# 19.LeapYearCheckProgram(White-BoxTesting)

Aim:

Todevelopaprogramthatcheckswhetheragivenyearisaleapyearand verifytheoutput using white-box testing with JUnit.

Algorithm:

1. **Step1:**Accepta yearasinput.
2. **Step 2:**Checkiftheyear is divisibleby4.
   * Iftheyearisdivisibleby4, proceedtothenext step.
   * Ifnot divisible by4, itisnot a leapyear.
3. **Step 3:**Checkiftheyear is divisibleby100.
   * Ifdivisibleby100, checkifitis divisibleby400.
     + Ifdivisibleby400, it is aleapyear.
     + Ifnot divisibleby400, itisnot aleapyear.
4. **Step4:**Iftheyearis divisibleby4but notby100,it isaleapyear.
5. **Step5:**Returnwhethertheyearisaleapyearornot.
6. **Step6:**WriteJUnittestcasestovalidatetheleapyearcheck,includingedgecases like 0, negative years, and century years.

Code:

Step1:

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

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

public class LeapYearCheckerTest {

private LeapYearChecker checker;

@BeforeEach

public void setUp() {

checker = new LeapYearChecker();

}

@Test

public void testLeapYear2020() {

assertTrue(checker.isLeapYear(2020)); // Test Case 1

}

@Test

public void testLeapYear2000() {

assertTrue(checker.isLeapYear(2000)); // Test Case 2

}

@Test

public void testNonLeapYear2021() {

assertFalse(checker.isLeapYear(2021)); // Test Case 3

}

@Test

public void testNonLeapYear1900() {

assertFalse(checker.isLeapYear(1900)); // Test Case 4

}

@Test

public void testLeapYear1600() {

assertTrue(checker.isLeapYear(1600)); // Test Case 5

}

@Test

public void testNegativeYear() {

IllegalArgumentException thrown = assertThrows(

IllegalArgumentException.class,

() -> checker.isLeapYear(-400) // Test Case 6

);

assertEquals("Year must be a non-negative integer.", thrown.getMessage());

}

}

**Step2:**

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

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

public class LeapYearCheckerTest {

private LeapYearChecker checker;

@BeforeEach

public void setUp() {

checker = new LeapYearChecker();

}

@Test

public void testLeapYear2020() {

assertTrue(checker.isLeapYear(2020)); // Test Case 1

}

@Test

public void testLeapYear2000() {

assertTrue(checker.isLeapYear(2000)); // Test Case 2

}

@Test

public void testNonLeapYear2021() {

assertFalse(checker.isLeapYear(2021)); // Test Case 3

}

@Test

public void testNonLeapYear1900() {

assertFalse(checker.isLeapYear(1900)); // Test Case 4

}

@Test

public void testLeapYear1600() {

assertTrue(checker.isLeapYear(1600)); // Test Case 5

}

@Test

public void testNegativeYear() {

IllegalArgumentException thrown = assertThrows(

IllegalArgumentException.class,

() -> checker.isLeapYear(-400) // Test Case 6

);

assertEquals("Year must be a non-negative integer.", thrown.getMessage());

}

}

Sample Input:

# TestCase1:

* + Input:2020
  + ExpectedOutput:
    - true(2020isaleapyear)

# TestCase2:

* + Input:2000
  + ExpectedOutput:
    - true(2000isaleapyear)

# TestCase3:

* + Input:2021
  + ExpectedOutput:
    - false(2021is not aleapyear)

# TestCase4:

* + Input:1900
  + ExpectedOutput:
    - false(1900is not aleapyear, asit isdivisible by100 butnot400)

# TestCase5:

* + Input:1600
  + ExpectedOutput:
    - true(1600isaleapyear, divisible by400)

# TestCase6:

* + Input:-400(Invalidinput)
  + ExpectedOutput:-ThrowsIllegalArgumentExceptionwiththemessage:"Year cannot be negative."

# TestCase7:

* + Input:0
  + ExpectedOutput:
    - true(Year0isconsideredaleapyearbydefinition)

SampleOutput:

# TestCase1:

* + Input:2020
  + Output:true

# TestCase2:

* + Input:2000
  + Output:true

# TestCase3:

* + Input:2021
  + Output:false

# TestCase4:

* + Input:1900
  + Output:false

# TestCase5:

* + Input:1600
  + Output:true

# TestCase6:

* + Input:-400
  + Output: ThrowsIllegalArgumentException

# TestCase7:

* + Input:0
  + Output:true

Results:

* **TestCase1:**Theprogramcorrectlyidentifiesthat2020is aleapyear.
* **TestCase2:**Theprogramcorrectlyidentifiesthat2000isaleapyear(sinceitis divisible by 400).
* **TestCase3:**Theprogramcorrectlyidentifiesthat2021is nota leapyear.
* **TestCase4:**Theprogramcorrectlyidentifiesthat1900isnotaleapyear(sinceitis divisible by 100 but not by 400).
* **TestCase5:**Theprogramcorrectlyidentifiesthat1600isaleapyear(sinceitis divisible by 400).
* **TestCase6:**TheprogramcorrectlythrowsanIllegalArgumentExceptionfora negative year input.
* **TestCase7:**Theprogramcorrectlyidentifiesthat0isaleapyear(since0 isdivisible by 4 and 400).



