calculateScoreForCategory

A picture containing diagram, sketch, drawing, line

Description automatically generated

Edge-pair coverage:

TR: {[1,2,3], [1,2,4], [1,2,5], [1,2,6], [1,2,7], [1,2,8], [1,2,9], [1,2,10], [1,2,11], [1,2,12], [1,2,13], [1,2,14], [1,2,15], [1,2,16], [2,3,17], [2,4,17], [2,5,17], [2,6,17], [2,7,17], [2,8,17], [2,9,17], [2,10,17], [2,11,17], [2,12,17], [2,13,17], [2,14,17], [2,15,17], [2,16,17]}

TP: {[1,2,3,17], [1,2,4,17], [1,2,5,17], [1,2,6,17] , [1,2,7,17], [1,2,8,17], , [1,2,9,17], [1,2,10,17], [1,2,11,17], [1,2,12,17], [1,2,13,17], [1,2,14,17], [1,2,15,17], [1,2,16,17], }

DU coverage

|  |  |  |
| --- | --- | --- |
| Variable | Du-pair | Du path |
| score | (1, 17)  (3, 17)  (4, 17)  (5, 17)  (6, 17)  (7, 17)  (8, 17)  (9, 17)  (10, 17)  (11, 17)  (12, 17)  (13, 17)  (14, 17)  (15, 17) | [1, 2, 16, 17]  [3, 17]  [4, 17]  [5, 17]  [6, 17]  [7, 17]  [8, 17]  [9, 17]  [10, 17]  [11, 17]  [12, 17]  [13, 17]  [14, 17]  [15, 17] |
| category | (1, 2) | [1, 2] |
| diceList | (1, 3)  (1, 4)  (1, 5)  (1, 6)  (1, 7)  (1, 8)  (1, 9)  (1, 10)  (1, 11)  (1, 12)  (1, 13)  (1, 14)  (1, 15) | [1,2,3]  [1,2,4]  [1,2,5]  [1,2,6]  [1,2,7]  [1,2,8]  [1,2,9]  [1,2,10]  [1,2,11]  [1,2,12]  [1,2,13]  [1,2,14]  [1,2,15] |

JUnit test

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

import java.util.ArrayList;

public class ScoreCalculatorTest {

private ArrayList<Dice> diceList;

@BeforeEach

public void setUp() {

diceList = new ArrayList<>();

diceList.add(new Dice(1));

diceList.add(new Dice(2));

diceList.add(new Dice(3));

diceList.add(new Dice(4));

diceList.add(new Dice(5));

}

@Test

public void testOnes() {

int score = ScoreCalculator.calculateScoreForCategory("Ones", diceList);

assertEquals(1, score);

}

@Test

public void testTwos() {

int score = ScoreCalculator.calculateScoreForCategory("Twos", diceList);

assertEquals(2, score);

}

// Similarly, write tests for other categories

@Test

public void testInvalidCategory() {

int score = ScoreCalculator.calculateScoreForCategory("Invalid Category", diceList);

assertEquals(0, score);

}

}

calcNumScore

A picture containing diagram, text, circle, screenshot

Description automatically generated

Edge-pair coverage

TR: {[1,2,3], [1,2,5], [2,3,4], [2,3,2], [3,4,2], [3,2,5], [3,2,3], [4,2,3], [4,2,5]}

TP: {[1,2,5], [1,2,3,4,2,5], [1,2,3,2,5], [1,2,3,2,3,4,2,5]}

Du coverage

|  |  |  |
| --- | --- | --- |
| variable | Du pair | Du path |
| Sum | (1, 4)  (4, 4)  (4, 5) | [1, 2, 3, 4]  [4, 2, 3, 4]  [4, 2, 5] |
| diceList | (1,2) | [1,2] |
| dice | (2,3) | [2, 3] |
| number | (1, 3)  (1, 4) | [1, 2, 3]  [1, 2, 3, 4] |

Junit Test

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

import java.util.ArrayList;

public class ScoreCalculatorTest {

private ArrayList<Dice> diceList;

@BeforeEach

public void setUp() {

diceList = new ArrayList<>();

diceList.add(new Dice(1));

diceList.add(new Dice(2));

diceList.add(new Dice(3));

diceList.add(new Dice(2));

diceList.add(new Dice(5));

}

@Test

public void testCalcNumScoreOnes() {

int sum = ScoreCalculator.calcNumScore(diceList, 1);

assertEquals(1, sum);

}

@Test

public void testCalcNumScoreTwos() {

int sum = ScoreCalculator.calcNumScore(diceList, 2);

assertEquals(4, sum);

}

// You can add more test cases for other numbers if necessary

@Test

public void testCalcNumScoreInvalidNumber() {

int sum = ScoreCalculator.calcNumScore(diceList, 7);

assertEquals(0, sum);

}

}