Day 3 Assignment

BankOperations.java (Interface)

```
public interface BankOperations {
    void deposit(double amount);
    void withdraw(double amount);
    double getBalance();
}
```

Account.java (Abstract Class)

```
public abstract class Account implements BankOperations {
    protected String accountNumber;
    protected double balance;
    protected Customer customer;
    public Account(String accountNumber, Customer customer) {
        this.accountNumber = accountNumber;
        this.customer = customer;
        this.balance = 0.0;
    }
    public String getAccountNumber() {
        return accountNumber;
    }
    public Customer getCustomer() {
        return customer;
    }
}
```

SavingsAccount.java

```
public class SavingsAccount extends Account {
    public SavingsAccount(String accountNumber, Customer customer) {
        super(accountNumber, customer);
    }
```

```
    public void deposit(double amount) {
        if (amount > 0) balance += amount;
    }
    public void withdraw(double amount) {
        if (amount > 0 && balance >= amount) balance -= amount;
    }
    public double getBalance() {
        return balance;
    }
}
```

CurrentAccount.java

```
public class CurrentAccount extends Account {
    public CurrentAccount(String accountNumber, Customer customer) {
        super(accountNumber, customer);
    }
    public void deposit(double amount) {
        if (amount > 0) balance += amount;
    }
    public void withdraw(double amount) {
        if (amount > 0 && balance >= amount) balance -= amount;
    }
    public double getBalance() {
        return balance;
    }
}
```

Customer.java

```
public class Customer {
    private String name;
    private String customerId;
```

```
    public Customer(String name, String customerId) {
        this.name = name;
        this.customerId = customerId;
    }
    public String getName() {
        return name;
    }
    public String getCustomerId() {
        return customerId;
    }
}
```

BankBranch.java

```
public class BankBranch {
    private String branchName;
    private String branchCode;

    public BankBranch(String branchName, String branchCode) {
        this.branchName = branchName;
        this.branchCode = branchCode;
    }
    public String getBranchName() {
        return branchName;
    }
    public String getBranchCode() {
        return branchCode;
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
```

```
        BankBranch branch = new BankBranch("City Center", "BNK001");
        Customer customer1 = new Customer("Alice", "C001");
        Customer customer2 = new Customer("Bob", "C002");
        SavingsAccount savings = new SavingsAccount("SA1001", customer1);
        CurrentAccount current = new CurrentAccount("CA2001", customer2);
        savings.deposit(1000);
        savings.withdraw(200);
        current.deposit(2000);
        current.withdraw(500);
        System.out.println("Bank Branch: " + branch.getBranchName() + " (" + branch.getBranchCode() +
")");
        System.out.println("Savings Account Details: ");
        System.out.println("Customer: " + savings.getCustomer().getName());
        System.out.println("Account Number: " + savings.getAccountNumber());
        System.out.println("Balance: " + savings.getBalance());
        System.out.println("\nCurrent Account Details: ");
        System.out.println("Customer: " + current.getCustomer().getName());
        System.out.println("Account Number: " + current.getAccountNumber());
        System.out.println("Balance: " + current.getBalance());
    }
}
```

Sample Output

```
Bank Branch: City Center (BNK001)
Savings Account Details:
Customer: Alice
Account Number: SA1001
Balance: 800.0

Current Account Details:
Customer: Bob
Account Number: CA2001
Balance: 1500.0
```

Also:

```
public class BankApp {
    public interface BankOperations {
        void deposit(double amount);
        void withdraw(double amount);
        double getBalance();
    }
    public abstract static class Account implements BankOperations {
        protected String accountNumber;
        protected double balance;
        protected Customer customer;
        public Account(String accountNumber, Customer customer) {
            this.accountNumber = accountNumber;
            this.customer = customer;
            this.balance = 0.0;
        }
        public String getAccountNumber() {
            return accountNumber;
        }
        public Customer getCustomer() {
            return customer;
        }
    }
    public static class SavingsAccount extends Account {
        public SavingsAccount(String accountNumber, Customer customer) {
            super(accountNumber, customer);
        }

        public void deposit(double amount) {
            if (amount > 0) balance += amount;
        }
        public void withdraw(double amount) {
            if (amount > 0 && balance >= amount) balance -= amount;
```

```
    }
    public double getBalance() {
        return balance;
    }
}
public static class CurrentAccount extends Account {
    public CurrentAccount(String accountNumber, Customer customer) {
        super(accountNumber, customer);
    }
    public void deposit(double amount) {
        if (amount > 0) balance += amount;
    }
    public void withdraw(double amount) {
        if (amount > 0 && balance >= amount) balance -= amount;
    }
    public double getBalance() {
        return balance;
    }
}
public static class Customer {
    private String name;
    private String customerId;
    public Customer(String name, String customerId) {
        this.name = name;
        this.customerId = customerId;
    }
    public String getName() {
        return name;
    }
    public String getCustomerId() {
        return customerId;
    }
}
public static class BankBranch {
```

```
    private String branchName;
    private String branchCode;
    public BankBranch(String branchName, String branchCode) {
        this.branchName = branchName;
        this.branchCode = branchCode;
    }
    public String getBranchName() {
        return branchName;
    }
    public String getBranchCode() {
        return branchCode;
    }
}

public static void main(String[] args) {
    BankBranch branch = new BankBranch("City Center", "BNK001");
    Customer customer1 = new Customer("Alice", "C001");
    Customer customer2 = new Customer("Bob", "C002");
    SavingsAccount savings = new SavingsAccount("SA1001", customer1);
    CurrentAccount current = new CurrentAccount("CA2001", customer2);
    savings.deposit(1000);
    savings.withdraw(200);
    current.deposit(2000);
    current.withdraw(500);
    System.out.println("Bank Branch: " + branch.getBranchName() + " (" + branch.getBranchCode() +
")");
    System.out.println("Savings Account Details:");
    System.out.println("Customer: " + savings.getCustomer().getName());
    System.out.println("Account Number: " + savings.getAccountNumber());
    System.out.println("Balance: " + savings.getBalance());
    System.out.println("\nCurrent Account Details:");
    System.out.println("Customer: " + current.getCustomer().getName());
    System.out.println("Account Number: " + current.getAccountNumber());
    System.out.println("Balance: " + current.getBalance());
```

```
    }
}
```

Sample Output:

Bank Branch: City Center (BNK001)

Savings Account Details:

Customer: Alice

Account Number: SA1001

Balance: 800.0

Current Account Details:

Customer: Bob

Account Number: CA2001

Balance: 1500.0

How to Compile & Run:

Save as BankApp.java

Compile: javac BankApp.java

Run: java BankApp