Name of the Course : Ultimate Java Masterclass

Level : Easy

Tool Stack : Java8 and Junit5

Problem Statement : Provide a code solution to do sum of cubes and squares of elements in an array

Description : Get an integer array as input and identify even and odd numbers. If number is odd get cube of it, if number is even get square of it. Finally add all cubes and squares together and return it as output.

Create two classes, one NumberList class with number[] field and with a parameterized constructor and another MainClass with two static methods.

1. public static Boolean addSquareCube (NumberList numberList), which accepts NumberList object and returns sum of cubes and squares.
2. Public static void main method, for reading the number from input devices and call the addSquareCube method to add cubes and squares.

Code:

**package** yaksha;

**public** **class** NumberList {

**private** **int** number[];

**public** **int**[] getNumber() {

**return** number;

}

**public** **void** setNumber(**int**[] number) {

**this**.number = number;

}

**public** NumberList(**int**[] number) {

**super**();

**this**.number = number;

}

}

**package** yaksha;

**import** java.util.\*;

**public** **class** MainClass {

**static** **int** addSquareCube(NumberList numberList) {

**int** length = numberList.getNumber().length;

**int** sumeven = 0, sumodd = 0;

**int** number[] = numberList.getNumber();

**for** (**int** i = 0; i < length; i++) {

**if** (number[i] % 2 == 0) {

sumeven = sumeven + (number[i] \* number[i]);

} **else** {

sumodd = sumodd + (number[i] \* number[i] \* number[i]);

}

}

**return** sumeven + sumodd;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter size of array");

**int** size = scanner.nextInt();

System.***out***.println("Enter numbers");

**int** number[] = **new** **int**[size];

**for** (**int** i = 0; i < size; i++) {

number[i] = scanner.nextInt();

}

NumberList numberList = **new** NumberList(number);

**int** sumSquareCube = *addSquareCube*(numberList);

System.***out***.print("sum of cubes and squares of elements in an array : ");

System.***out***.println(sumSquareCube);

scanner.close();

}

}

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>iiht.yaksha.eqtwo</groupId>

<artifactId>JavaMasterClassEq2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>JavaMasterClassEq2</name>

<description>JavaMasterClassEq2</description>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>${maven.compiler.source}</maven.compiler.target>

<junit.jupiter.version>5.5.2</junit.jupiter.version>

<junit.platform.version>1.5.2</junit.platform.version>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.12</version>

<scope>provided</scope>

</dependency>

<dependency>

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

<artifactId>junit-jupiter-engine</artifactId>

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

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.platform</groupId>

<artifactId>junit-platform-runner</artifactId>

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

</plugin>

<plugin>

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

<version>2.22.2</version>

</plugin>

</plugins>

</build>

</project>

Junit Testing

**package** yaksha;

**import** java.io.File;

**import** java.io.FileWriter;

**import** java.io.IOException;

// boiler-plate code

**public** **class** TestUtils {

**public** **static** File *businessTestFile*;

**public** **static** File *boundaryTestFile*;

**public** **static** File *exceptionTestFile*;

**static** {

*businessTestFile* = **new** File("./output\_revised.txt");

*businessTestFile*.delete();

*boundaryTestFile* = **new** File("./output\_boundary\_revised.txt");

*boundaryTestFile*.delete();

*exceptionTestFile* = **new** File("./output\_exception\_revised.txt");

*exceptionTestFile*.delete();

}

**public** **static** **void** yakshaAssert(String testName, Object result, File file) **throws** IOException {

System.***out***.println("\n" + testName + "=" + result);

FileWriter writer = **new** FileWriter(file, **true**);

writer.append("\n" + testName + "=" + result);

writer.flush();

writer.close();

}

**public** **static** String currentTest() {

**return** Thread.*currentThread*().getStackTrace()[2].getMethodName();

}

}

**package** yaksha;

**import** **static** yaksha.TestUtils.*businessTestFile*;

**import** **static** yaksha.TestUtils.*currentTest*;

**import** **static** yaksha.TestUtils.*yakshaAssert*;

**import** org.junit.jupiter.api.Test;

**class** MainClassTest {

@Test

**void** testAddSquareCube() **throws** Exception {

// Test will pass

**int** numberArray1[] = **new** **int**[] { 2, 6, 3, 4, 5 };

NumberList numberList1 = **new** NumberList(numberArray1);

*yakshaAssert*(*currentTest*(), (MainClass.*addSquareCube*(numberList1) == 208 ? "true" : "false"), *businessTestFile*);

}

}

output\_revised.txt

testAddSquareCube=true

testing-JavaMasterClassEq2.xml

<test-cases>

<cases xsi:type="java:com.assessment.data.TestCase">

<test-case-type>Functional</test-case-type>

<expected-ouput>true</expected-ouput>

<name>testAddSquareCube</name>

<weight>6</weight>

<mandatory>true</mandatory>

<desc>Testing sum of sqaure and cube of numbers in an array</desc>

</cases>

</test-cases>

Test Data1

Enter size of array

5

Enter numbers

2

6

3

4

5

sum of cubes and squares of elements in an array : 208

Learning outcome: Participant would be able to know the use of array with control statement.