APPLICATION DEVELOPMENT AND DEPLOYMENT ARCHITECTURE

THEORY DIGITAL ASSESSMENT

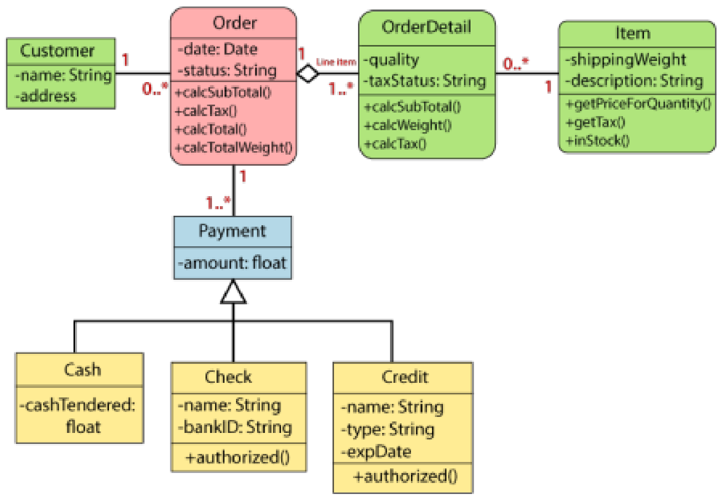
CHARATH H J (22MIC0088)

SREE KRISHNA A K (22MIC0093)

VIT Vellore Institute of Technology	Digital Assignment	
Course code: CSI3025	Course Title :Adavanced Development and Deployment Architecture	Batch : B1
	Course mode: Theory	Batch : B2
Class No:3194/3202	School: SCOPE	Venue: PRP G32
Emp. ID: 20774	Faculty Name: SUDHAKAR. P	**Code : B**

Instructions:

- Read the question carefully and answer to the question.
- Each questions carry marks.
- Submit the answers in the vtop (Before submission, double check document and submit the right document.)
- For any reason, resubmission or reconduct of DA was not allowed.
- Plagirism of any source is merely rejection of document and zero will be awarded. Hence do not ask your friends to submit / copy or do any action towards your assignment. No further consideration will be entertained.
- Your answer should be well explainable according to the question. If any code segment is required, include the same.
- Finally submit your answer as a pdf containing the answers and source code developed (after all questions are answered, add the source codes with appropriate explanation/ headline)



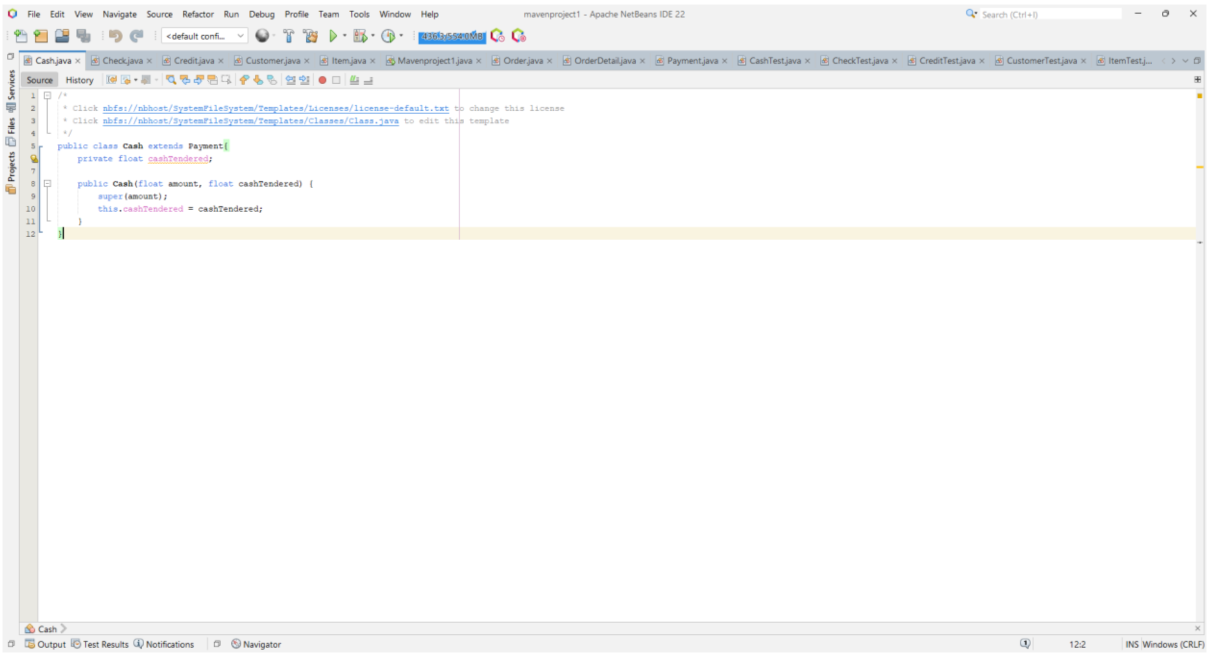
Question 1: Develop a prototype code for the Customer management software (refer Figure 1) using Java/C++/Python.[5 Marks]

Question 2: Develop a Unit test case for the given figure 1. Atleast 5 test cases must be developed [5 Marks]

Question 3: Study any automated testing software and test your code. Demonstrate your answers step-by-step screenshots alongwith necessary testing scripts. [10 Marks]

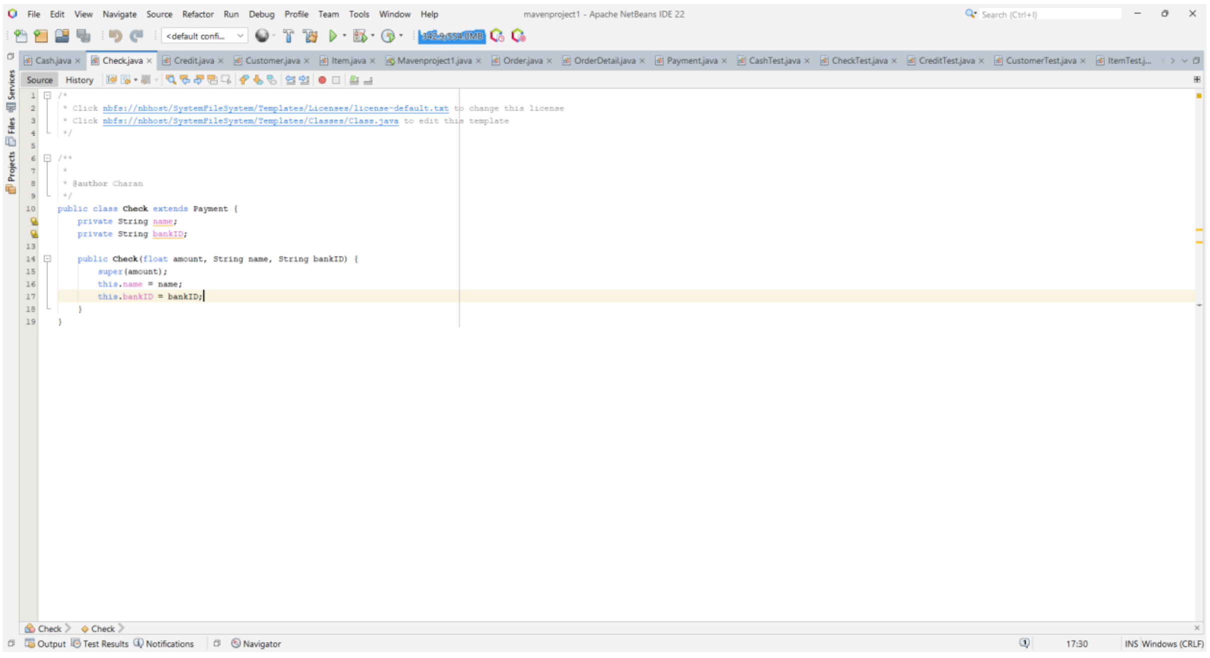
✓ **SOURCE PACKAGES:**

Cash.java:



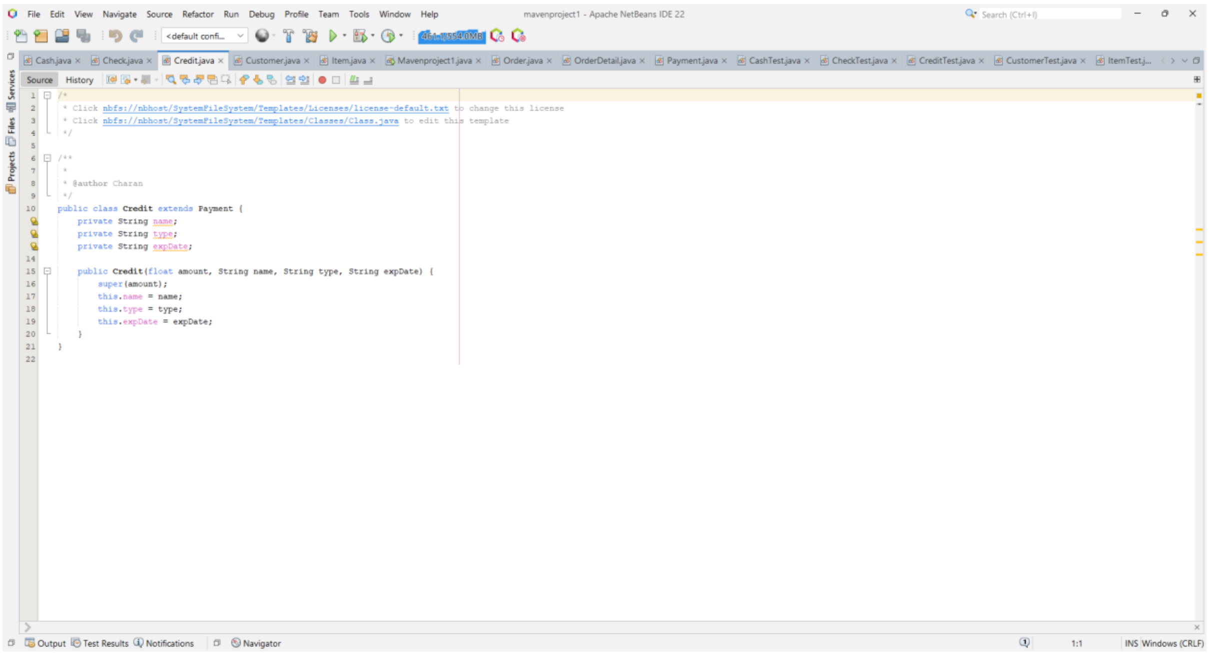
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
public class Cash extends Payment{
    private float cashTendered;

    public Cash(float amount, float cashTendered) {
        super(amount);
        this.cashTendered = cashTendered;
    }
}
```

Check.java



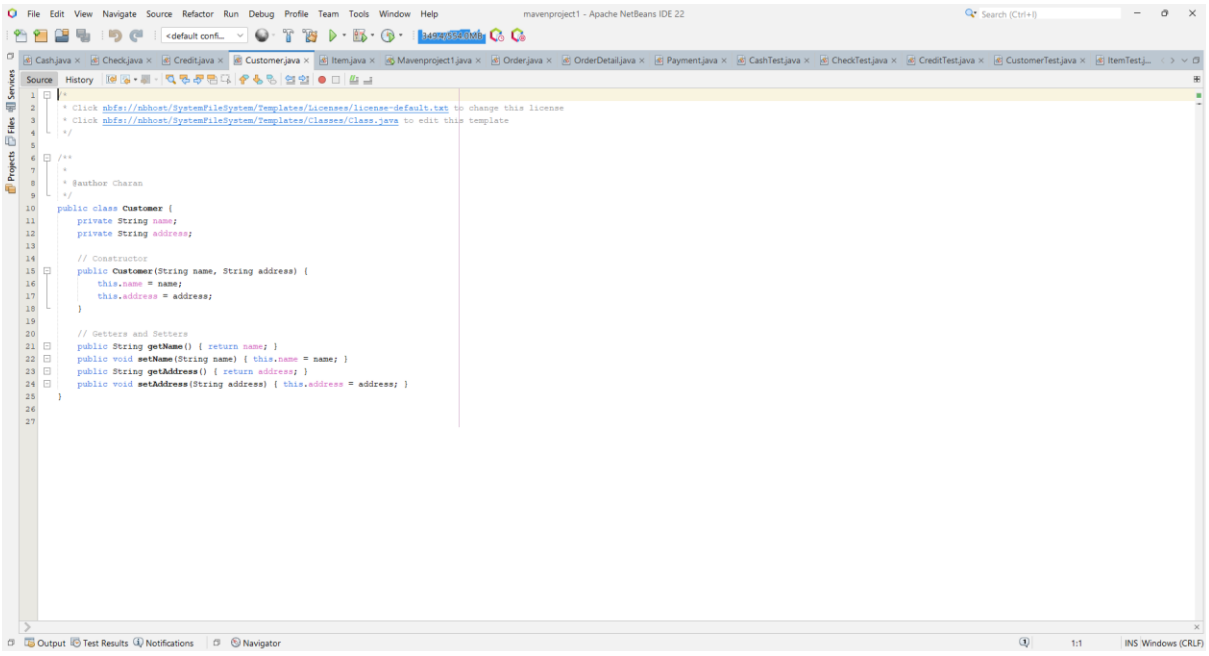
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author Charan
 */
public class Check extends Payment {
    private String name;
    private String bankID;

    public Check(float amount, String name, String bankID) {
        super(amount);
        this.name = name;
        this.bankID = bankID;
    }
}
```

Credit.java



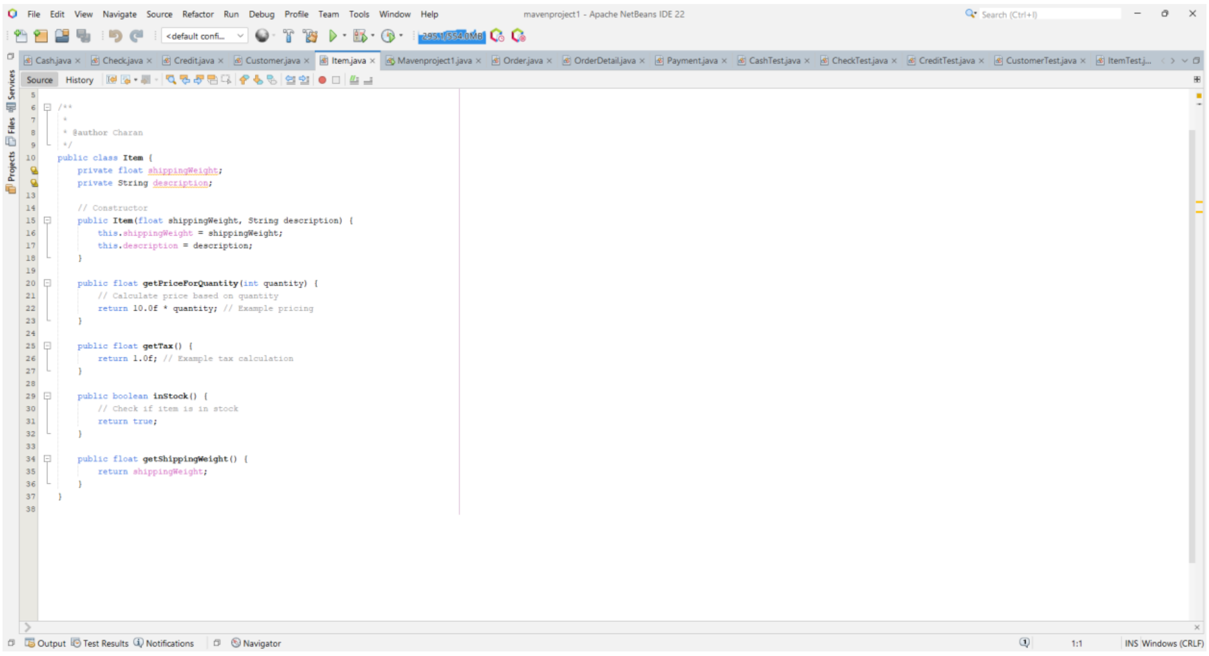
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author Charan
 */
public class Credit extends Payment {
    private String name;
    private String type;
    private String expDate;

    public Credit(float amount, String name, String type, String expDate) {
        super(amount);
        this.name = name;
        this.type = type;
        this.expDate = expDate;
    }
}
```

Customer.java



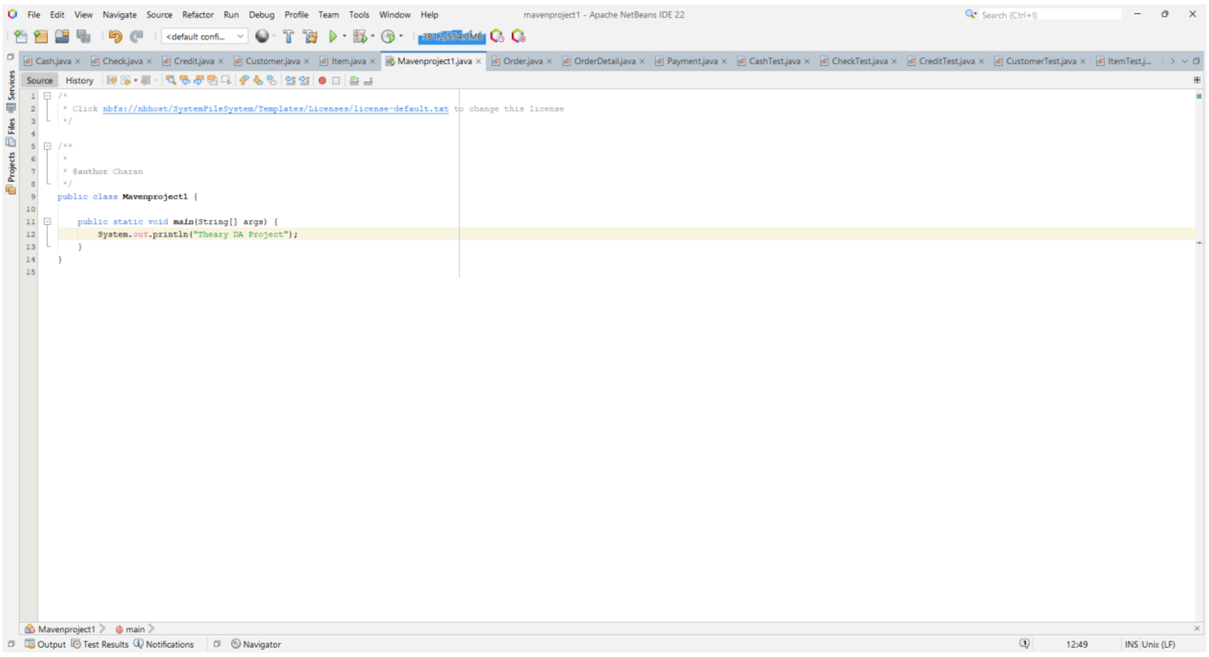
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author Charan
 */
public class Customer {
    private String name;
    private String address;

    // Constructor
    public Customer(String name, String address) {
        this.name = name;
        this.address = address;
    }

    // Getters and Setters
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getAddress() { return address; }
    public void setAddress(String address) { this.address = address; }
}
```

Item.java



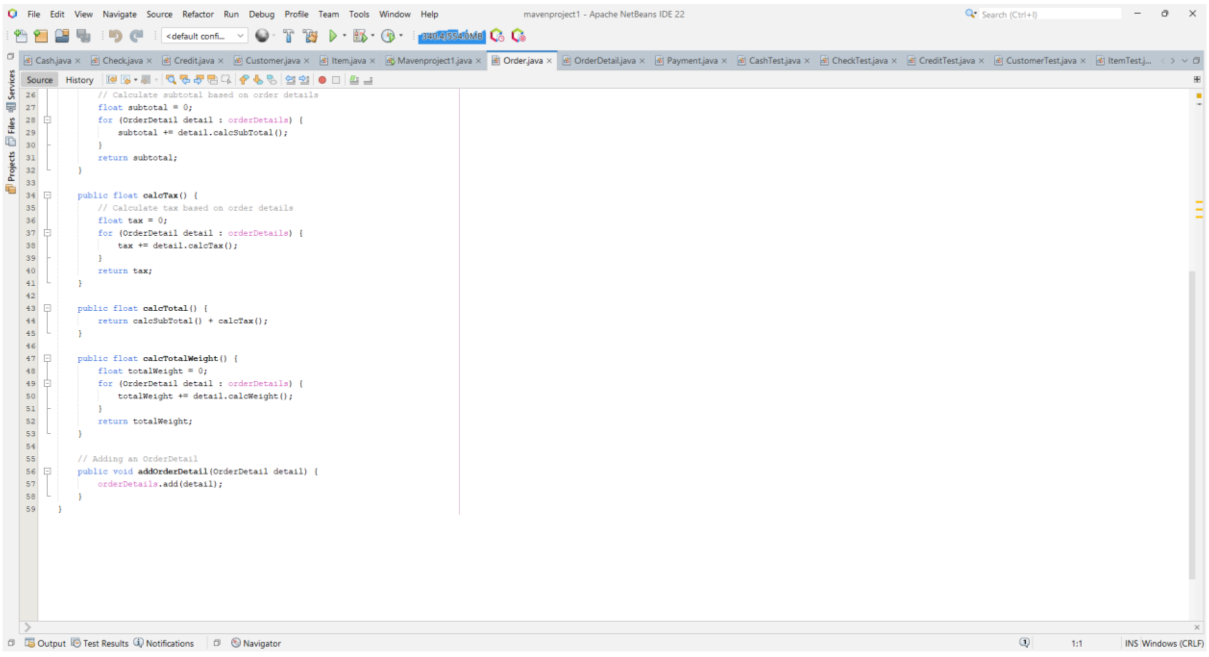
```java
/**
 *
 * @author Charan
 */
public class Item {
    private float shippingWeight;
    private String description;

    // Constructor
    public Item(float shippingWeight, String description) {
        this.shippingWeight = shippingWeight;
        this.description = description;
    }

    public float getPriceForQuantity(int quantity) {
        // Calculate price based on quantity
        return 10.0f * quantity; // Example pricing
    }

    public float getTax() {
        return 1.0f; // Example tax calculation
    }

    public boolean inStock() {
        // Check if item is in stock
        return true;
    }

    public float getShippingWeight() {
        return shippingWeight;
    }
}
```

Mavenproject1.java



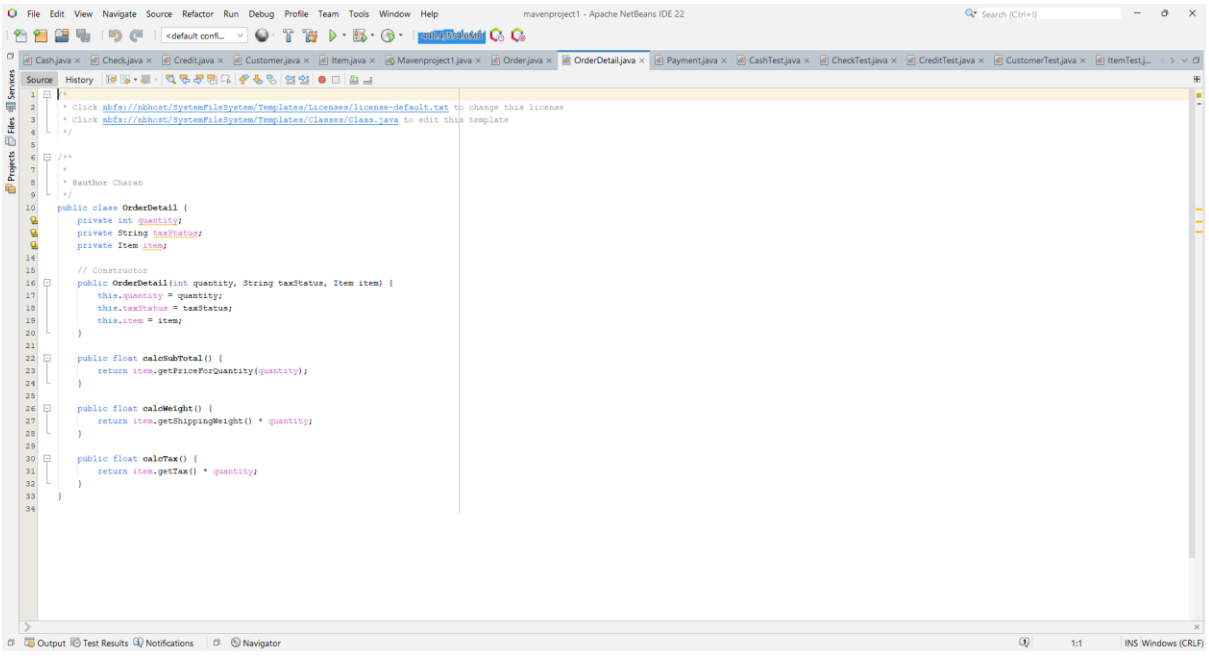
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

/**
 *
 * @author Charan
 */
public class Mavenproject1 {

    public static void main(String[] args) {
        System.out.println("Theary DA Project");
    }
}
```

Order.java



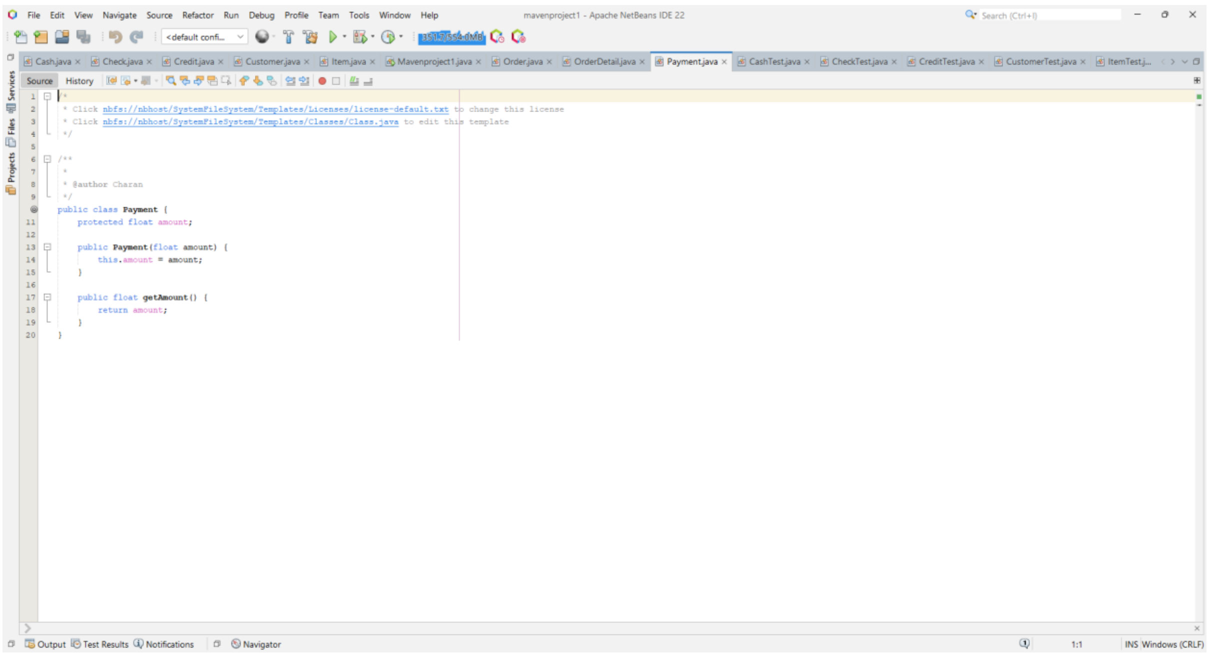
```java
        // Calculate subtotal based on order details
        float subtotal = 0;
        for (OrderDetail detail : orderDetails) {
            subtotal += detail.calcSubTotal();
        }
        return subtotal;
    }

    public float calcTax() {
        // Calculate tax based on order details
        float tax = 0;
        for (OrderDetail detail : orderDetails) {
            tax += detail.calcTax();
        }
        return tax;
    }

    public float calcTotal() {
        return calcSubTotal() + calcTax();
    }

    public float calcTotalWeight() {
        float totalWeight = 0;
        for (OrderDetail detail : orderDetails) {
            totalWeight += detail.calcWeight();
        }
        return totalWeight;
    }

    // Adding an OrderDetail
    public void addOrderDetail(OrderDetail detail) {
        orderDetails.add(detail);
    }
}
```

OrderDetail.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author Charan
 */
public class OrderDetail {
    private int quantity;
    private String taxStatus;
    private Item item;

    // Constructor
    public OrderDetail(int quantity, String taxStatus, Item item) {
        this.quantity = quantity;
        this.taxStatus = taxStatus;
        this.item = item;
    }

    public float calcSubTotal() {
        return item.getPriceForQuantity(quantity);
    }

    public float calcWeight() {
        return item.getShippingWeight() * quantity;
    }

    public float calcTax() {
        return item.getTax() * quantity;
    }
}
```

Payment.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author Charan
 */
public class Payment {
    protected float amount;

    public Payment(float amount) {
        this.amount = amount;
    }

    public float getAmount() {
        return amount;
    }
}
```

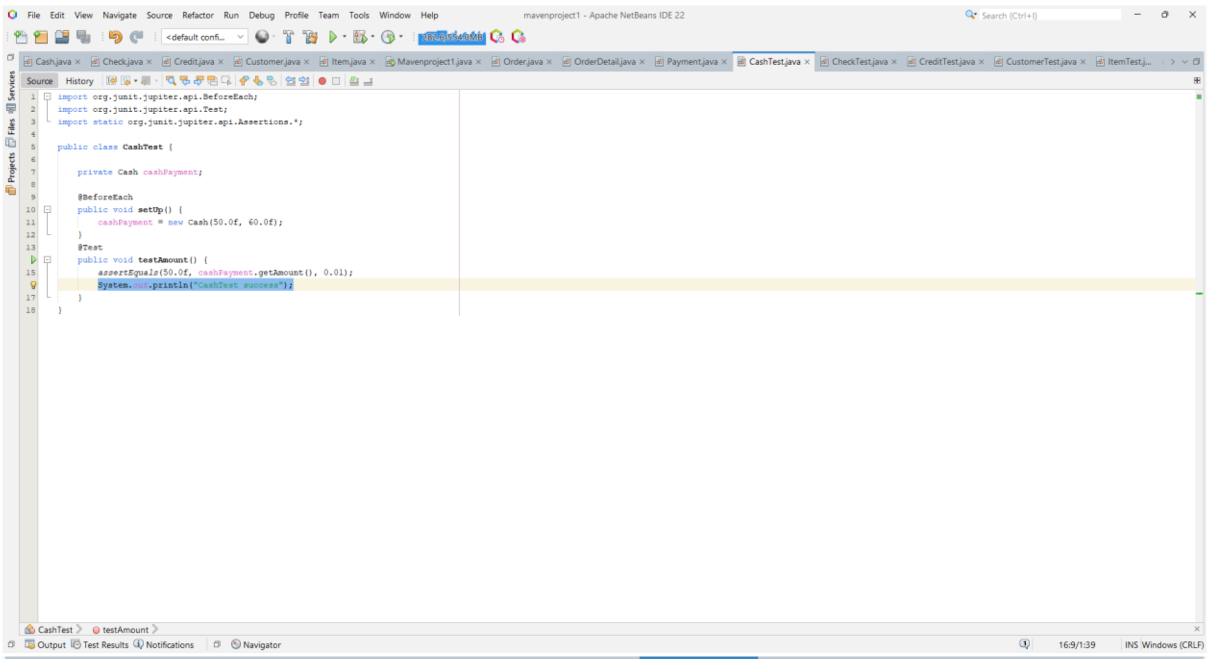
- ✓ **<u>Tool Used:</u>**

We have used JUnit for automated testing, which is one of the most popular testing frameworks in Java for automated testing. It allows a developer to write and run repeatable tests so that functionality in code can be verified. Consequently, you will be able to develop unit tests that will determine how separate components, like classes and methods, behave in isolation; hence, helping you ensure your code works as you expect before it is shipped.

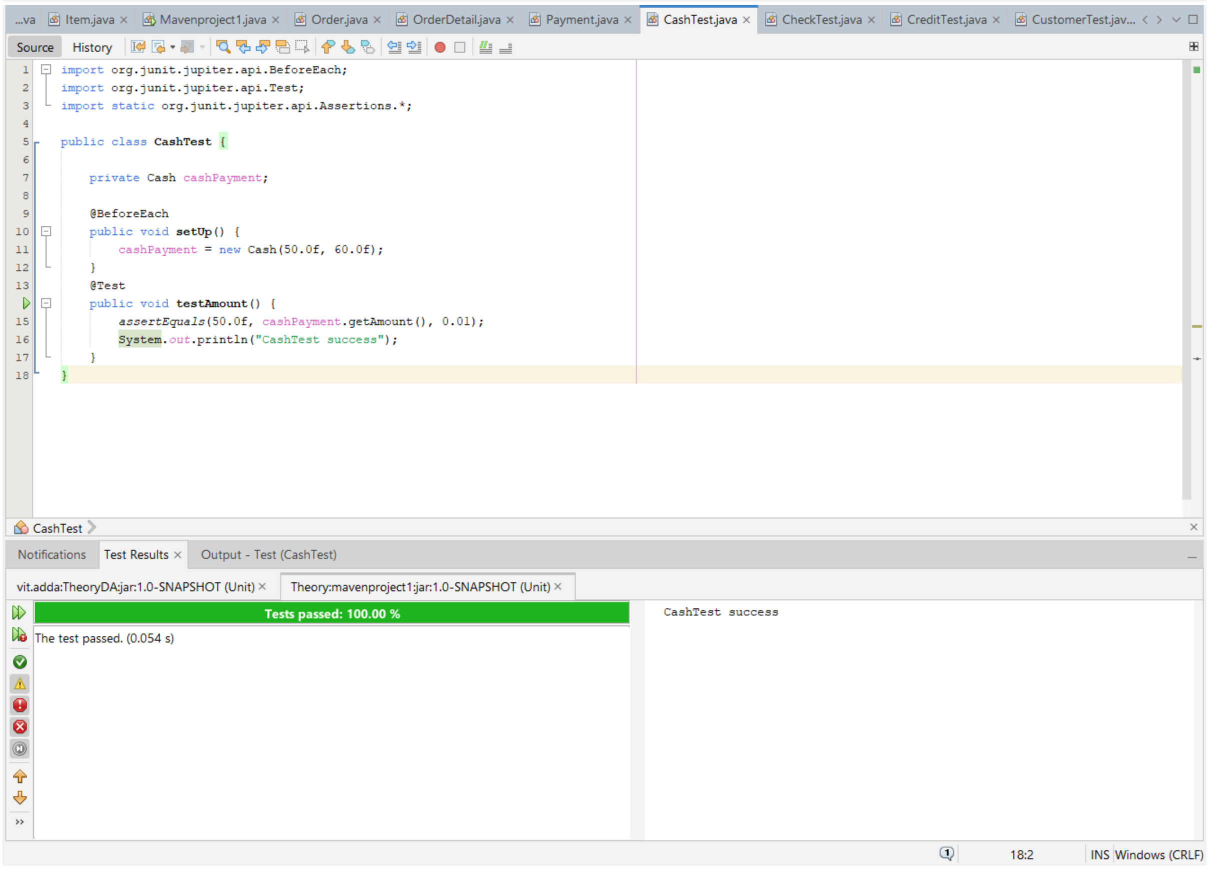
These are the test-codes for automated testing of each class of the application, each test can be run multiple times with different inputs to check different conditions. It very easily integrates with Maven and Gradle, making it a powerful test automation tool for continuous integration in Java development.

✓ **Test Packages:**

CashTest.java



```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CashTest {

    private Cash cashPayment;

    @BeforeEach
    public void setUp() {
        cashPayment = new Cash(50.0f, 60.0f);
    }
    @Test
    public void testAmount() {
        assertEquals(50.0f, cashPayment.getAmount(), 0.01);
        System.out.println("CashTest success");
    }
}
```



```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CashTest {

    private Cash cashPayment;

    @BeforeEach
    public void setUp() {
        cashPayment = new Cash(50.0f, 60.0f);
    }
    @Test
    public void testAmount() {
        assertEquals(50.0f, cashPayment.getAmount(), 0.01);
        System.out.println("CashTest success");
    }
}
```

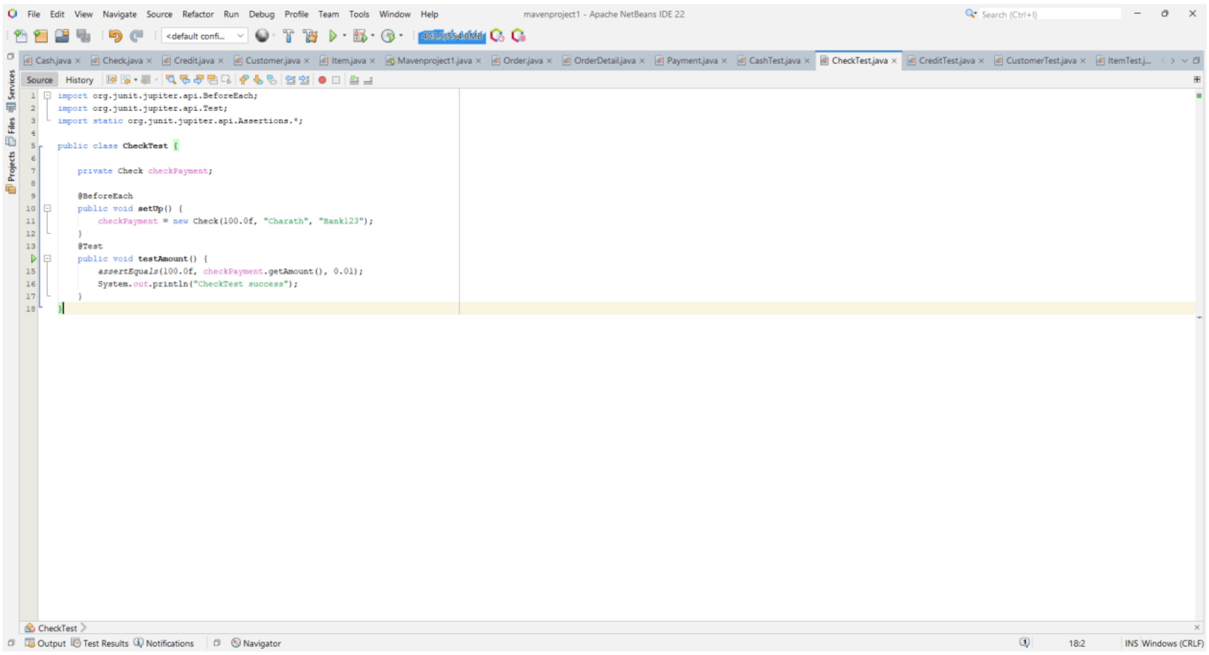
Theory:mavenproject1:jar:1.0-SNAPSHOT (Unit)

Tests passed: 100.00 %

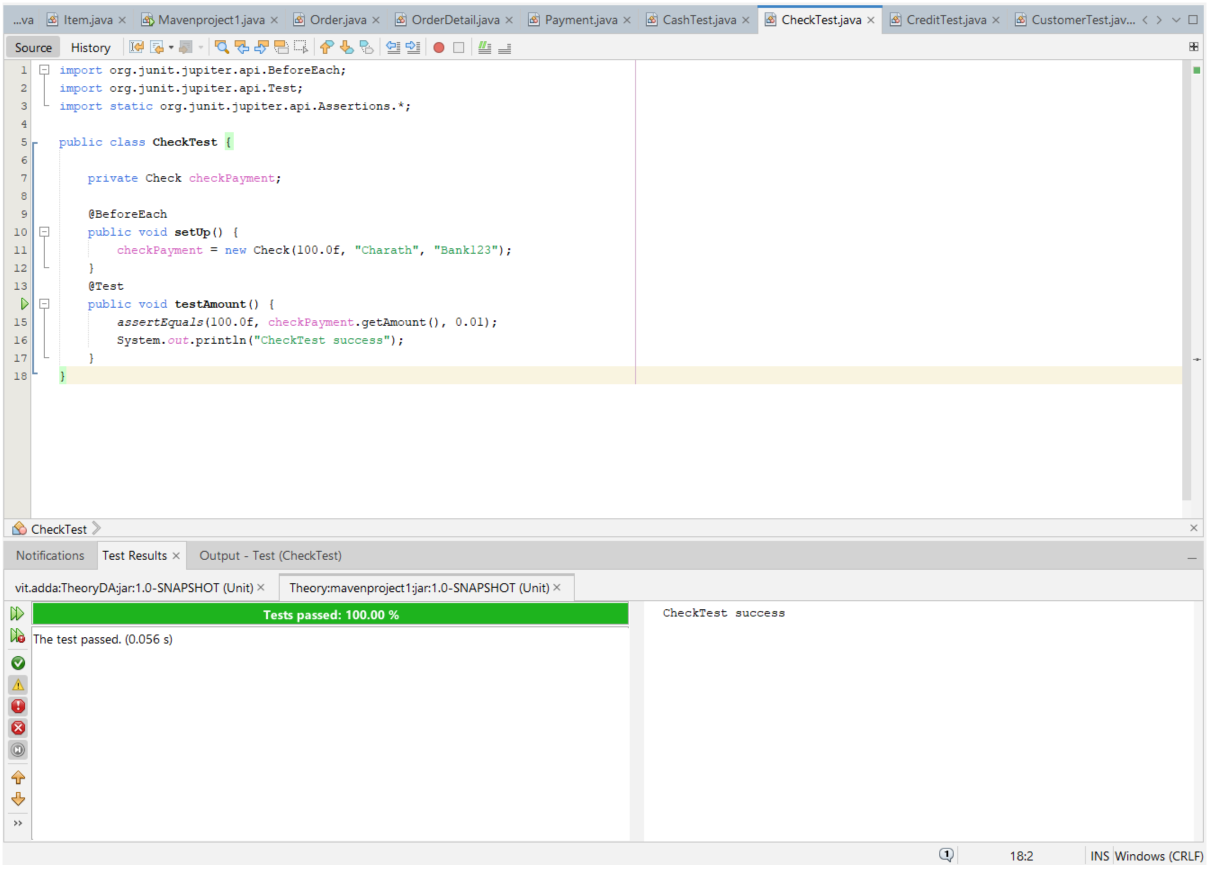
The test passed. (0.054 s)

```
CashTest success
```

CheckTest.java



```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CheckTest {

    private Check checkPayment;

    @BeforeEach
    public void setUp() {
        checkPayment = new Check(100.0f, "Charath", "Bank123");
    }
    @Test
    public void testAmount() {
        assertEquals(100.0f, checkPayment.getAmount(), 0.01);
        System.out.println("CheckTest success");
    }
}
```

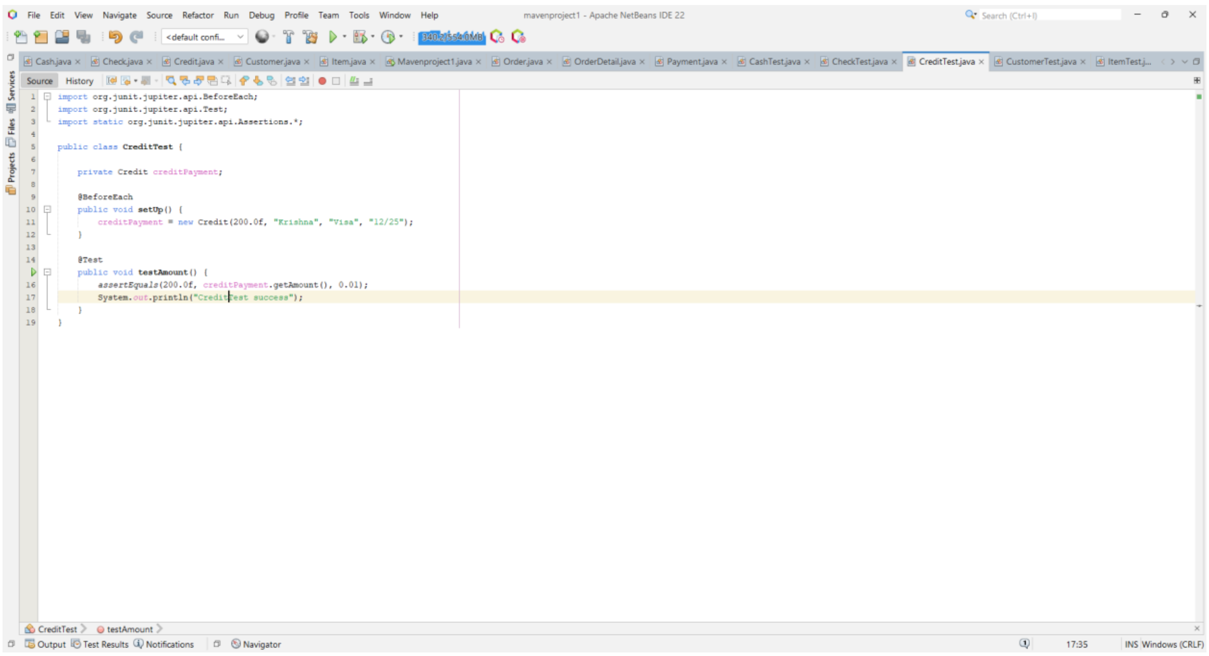


```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CheckTest {

    private Check checkPayment;

    @BeforeEach
    public void setUp() {
        checkPayment = new Check(100.0f, "Charath", "Bank123");
    }
    @Test
    public void testAmount() {
        assertEquals(100.0f, checkPayment.getAmount(), 0.01);
        System.out.println("CheckTest success");
    }
}
```

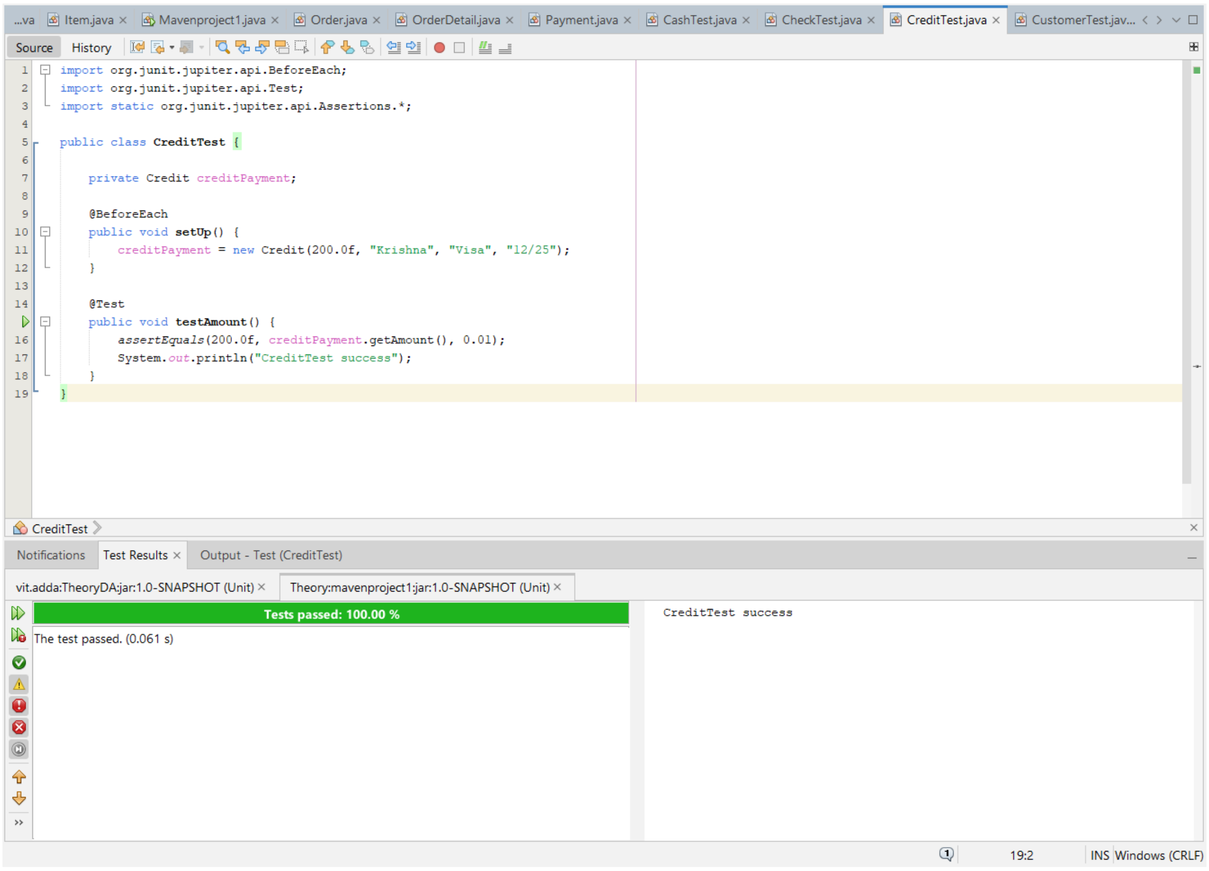
```
Tests passed: 100.00 %
The test passed. (0.056 s)
```

```
CheckTest success
```

CreditTest.java


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
mavenproject1 - Apache NetBeans IDE 22
Cash.java Check.java Credit.java Customer.java Item.java Mavenproject1.java Order.java OrderDetail.java Payment.java CashTest.java CheckTest.java CreditTest.java CustomerTest.java ItemTest.j...
Source History
CreditTest testAmount
Output Test Results Notifications Navigator
17:35 INS Windows (CRLF)

```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CreditTest {

    private Credit creditPayment;

    @BeforeEach
    public void setUp() {
        creditPayment = new Credit(200.0f, "Krishna", "Visa", "12/25");
    }

    @Test
    public void testAmount() {
        assertEquals(200.0f, creditPayment.getAmount(), 0.01);
        System.out.println("CreditTest success");
    }
}
```

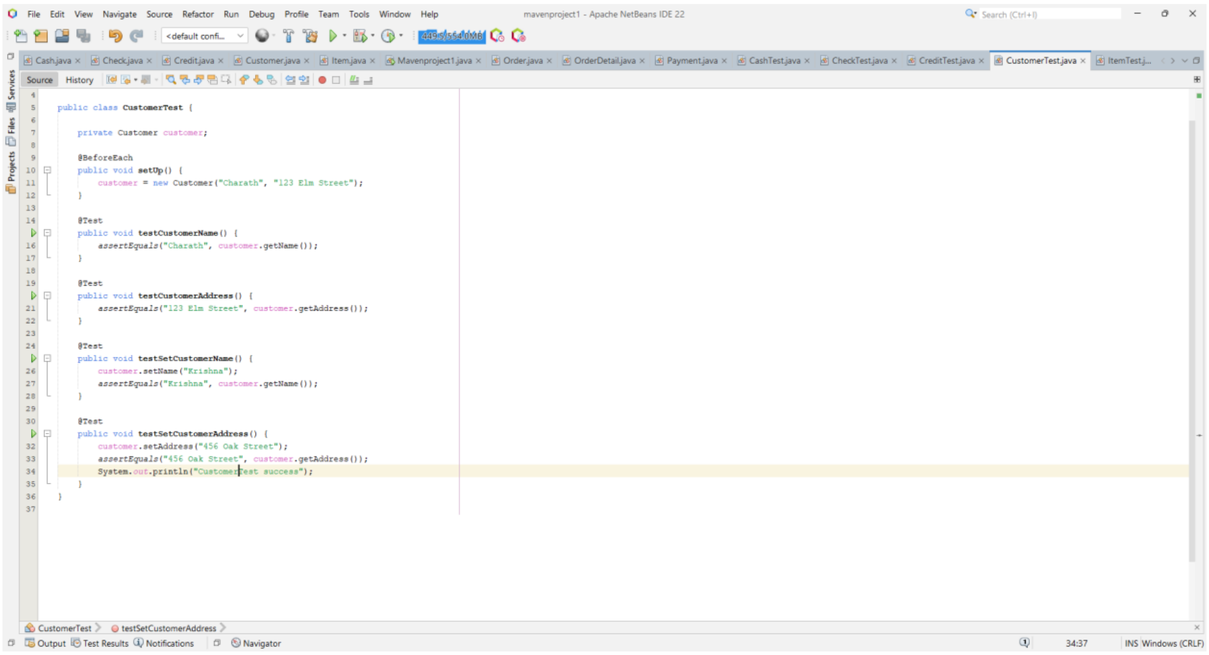

...va Item.java Mavenproject1.java Order.java OrderDetail.java Payment.java CashTest.java CheckTest.java CreditTest.java CustomerTest.jav...
Source History
CreditTest
Notifications Test Results Output - Test (CreditTest)
vit.adda:TheoryDA:jar:1.0-SNAPSHOT (Unit) Theory:mavenproject1:jar:1.0-SNAPSHOT (Unit)
19:2 INS Windows (CRLF)

```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CreditTest {

    private Credit creditPayment;

    @BeforeEach
    public void setUp() {
        creditPayment = new Credit(200.0f, "Krishna", "Visa", "12/25");
    }

    @Test
    public void testAmount() {
        assertEquals(200.0f, creditPayment.getAmount(), 0.01);
        System.out.println("CreditTest success");
    }
}
```

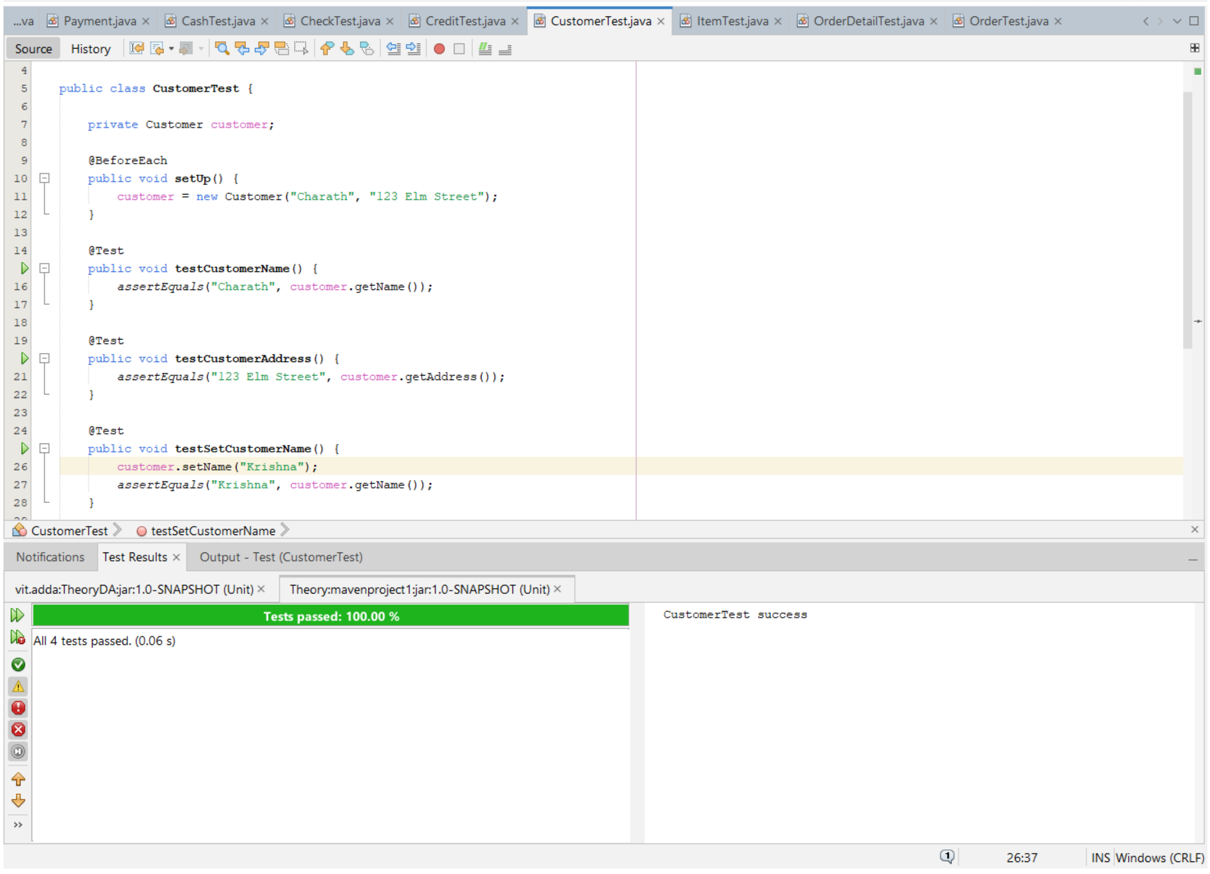
```
Tests passed: 100.00 %
The test passed. (0.061 s)
```

```
CreditTest success
```

CustomerTest.java



```java
public class CustomerTest {

    private Customer customer;

    @BeforeEach
    public void setUp() {
        customer = new Customer("Charath", "123 Elm Street");
    }

    @Test
    public void testCustomerName() {
        assertEquals("Charath", customer.getName());
    }

    @Test
    public void testCustomerAddress() {
        assertEquals("123 Elm Street", customer.getAddress());
    }

    @Test
    public void testSetCustomerName() {
        customer.setName("Krishna");
        assertEquals("Krishna", customer.getName());
    }

    @Test
    public void testSetCustomerAddress() {
        customer.setAddress("456 Oak Street");
        assertEquals("456 Oak Street", customer.getAddress());
        System.out.println("CustomerTest success");
    }
}
```



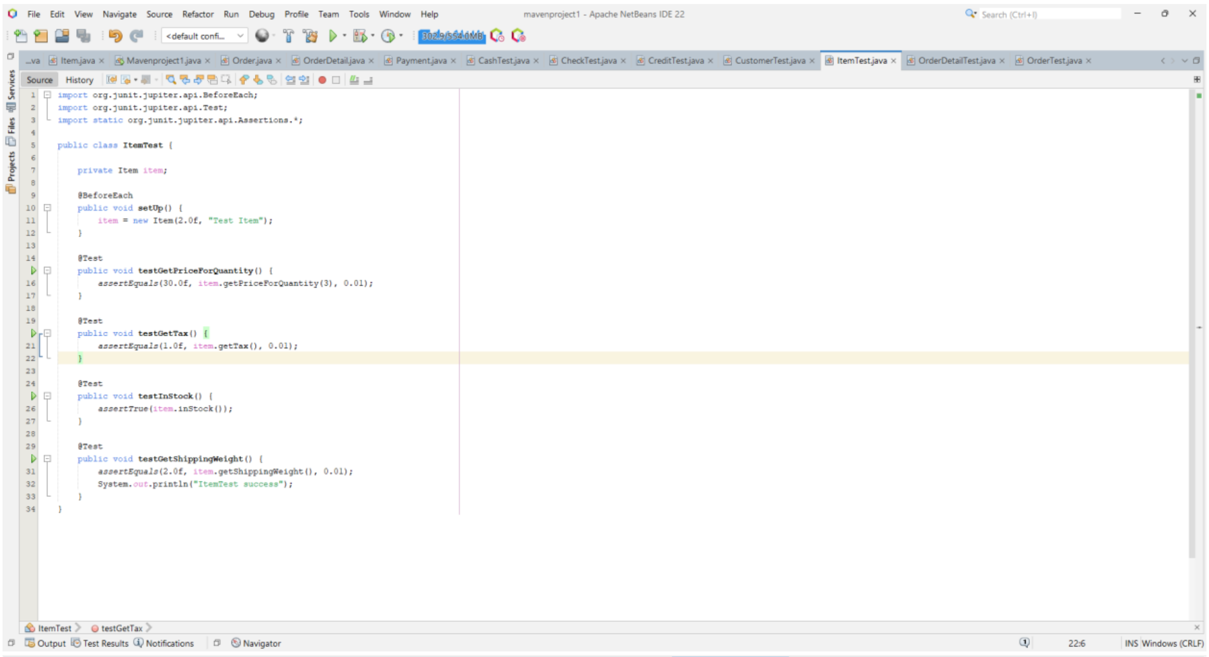
```java
public class CustomerTest {

    private Customer customer;

    @BeforeEach
    public void setUp() {
        customer = new Customer("Charath", "123 Elm Street");
    }

    @Test
    public void testCustomerName() {
        assertEquals("Charath", customer.getName());
    }

    @Test
    public void testCustomerAddress() {
        assertEquals("123 Elm Street", customer.getAddress());
    }

    @Test
    public void testSetCustomerName() {
        customer.setName("Krishna");
        assertEquals("Krishna", customer.getName());
    }
```

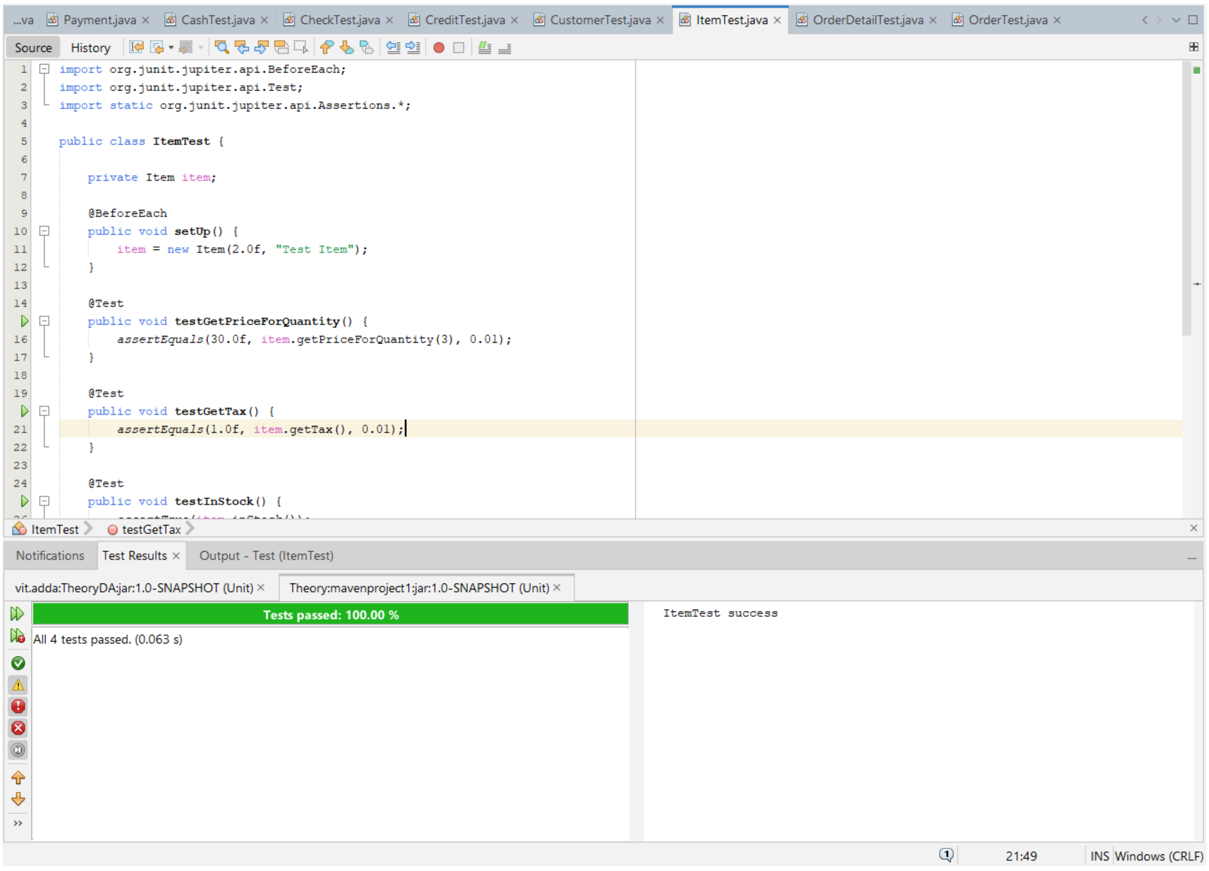
Tests passed: 100.00 %

All 4 tests passed. (0.06 s)

```
CustomerTest success
```

ItemTest.java


```
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class ItemTest {

    private Item item;

    @BeforeEach
    public void setUp() {
        item = new Item(2.0f, "Test Item");
    }

    @Test
    public void testGetPriceForQuantity() {
        assertEquals(30.0f, item.getPriceForQuantity(3), 0.01);
    }

    @Test
    public void testGetTax() {
        assertEquals(1.0f, item.getTax(), 0.01);
    }

    @Test
    public void testInStock() {
        assertTrue(item.inStock());
    }

    @Test
    public void testGetShippingWeight() {
        assertEquals(2.0f, item.getShippingWeight(), 0.01);
        System.out.println("ItemTest success");
    }
}
```



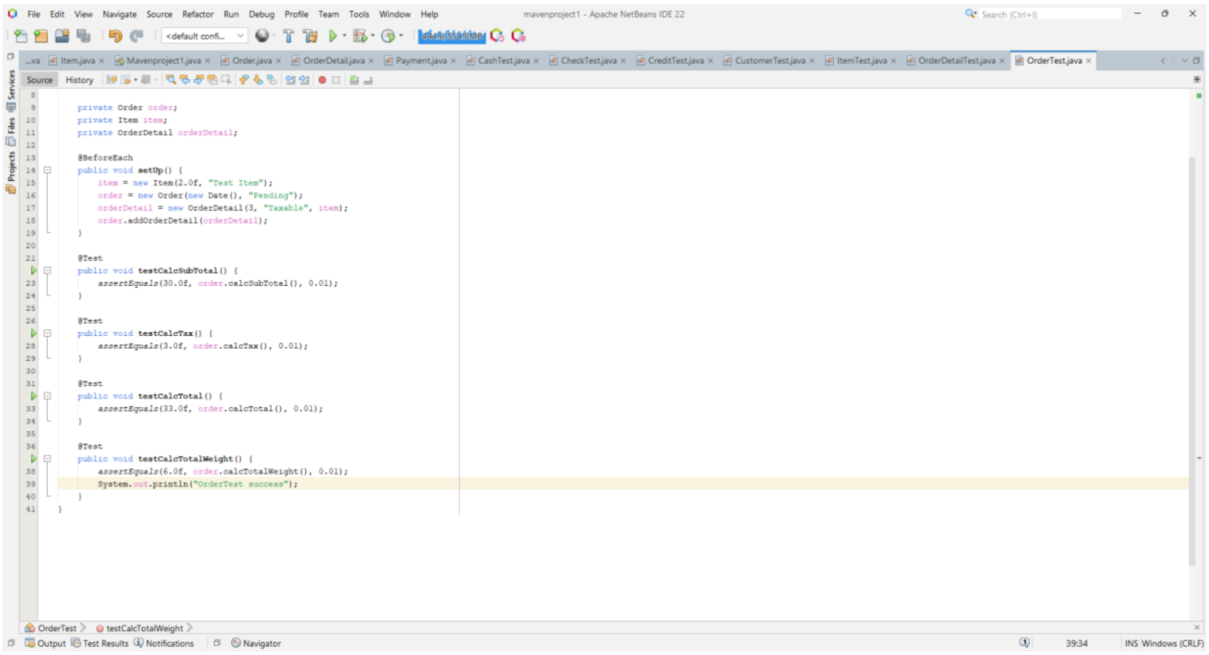
```
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class ItemTest {

    private Item item;

    @BeforeEach
    public void setUp() {
        item = new Item(2.0f, "Test Item");
    }

    @Test
    public void testGetPriceForQuantity() {
        assertEquals(30.0f, item.getPriceForQuantity(3), 0.01);
    }

    @Test
    public void testGetTax() {
        assertEquals(1.0f, item.getTax(), 0.01);
    }

    @Test
    public void testInStock() {
```

Test Results — Theory:mavenproject1:jar:1.0-SNAPSHOT (Unit)

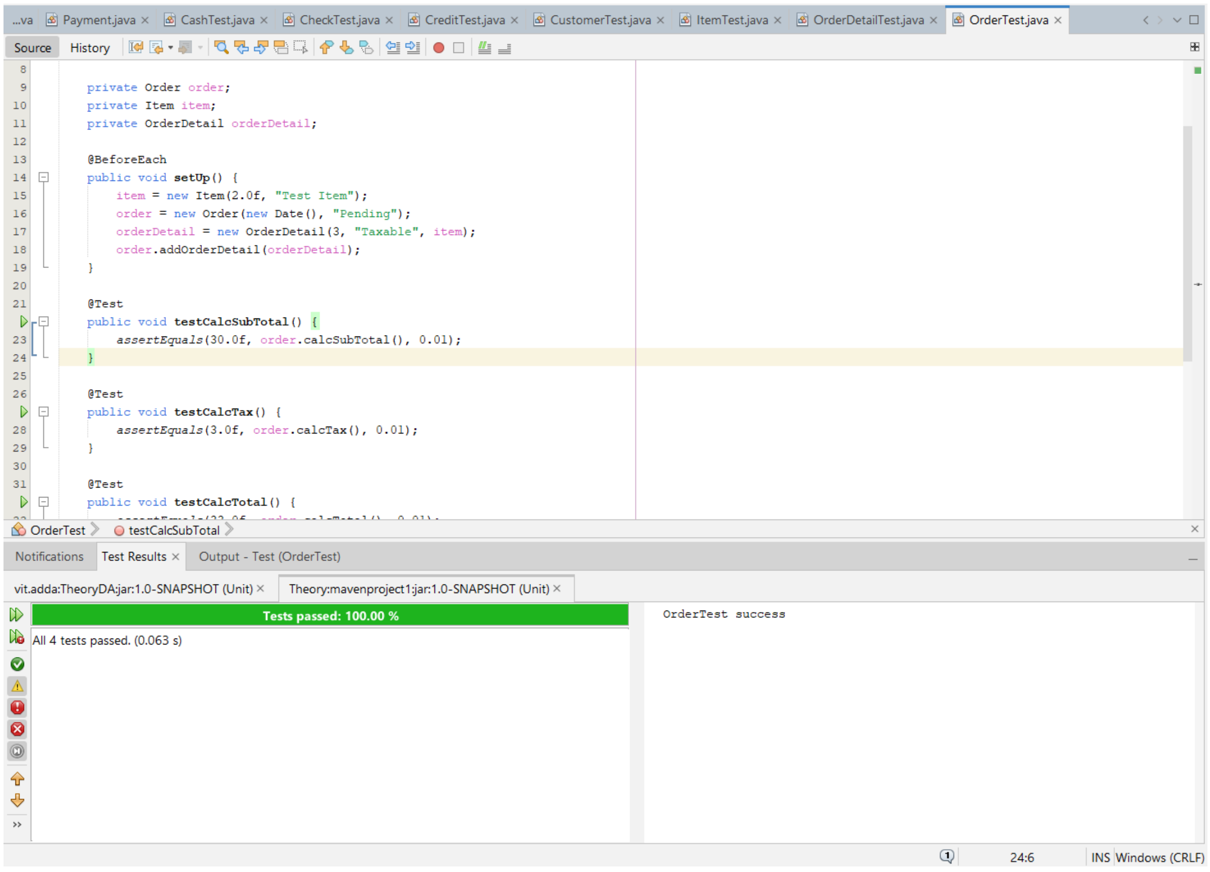
```
Tests passed: 100.00 %
All 4 tests passed. (0.063 s)
```

```
ItemTest success
```

OrderTest.java



```java

    private Order order;
    private Item item;
    private OrderDetail orderDetail;

    @BeforeEach
    public void setUp() {
        item = new Item(2.0f, "Test Item");
        order = new Order(new Date(), "Pending");
        orderDetail = new OrderDetail(3, "Taxable", item);
        order.addOrderDetail(orderDetail);
    }

    @Test
    public void testCalcSubTotal() {
        assertEquals(30.0f, order.calcSubTotal(), 0.01);
    }

    @Test
    public void testCalcTax() {
        assertEquals(3.0f, order.calcTax(), 0.01);
    }

    @Test
    public void testCalcTotal() {
        assertEquals(33.0f, order.calcTotal(), 0.01);
    }

    @Test
    public void testCalcTotalWeight() {
        assertEquals(6.0f, order.calcTotalWeight(), 0.01);
        System.out.println("OrderTest success");
    }
}
```



```java

    private Order order;
    private Item item;
    private OrderDetail orderDetail;

    @BeforeEach
    public void setUp() {
        item = new Item(2.0f, "Test Item");
        order = new Order(new Date(), "Pending");
        orderDetail = new OrderDetail(3, "Taxable", item);
        order.addOrderDetail(orderDetail);
    }

    @Test
    public void testCalcSubTotal() {
        assertEquals(30.0f, order.calcSubTotal(), 0.01);
    }

    @Test
    public void testCalcTax() {
        assertEquals(3.0f, order.calcTax(), 0.01);
    }

    @Test
    public void testCalcTotal() {
```

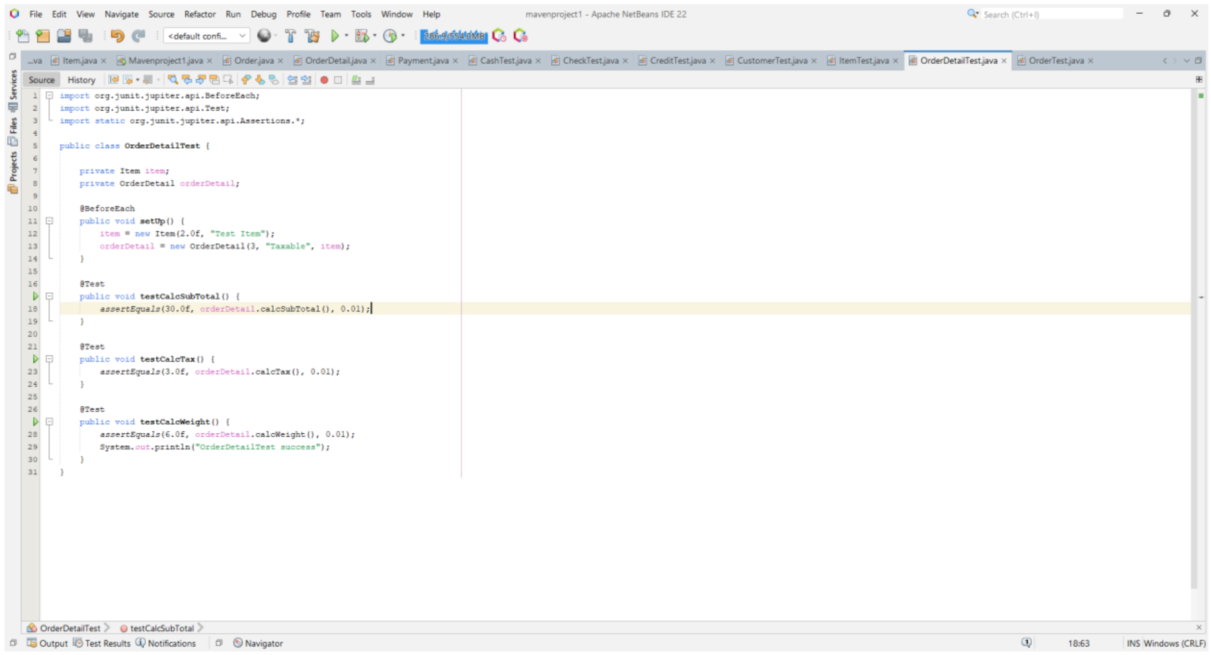
Test Results — Theory:mavenproject1:jar:1.0-SNAPSHOT (Unit)

Tests passed: 100.00 %

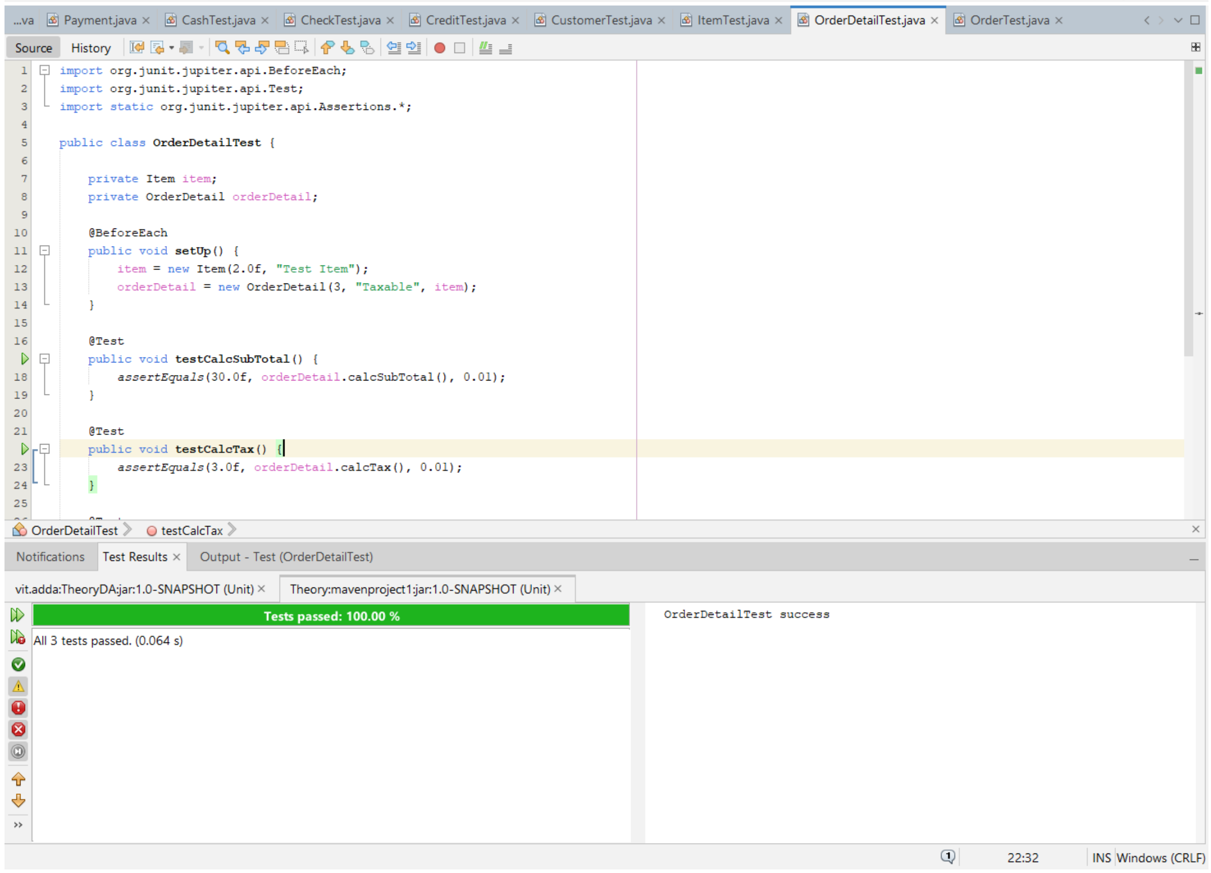
All 4 tests passed. (0.063 s)

```
OrderTest success
```

OrderDetailTest.java



```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class OrderDetailTest {

    private Item item;
    private OrderDetail orderDetail;

    @BeforeEach
    public void setUp() {
        item = new Item(2.0f, "Test Item");
        orderDetail = new OrderDetail(3, "Taxable", item);
    }

    @Test
    public void testCalcSubTotal() {
        assertEquals(30.0f, orderDetail.calcSubTotal(), 0.01);
    }

    @Test
    public void testCalcTax() {
        assertEquals(3.0f, orderDetail.calcTax(), 0.01);
    }

    @Test
    public void testCalcWeight() {
        assertEquals(6.0f, orderDetail.calcWeight(), 0.01);
        System.out.println("OrderDetailTest success");
    }
}
```



```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class OrderDetailTest {

    private Item item;
    private OrderDetail orderDetail;

    @BeforeEach
    public void setUp() {
        item = new Item(2.0f, "Test Item");
        orderDetail = new OrderDetail(3, "Taxable", item);
    }

    @Test
    public void testCalcSubTotal() {
        assertEquals(30.0f, orderDetail.calcSubTotal(), 0.01);
    }

    @Test
    public void testCalcTax() {
        assertEquals(3.0f, orderDetail.calcTax(), 0.01);
    }
```

```
Tests passed: 100.00 %
All 3 tests passed. (0.064 s)
```

```
OrderDetailTest success
```

✓ **<u>Git Hub Link (for all source and junit test codes of maven project):</u>**

https://github.com/Sree-Krishna04/ADDA_TH_DA.git