Build a banking system

Step 1: Create a Class for Bank Accounts

1. **Create a class called BankAccount.**
   - The class should have the following private fields:
     - accountNumber (String)
     - balance (BigDecimal)
     - accountHolder (String)
   - Create a constructor that initializes these fields. For balance, initialize it with BigDecimal.ZERO.
2. **Add public methods for encapsulation:**
   - deposit(BigDecimal amount) that adds the amount to the balance.
   - withdraw(BigDecimal amount) that subtracts the amount from the balance, but only if the balance is sufficient. Use BigDecimal.compareTo() to check the balance.
   - getBalance() that returns the current balance.

Step 2: Introduce Inheritance

3. **Create a subclass called SavingsAccount that inherits from BankAccount.**
   - Add a new private field: interestRate (BigDecimal).
   - Create a constructor that initializes the fields of the superclass (BankAccount) and interestRate.
   - Add a method applyInterest() that calculates interest as balance * interestRate and adds it to the balance. Use balance.multiply(interestRate) for this.

Step 3: Add Polymorphism with Method Overriding

4. **Override the withdraw(BigDecimal amount) method in SavingsAccount.**
   - In this subclass, ensure that the balance does not drop below $500. If an attempt is made to withdraw and the balance would drop below $500, print a message and prevent the withdrawal. Use BigDecimal.valueOf(500) for comparison.

Step 4: Implement Abstraction with an Interface

5. **Create an interface called BankServices.**
   - Add the following abstract methods:
     - void deposit(BigDecimal amount)
     - void withdraw(BigDecimal amount)

        - BigDecimal getBalance()

6. **Have BankAccount implement the BankServices interface.**
    - Ensure that BankAccount provides implementations for the methods in BankServices.

Step 5: Demonstrate Encapsulation, Inheritance, Polymorphism, and Abstraction

7. **Write a main method to test your classes.**
    - Create a SavingsAccount object.
    - Try depositing and withdrawing money using the methods from BankAccount and SavingsAccount.
    - Apply interest using the applyInterest() method.
    - Show how polymorphism works by calling the overridden withdraw() method on the SavingsAccount.
    - Try to access the balance directly from outside the class. What happens? Why? Explain briefly in comments how encapsulation is protecting the data.