Write a Java program to create a base class BankAccount with methods deposit() and withdraw(). Create two subclasses SavingsAccount and CheckingAccount. Override the withdraw() method in each subclass to impose different withdrawal limits and fees.

Sample Solution:

Java Code:

```
// BankAccount.java
// Define a class named BankAccount
class BankAccount {

  // Declare a private double variable balance
  private double balance;

  // Define a constructor that takes a double initialBalance as a parameter
  public BankAccount(double initialBalance) {
    // Assign the parameter initialBalance to the instance variable balance
    this.balance = initialBalance;
  }

  // Define a method to get the current balance
  public double getBalance() {
    // Return the current balance
    return balance;
  }

  // Define a method to deposit an amount to the balance
  public void deposit(double amount) {
    // Add the amount to the current balance
    balance += amount;
  }

  // Define a method to withdraw an amount from the balance
  public void withdraw(double amount) {
    // Check if the amount to be withdrawn is less than or equal to the cur
    if (amount <= balance) {
      // Subtract the amount from the current balance
      balance -= amount;
    } else {
      // Print "Insufficient funds." to the console if the balance is insuf
```

```
            System.out.println("Insufficient funds.");
        }
    }
}
```

```
// SavingsAccount.java
// Define a class named SavingsAccount that extends BankAccount
class SavingsAccount extends BankAccount {

    // Declare a private double variable withdrawalLimit
    private double withdrawalLimit;

    // Define a constructor that takes a double initialBalance and a double w
    public SavingsAccount(double initialBalance, double withdrawalLimit) {
        // Call the constructor of the superclass BankAccount with initialBalar
        super(initialBalance);
        // Assign the parameter withdrawalLimit to the instance variable withdr
        this.withdrawalLimit = withdrawalLimit;
    }

    // Override the withdraw method from the BankAccount class
    @Override
    public void withdraw(double amount) {
        // Check if the amount to be withdrawn is less than or equal to the wit
        if (amount <= withdrawalLimit) {
            // Call the withdraw method of the superclass BankAccount with amount
            super.withdraw(amount);
        } else {
            // Print "Withdrawal limit exceeded." to the console if the amount ex
            System.out.println("Withdrawal limit exceeded.");
        }
    }
}
```

```
// CheckingAccount.java
// Define a class named CheckingAccount that extends BankAccount
class CheckingAccount extends BankAccount {

    // Declare a private double variable overdraftFee
```

```
    private double overdraftFee;

    // Define a constructor that takes a double initialBalance and a double o
    public CheckingAccount(double initialBalance, double overdraftFee) {
      // Call the constructor of the superclass BankAccount with initialBalan
      super(initialBalance);
      // Assign the parameter overdraftFee to the instance variable overdraft
      this.overdraftFee = overdraftFee;
    }

    // Override the withdraw method from the BankAccount class
    @Override
    public void withdraw(double amount) {
      // Check if the amount to be withdrawn is less than or equal to the cur
      if (amount <= getBalance()) {
        // Call the withdraw method of the superclass BankAccount with amount
        super.withdraw(amount);
      } else {
        // Print "Overdraft fee applied." to the console if the amount exceed
        System.out.println("Overdraft fee applied.");
        // Call the withdraw method of the superclass BankAccount with the am
        super.withdraw(amount + overdraftFee);
      }
    }
  }
```

```
  // Main.java
  // Main class definition
  public class Main {
    // Main method, program entry point
    public static void main(String[] args) {
      // Create a SavingsAccount object with a balance of 2000 and a limit of
      BankAccount savingsAccount = new SavingsAccount(2000, 650);
      // Create a CheckingAccount object with a balance of 1000 and a limit o
      BankAccount checkingAccount = new CheckingAccount(1000, 100);

      // Withdraw 300 from the savings account
      withdrawFromAccount(savingsAccount, 300);
      // Withdraw 250 from the checking account
      withdrawFromAccount(checkingAccount, 250);
```

```
        // Print the current balance of the savings account
        System.out.println("Savings Account Balance: " + savingsAccount.getBala
        // Print the current balance of the checking account
        System.out.println("Checking Account Balance: " + checkingAccount.getBa
    }

    // Method to withdraw a specified amount from a given bank account
    public static void withdrawFromAccount(BankAccount account, double amount
        // Call the withdraw method on the account with the specified amount
        account.withdraw(amount);
    }
}
```

Output:

```
Savings Account Balance: 1700.0
Checking Account Balance: 750.0
```

Explanation: