**Exercise 5:** Mocking and Stubbing with Multiple Returns

**Scenario:** You need to test a service that depends on an external API with multiple return values.

**Ex5MultiReturnApi.java**

package demo;

public interface Ex5MultiReturnApi {

    String fetchData();

}

**Ex5MultiReturn.java**

package demo;

public class Ex5MultiReturn {

    private Ex5MultiReturnApi api;

    public Ex5MultiReturn(Ex5MultiReturnApi api) {

        this.api = api;

    }

    public String getNextData() {

        return api.fetchData();

    }

}

**Ex5MultiReturnTest.java**

package demo;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class Ex5MultiReturnTest {

    @Test

    public void testMultipleReturns() {

        Ex5MultiReturnApi mockApi = mock(Ex5MultiReturnApi.class);

        when(mockApi.fetchData())

            .thenReturn("First Response")

            .thenReturn("Second Response")

            .thenReturn("Third Response");

        Ex5MultiReturn service = new Ex5MultiReturn(mockApi);

        String result1 = service.getNextData();

        String result2 = service.getNextData();

        String result3 = service.getNextData();

        assert result1.equals("First Response");

        assert result2.equals("Second Response");

        assert result3.equals("Third Response");

        verify(mockApi, times(3)).fetchData();

    }

}

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>demo</artifactId>

  <version>0.0.1-SNAPSHOT</version>

  <name>Ex1\_Mocking\_Stubbing</name>

  <properties>

        <maven.compiler.source>17</maven.compiler.source> <!-- Java version -->

        <maven.compiler.target>17</maven.compiler.target>

        <junit.jupiter.version>5.10.0</junit.jupiter.version>

        <mockito.version>5.8.0</mockito.version>

        <bytebuddy.version>1.14.14</bytebuddy.version>

    </properties>

    <dependencies>

        <!-- JUnit 5 -->

        <dependency>

            <groupId>org.junit.jupiter</groupId>

            <artifactId>junit-jupiter</artifactId>

            <version>${junit.jupiter.version}</version>

            <scope>test</scope>

        </dependency>

        <!-- Mockito Core -->

        <dependency>

            <groupId>org.mockito</groupId>

            <artifactId>mockito-core</artifactId>

            <version>${mockito.version}</version>

            <scope>test</scope>

        </dependency>

        <!-- Mockito JUnit Jupiter Integration -->

        <dependency>

            <groupId>org.mockito</groupId>

            <artifactId>mockito-junit-jupiter</artifactId>

            <version>${mockito.version}</version>

            <scope>test</scope>

        </dependency>

        <!-- Byte Buddy -->

        <dependency>

            <groupId>net.bytebuddy</groupId>

            <artifactId>byte-buddy</artifactId>

            <version>${bytebuddy.version}</version>

            <scope>test</scope>

        </dependency>

        <!-- Byte Buddy Agent -->

        <dependency>

            <groupId>net.bytebuddy</groupId>

            <artifactId>byte-buddy-agent</artifactId>

            <version>${bytebuddy.version}</version>

            <scope>test</scope>

        </dependency>

        <!-- AssertJ for fluent assertions -->

        <dependency>

            <groupId>org.assertj</groupId>

            <artifactId>assertj-core</artifactId>

            <version>3.24.2</version>

            <scope>test</scope>

        </dependency>

        <!-- Hamcrest for matchers -->

        <dependency>

            <groupId>org.hamcrest</groupId>

            <artifactId>hamcrest-library</artifactId>

            <version>2.2</version>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Maven Surefire Plugin for running tests -->

            <plugin>

                <groupId>org.apache.maven.plugins</groupId>

                <artifactId>maven-surefire-plugin</artifactId>

                <version>3.1.2</version>

                <configuration>

                <argLine>-Dnet.bytebuddy.experimental=true</argLine>

            </configuration>

            </plugin>

        </plugins>

    </build>

</project>

**Output:**

