Name of the Course : Complete Java SE8 Developer Bootcamp

Level : Easy

Tool Stack : Java11 and Junit5

Problem Statement : Provide a code solution to check Kaprekar Number.

Description : Create three classes one Number class with number field with a parameterized constructor and one Kaprekar Number class with one static method

1. public static int checkKaprekarNumber(Number number), which accepts Number object and returns 1 or 0.

Hint:  A positive whole number ‘n’ that has ‘d’ number of digits is squared and split into two pieces, a right-hand piece that has ‘d’ digits and a left-hand piece that has remaining ‘d’ or ‘d-1’ digits. If the sum of the two pieces is equal to the number, then ‘n’ is a Kaprekar number.

For example if input-number is 45 then

45^2 = 2025, right-hand piece of 2025 = 25 and left hand piece of 2025 = 20

Sum = 25 + 20 = 45, i.e. equal to the number. Hence, 45 is a Kaprekar number."

and one MainClass with one static method

1. public static void main method, for reading the Number from input devices and call the checkKaprekarNumber method to test it.

Code:

**package** yaksha;

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

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

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

**super**();

**this**.number = number;

}

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

**return** number;

}

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

**this**.number = number;

}

}

**package** yaksha;

**public** **class** KaprekarNumber {

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

**int** count = 0, j = 0;

**int** a = number.getNumber();

**int** a1 = a;

**while** (a1 != 0) {

count = count + 1;

a1 = a1 / 10;

}

**int** square = a \* a;

String s = Integer.*toString*(square);

String s1 = s.substring(0, count);

String s2 = s.substring(count);

**int** x = Integer.*parseInt*(s1);

**int** y = Integer.*parseInt*(s2);

**int** result = x + y;

**if** (result == a) {

j = 1;

} **else** {

j = 0;

}

**return** j;

}

}

**package** yaksha;

**import** java.util.Scanner;

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

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

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

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

**int** n = sc.nextInt();

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

**int** result = KaprekarNumber.*checkKaprekarNumber*(number);

**if** (result == 1) {

System.***out***.println("Kaprekar Number");

} **else** {

System.***out***.println("Not a Kaprekar Number");

}

sc.close();

}

}

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** testCheckKaprekarNumber() **throws** Exception {

// Test will pass

Number number1 = **new** Number(45);

*yakshaAssert*(*currentTest*(), (KaprekarNumber.*checkKaprekarNumber*(number1) == 1 ? "true" : "false"),

*businessTestFile*);

}

}

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.kaprekar</groupId>

<artifactId>KaprekarNumberJ11EQ2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>KaprekarNumberJ11EQ2</name>

<description>KaprekarNumberJ11EQ2</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>

output\_revised.txt

testCheckKaprekarNumber=true

testing-testCheckKaprekarNumber.xml

<test-cases>

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

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

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

<name>testCheckKaprekarNumber</name>

<weight>10</weight>

<mandatory>true</mandatory>

<desc>Test to check Kaprekar Number</desc>

</cases>

</test-cases>

Test Data1

Enter a number

45

Kaprekar Number

Learning outcome: Participant could able to learn how to use control statements.