

// Constructor to link an account

public ATM(BankAccount linkedAccount) {

this.linkedAccount = linkedAccount;

}

// Perform withdrawal process

public void withdraw(int enteredPin, double amount) {

System.out.println("\n--- ATM Processing Transaction ---");

System.out.println("Validating PIN...");

// Validate PIN using linkedAccount.validatePin()

if (linkedAccount.validatePin(enteredPin)) {

System.out.println("PIN Validated Successfully!");

System.out.println("Processing withdrawal of ₹" + amount + "...");

// Call linkedAccount.debit(amount)

linkedAccount.debit(amount);

System.out.println("Dispensing cash...");

System.out.println("✓ Transaction Successful!");

System.out.println("Please collect your cash.");

} else {

System.out.println("✗ Invalid PIN. Transaction failed.");

}

System.out.println("----------------------------------\n");

}

}

// Customer.java

public class Customer {

// Private attributes

private String name;

private ATM atm;

// Constructor to initialize customer name and ATM reference

public Customer(String name, ATM atm) {

this.name = name;

this.atm = atm;

}

// Interact with ATM to perform withdrawal

public void performWithdrawal(int pin, double amount) {

System.out.println(name + " is requesting withdrawal of ₹" + amount + "...");

// Call atm.withdraw(pin, amount)

atm.withdraw(pin, amount);

}

// Getter for customer name

public String getName() {

return name;

}

}

// ATMDemo.java

public class ATMDemo {

public static void main(String[] args) {

System.out.println("╔════════════════════════════════════════╗");

System.out.println("║ ATM TRANSACTION SYSTEM DEMO ║");

System.out.println("║ Sequence Diagram Implementation ║");

System.out.println("╚════════════════════════════════════════╝\n");

// Step 1 - Create BankAccount object with sample data

BankAccount account = new BankAccount("ACC123456", 50000.0, 1234);

System.out.println("Bank Account Created:");

System.out.println("Account Number: " + account.getAccountNumber());

System.out.println("Initial Balance: ₹" + account.getBalance());

System.out.println("PIN: \*\*\*\*\n");

// Step 2 - Create ATM object linked to BankAccount

ATM atm = new ATM(account);

System.out.println("ATM Initialized and linked to account.\n");

// Step 3 - Create Customer object associated with ATM

Customer customer = new Customer("Rajesh Kumar", atm);

System.out.println("Customer: " + customer.getName());

System.out.println("═══════════════════════════════════════════\n");

// Step 4 - Call performWithdrawal() with correct PIN

System.out.println("【 TRANSACTION 1: Valid PIN 】");

customer.performWithdrawal(1234, 5000.0);

// Step 5 - Call performWithdrawal() with incorrect PIN

System.out.println("【 TRANSACTION 2: Invalid PIN 】");

customer.performWithdrawal(9999, 3000.0);

// Step 6 - Another successful transaction

System.out.println("【 TRANSACTION 3: Valid PIN 】");

customer.performWithdrawal(1234, 2000.0);

System.out.println("═══════════════════════════════════════════");

System.out.println("NOTE: This demonstrates the Sequence flow:");

System.out.println("Customer → ATM → BankAccount");

System.out.println("═══════════════════════════════════════════");

}

}