Name of the Course : Ultimate Java Masterclass

Level : Easy

Tool Stack : Java8 and Junit5

Problem Statement : Provide a code solution to check sum of odd and even digits.

Description : Create two classes, one Number class with number field and with a parameterized constructor and another MainClass with four static methods

1. public static int checkSumOfOddEvenDigits(int number), which accepts an integer and returns 0 or 1. If sum of digits is odd it returns 1 otherwise if sum of digits is even then it returns 0.
2. public static int SumOfOddDigits(int number), which accepts an integer and returns 1 if sum of odd digits is odd.
3. public static int SumOfEvenDigits(int number), which accepts an integer and returns 1 if sum of even digits is even.

2. public static void main method, for reading the number from input devices and call the checkSumOfOddEvenDigits method to test it.

Code:

**package** yaksha;

**public** **class** Number {

**private** **int** number;

**public** **int** getNumber() {

**return** number;

}

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

**this**.number = number;

}

**public** Number(**int** number) {

**super**();

**this**.number = number;

}

}

**package** yaksha;

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

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

**static** **int** *sum* = 0;

**static** **int** *result* = 0;

**public** **static** **int** checkSumOfOddEvenDigits(**int** number) {

**while** (number != 0) {

**int** remainder = number % 10;

*sum* = *sum* + remainder;

number = number / 10;

}

**if** (*sum* % 2 != 0)

**return** 1;

**else**

**return** 0;

}

**public** **static** **int** SumOfOddDigits(**int** number) {

**if** (*sum* % 2 != 0) {

*sum* = 0;

**while** (number != 0) {

**int** remainder = number % 10;

**if** (remainder % 2 != 0) {

*sum* = *sum* + remainder;

}

number = number / 10;

}

**if** (*sum* % 2 != 0) {

*result* = 1;

}

}

**return** *result*;

}

**public** **static** **int** SumOfEvenDigits(**int** number) {

**if** (*sum* % 2 == 0) {

*sum* = 0;

**while** (number != 0) {

**int** remainder = number % 10;

**if** (remainder % 2 == 0) {

*sum* = *sum* + remainder;

}

number = number / 10;

}

**if** (*sum* % 2 == 0) {

*result* = 1;

}

}

**return** *result*;

}

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

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

**int** oddResult = 0;

**int** evenResult = 0;

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

System.***out***.println("Enter a number");

**int** number1 = scanner.nextInt();

Number number = **new** Number(number1);

**if** (*checkSumOfOddEvenDigits*(number.getNumber()) == 1)

oddResult = *SumOfOddDigits*(number.getNumber());

**else**

evenResult = *SumOfEvenDigits*(number.getNumber());

**if** (oddResult == 1)

System.***out***.println("Sum of odd digits is odd");

**if** (evenResult == 1)

System.***out***.println("Sum of even digits is even");

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.eqone</groupId>

<artifactId>JavaMasterClassEq1</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>JavaMasterClassEq1</name>

<description>JavaMasterClassEq1</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 org.junit.jupiter.api.Assertions.\*;

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

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

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

//import org.junit.jupiter.api.Assertions;

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

**class** MainClassTest {

@Test

**void** testCheckSumOfOddEvenDigits() **throws** Exception {

// Test will pass

Number number1 = **new** Number(56895);

*yakshaAssert*(*currentTest*(), (MainClass.*checkSumOfOddEvenDigits*(number1.getNumber()) == 1 ? "true" : "false"),

*businessTestFile*);

}

@Test

**void** testSumOfOddDigits() **throws** Exception {

// Test will pass

Number number1 = **new** Number(56895);

*yakshaAssert*(*currentTest*(), (MainClass.*SumOfOddDigits*(number1.getNumber()) == 1 ? "true" : "false"),

*businessTestFile*);

}

@Test

**void** testSumOfEvenDigits() **throws** Exception {

// Test will pass

Number number2 = **new** Number(56896);

*yakshaAssert*(*currentTest*(), (MainClass.*SumOfEvenDigits*(number2.getNumber()) == 1 ? "true" : "false"),

*businessTestFile*);

}

}

output\_revised.txt

testSumOfEvenDigits=true

testSumOfOddDigits=true

testCheckSumOfOddEvenDigits=true

testing-JavaMasterClassEq1.xml

<test-cases>

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

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

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

<name>testCheckSumOfOddEvenDigits</name>

<weight>2</weight>

<mandatory>true</mandatory>

<desc>Testing sum of number is even or odd</desc>

</cases>

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

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

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

<name>testSumOfOddDigits</name>

<weight>2</weight>

<mandatory>true</mandatory>

<desc>Testing sum of odd digits</desc>

</cases>

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

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

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

<name>testSumOfEvenDigits</name>

<weight>2</weight>

<mandatory>true</mandatory>

<desc>Testing sum of even digits</desc>

</cases>

</test-cases>

Test Data1

Enter a number

56895

Sum of odd digits is odd

Test Data2

Enter a number

56896

Sum of even digits is even

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