JUnit and Mockito Equivalent for NUnit and Moq (Java Version)

1. Calculator and Unit Test (JUnit equivalent)

Calculator.java

```
package calc;

public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
}
```

CalculatorTest.java

```
package calc;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

class CalculatorTest {

    private Calculator calculator;

    @BeforeEach
    void setup() {
        calculator = new Calculator();
    }

    @AfterEach
    void teardown() {
        calculator = null;
    }

    @Test
    @DisplayName("Add: Valid inputs")
    void testAdd() {
        assertEquals(5, calculator.add(2, 3));
        assertEquals(0, calculator.add(-1, 1));
        assertEquals(0, calculator.add(0, 0));
        assertEquals(300, calculator.add(100, 200));
    }

    @Disabled("Subtraction not implemented yet")
    @Test
    void testSubtract() {
        // Placeholder
    }
}
```

2. Mocking Example using Mockito

IMailSender.java

```
package customer;

public interface IMailSender {
    boolean sendMail(String toAddress, String message);
}
```

MailSender.java

```
package customer;

import javax.mail.*;
import javax.mail.internet.*;
import java.util.Properties;

public class MailSender implements IMailSender {

    @Override
    public boolean sendMail(String toAddress, String messageText) {
        final String from = "your_email@gmail.com";
        final String host = "smtp.gmail.com";
        final String username = "your_email@gmail.com";
        final String password = "your_password";

        Properties props = new Properties();
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", host);
        props.put("mail.smtp.port", "587");

        Session session = Session.getInstance(props,
            new Authenticator() {
                protected PasswordAuthentication getPasswordAuthentication() {
                    return new PasswordAuthentication(username, password);
                }
            });

        try {
            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress(from));
            message.setRecipients(
                Message.RecipientType.TO, InternetAddress.parse(toAddress));
            message.setSubject("Test Mail");
            message.setText(messageText);
            Transport.send(message);
            return true;
        } catch (MessagingException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

CustomerComm.java

```
package customer;

public class CustomerComm {
    private final IMailSender mailSender;

    public CustomerComm(IMailSender mailSender) {
        this.mailSender = mailSender;
    }

    public boolean sendMailToCustomer() {
        String email = "cust123@abc.com";
        String message = "Some Message";
        return mailSender.sendMail(email, message);
    }
}
```

MainApp.java

```
package customer;

public class MainApp {
    public static void main(String[] args) {
        System.out.println("Sending mail to customer...");

        IMailSender sender = new MailSender();
        CustomerComm comm = new CustomerComm(sender);

        boolean result = comm.sendMailToCustomer();
        System.out.println(result ? "Mail sent successfully." : "Failed to send mail.");
    }
}
```

CustomerCommTest.java

```
package customer;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;
import static org.mockito.Mockito.*;

class CustomerCommTest {
    private IMailSender mockSender;
    private CustomerComm customerComm;

    @BeforeEach
    void setup() {
        mockSender = mock(IMailSender.class);
        when(mockSender.sendMail(anyString(), anyString())).thenReturn(true);

        customerComm = new CustomerComm(mockSender);
    }

    @Test
    void testSendMailToCustomer_ReturnsTrue() {
        assertTrue(customerComm.sendMailToCustomer());
    }
}
```

Maven Dependencies (pom.xml)

pom.xml

```
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.10.0</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>5.11.0</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.sun.mail</groupId>
        <artifactId>javax.mail</artifactId>
        <version>1.6.2</version>
    </dependency>
</dependencies>
```