**15.PalindromeCheckProgram(White-BoxTesting)**

Aim:

TodevelopaJavaprogramthatcheckswhetheragivennumberisapalindromeandvalidate the output using white-box testing with JUnit.

Algorithm:

1. **Step1:**Accepttheinput number(integer).
2. **Step 2:**Reversethedigits ofthenumber.
3. **Step3:**Comparethereversed numberwiththeoriginalnumber.
4. **Step4:**Iftheoriginalnumberisequaltothereversednumber,itisapalindrome; otherwise, it is not.
5. **Step5:**WriteJUnittestcasestoverifythecorrectnessofthepalindromelogic, ensuring all possible edge cases are tested.

**Code:**

**Step 1 : PalindromeChecker.java**

public class PalindromeChecker {

public boolean isPalindrome(int number) {

// Negative numbers are not palindromes

if (number < 0) return false;

int original = number;

int reversed = 0;

while (number != 0) {

int digit = number % 10;

reversed = reversed \* 10 + digit;

number /= 10;

}

return original == reversed;

}

}

**Step 2: PalindromeCheckerTest.java**

**package** ex6.Palindrome;

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

**importstatic** org.junit.jupiter.api.Assertions.\*;

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

**public class** PalindromeCheckerTest {

**private** PalindromeChecker checker;

@BeforeEach

**publicvoid** setUp() {

checker = **new** PalindromeChecker();

}

@Test

**publicvoid** testPalindrome121() {

*assertTrue*(checker.isPalindrome(121)); // Test Case 1

}

@Test

**publicvoid** testNotPalindrome123() {

*assertFalse*(checker.isPalindrome(123)); // Test Case 2

}

@Test

**publicvoid** testPalindrome555() {

*assertTrue*(checker.isPalindrome(555)); // Test Case 3

}

@Test

**publicvoid** testPalindromeZero() {

*assertTrue*(checker.isPalindrome(0)); // Test Case 4

}

@Test

**publicvoid** testNegativeNumber() {

*assertFalse*(checker.isPalindrome(-121)); // Test Case 5

}

@Test

**publicvoid** testLargePalindrome() {

*assertTrue*(checker.isPalindrome(1234321)); // Test Case 6

}

@Test

**publicvoid** testLargeNonPalindrome() {

*assertFalse*(checker.isPalindrome(987654)); // Test Case 7

}

}

Sample Input:

# TestCase1:

* + - Input: 121
    - ExpectedOutput:true(Palindrome)

# TestCase2:

* + - Input: 123
    - ExpectedOutput:false(Notapalindrome)

# TestCase3:

* + - Input: 555
    - ExpectedOutput:true(Palindrome)

# TestCase4:

* + - Input:0
    - ExpectedOutput: true(Palindrome,edge case)

# TestCase5:

* + - Input:-121
    - ExpectedOutput:false(Negativenumberisnotapalindrome)

# TestCase6:

* + - Input:1234321
    - ExpectedOutput:true(Largepalindromenumber)

# TestCase7:

* + - Input:987654
    - ExpectedOutput:false(Largenon-palindromenumber) Sample Output:

# TestCase1:

* + - Input: 121
    - Output:true

# TestCase2:

* + - Input: 123
    - Output:false

# TestCase3:

* + - Input: 555
    - Output:true

# TestCase4:

* + - Input:0
    - Output:true

# TestCase5:

* + - Input:-121
    - Output:false

# TestCase6:

* + - Input:1234321
    - Output:true

# TestCase7:

* + - Input:987654
    - Output:false

Results:

* + **TestCase1:**Thenumber 121is correctlyidentifiedas apalindrome.
  + **TestCase2:**Thenumber123iscorrectlyidentified asnot a palindrome.
  + **TestCase3:**Thenumber 555is correctlyidentifiedas apalindrome.
  + **TestCase4:**Thenumber0 iscorrectlyidentifiedas apalindrome.
  + **TestCase5:**Thenegativenumber-121iscorrectlyidentifiedasnotapalindrome (edge case).
  + **TestCase6:**Thelargenumber1234321is correctlyidentifiedasapalindrome.
  + **TestCase7:**Thelargenumber987654 is correctlyidentifiedas not apalindrome.



