Assignment 1
Software Enginneering
Tyler Wilding

Source Code

Bowling.java

```java
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
/**
 * Author - Tyler Wilding
 * Program - Bowling Score
 * Description - Keeps track of the players score while playing 10 pin bowling.
 */
public class Bowling {
    //Variables
    public static int[] frameScore;
    public static int[] bonusCount;
    public static int[] pins;
    public static int frameNumber;
    public static int throwCounter;
    public static boolean finalFrame;
    /**
     * Bowling Construction, resets the variables for the bowling object.
     */
    public Bowling() {
        frameScore = new int[10];
        bonusCount = new int[10];
        pins = new int[21];
        throwCounter = 0;
        finalFrame = false;
        frameNumber = 0;
    }
    /**
     * Main method that runs through 10 frame inputs with a scanner.
     * @param args Command line arguments not used.
     */
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        while(frameNumber < 9) { //Loop 9 times for 9 frames, 10th time is outside the loop as it is
slightly different.
            frame(scan.next(), scan.next(), null);
        }
        frame(scan.next(), scan.next(), scan.next());
        System.out.println(totalScore());
    }
    /**
     * Accepts one input for the frame, would be used for a strike for example.
     * @param pinsOne First throw's details.
     */
    public static void bowl(String pinsOne) {
        frame(pinsOne, "0", null);
    }
    /**
     * Accepts two inputs for the frame, can be used for normal hits or a spare.
     * @param pinsOne First throw's details
     * @param pinsTwo Second throw's details
     */
    public static void bowl(String pinsOne, String pinsTwo) {
        frame(pinsOne, pinsTwo, null);
    }
    /**
     * Accepts three inputs for the frame, used for the final frame, where a bonus throw is possible.
     * @param pinsOne First throw's details
     * @param pinsTwo Second throw's details
     * @param pinsThree Third throw's details
     */
    public static void bowl(String pinsOne, String pinsTwo, String pinsThree) {
        frame(pinsOne, pinsTwo, pinsThree);
    }
    /**
     * Processes the frame's input to enter into the datastructures
     * @param pinsOne First throw's details
     * @param pinsTwo Second throw's details
     * @param bonusMove Bonus throw's details
     */
    public static void frame(String pinsOne, String pinsTwo, String bonusMove) {
        //Variable is used to keep track of the previous shot's pin count, calculate for spare.
        int previousShot = 0;
        if(!checkUserInput(pinsOne)) //Verify user input
            return;
        int testKnocked = Integer.parseInt(pinsOne);
        if(!updateScore(testKnocked, 0, throwCounter)) //Update score, will also verify if it was a
valid input.
            return;
        //Update array, print score as of so far
        pins[throwCounter++] = testKnocked;
        previousShot = testKnocked;
        printScore();
        if(frameNumber == 9) //If we are on the last frame, use the marker variable.
            finalFrame = true;
        if(testKnocked == 10 && !finalFrame) { //If we hit a strike not on the final frame, skip the
second input.
            frameNumber++;
            throwCounter++;
            return;
        }
        //Repeat for the second throw
        if(!checkUserInput(pinsTwo))
            return;
        testKnocked = Integer.parseInt(pinsTwo);
        if(!updateScore(testKnocked, previousShot, throwCounter))
            return;
        pins[throwCounter++] = testKnocked;
        previousShot = testKnocked;
        printScore();
        if(finalFrame) { //If we are on the final frame, it is a special case
            if(pins[18] + pins[19] == 10) //If we hit a spare, mark it so.
                bonusCount[9] = 1;
            if(bonusCount[9] == 1 || bonusCount[9] == 2) { //If we hit a spare or a strike, we get an
extra throw.
                if(!checkUserInput(bonusMove))
                    return;
                testKnocked = Integer.parseInt(bonusMove);
                if(!updateScore(testKnocked, 0, throwCounter))
                    return;
                pins[throwCounter++] = testKnocked;
                printScore();
            }
        }
        //Move to the next frame.
        frameNumber++;
    }
    /**
     * Computes the total score from all of the frames.
     * @return The total score as of so far.
     */
    public static int totalScore() {
        int totalScore = 0;
        for(int i = 0; i < frameScore.length; i++) {
            totalScore += frameScore[i];
        }
        return totalScore;
    }
```

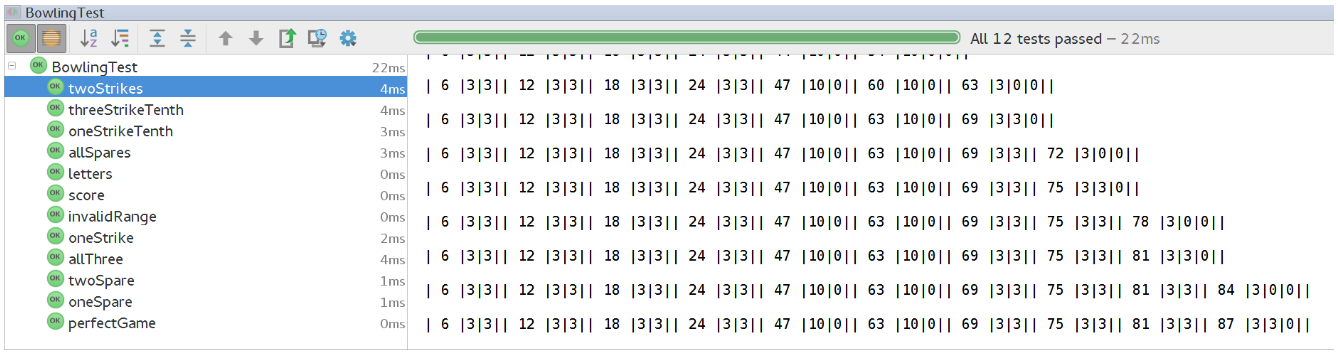
```java
    /**
     * Computes the runningTotal up to the current frame to be output at that frame.
     * @param length How many frames to add up.
     * @return Total up to that frame
     */
    public static int runningTotal(int length) {
        int runningTotal = 0;
        for(int i = 0; i <= length; i++)
            runningTotal += frameScore[i];
        return runningTotal;
    }
    /**
     * Update score method that checks if the throw input is valid, and deals with spares and strikes.
     * @param knocked The amount of pins knocked down.
     * @param previousShot The amount of pins knocked down in the previous shot.
     * @param throwCounter What throw out of 21 is it.
     * @return Will return -1 if error.
     */
    public static boolean updateScore(int knocked, int previousShot, int throwCounter) {
        if(frameNumber != 9 && previousShot + knocked > 10) { //If we are not on the final frame, and 
we hit more than 10, wrong.
            System.out.println("Cant hit more than 10 points in a frame");
            return false;
        }
        if(!finalFrame && knocked == 10) //Strike
            bonusCount[frameNumber] = 2;
        else if(!finalFrame && previousShot + knocked == 10) //Spare
            bonusCount[frameNumber] = 1;
        for(int i = 0; i < frameNumber; i++) { //Iterate through bonus array, consecutive throws will 
add to the previous frames.
            if(bonusCount[i] > 0) {
                bonusCount[i]--;
                frameScore[i] += knocked;
            }
        }
        //Add the pins knocked down to the score card.
        frameScore[frameNumber] += knocked;
        return true;
    }
    /**
     * Print scorecard method, loops through and prints.
     */
    public static void printScore() {
        System.out.println();
        int throwCounter = 0;
        for(int i = 0; i < frameNumber; i++){
            System.out.print("| "+runningTotal(i)+" |"+pins[throwCounter++]+"|"+pins[throwCounter++]
+"|");
        }
        System.out.print("| "+runningTotal(9)+" |"+pins[throwCounter++]+"|"+pins[throwCounter++]
+"|"+pins[throwCounter++]+"||");
        System.out.println();
    }
    /**
     * Check user input to see if it is not a letter or more than 10.
     * @param input Users input as a string.
     * @return True or False
     */
    public static boolean checkUserInput(String input) {
        //Define pattern, everything that is not 0-9
        Pattern p = Pattern.compile("[^0-9]");
        Matcher m = p.matcher(input);
        if(m.find()) { //If the input contains anything besides 0-9, false.
            System.out.println("Only enter numeric values!");
            return false;
        }
        //If the input is a number but less than 0 or greater than 10, remove.
        if(Integer.parseInt(input) > 10 || Integer.parseInt(input) < 0) {
            System.out.println("Only accepts values between 0 and 10, inclusive!");
            return false;
        }
        return true;
    }
}
```

BowlingTest.java

```java
import org.junit.Test;
import static org.junit.Assert.*;
/**
 * Author - Tyler Wilding
 * Description - Test class for bowling program.
 */
public class BowlingTest {
    @Test
    public void invalidRange() throws Exception {
        Bowling test = new Bowling();
        System.out.println("Testing Range!");
        assertEquals(false, test.checkUserInput("15"));
        assertEquals(false, test.checkUserInput("-15"));
        System.out.println();
    }
    @Test
    public void letters() throws Exception {
        Bowling test = new Bowling();
        System.out.println("Testing letter inputs!");
        assertEquals(false, test.checkUserInput("a"));
        assertEquals(false, test.checkUserInput("1a"));
        assertEquals(false, test.checkUserInput("a1"));
        System.out.println();
    }
    @Test
    public void score() throws Exception {
        Bowling test = new Bowling();
        System.out.println("Testing Score");
        assertEquals(true, test.updateScore(5, 0, 1));
        assertEquals(false, test.updateScore(6, 5, 1));
    }
    @Test
    public void perfectGame() throws Exception {
        System.out.println("Perfect Score");
        Bowling test = new Bowling();
        for(int i = 0; i < 9; i++)
            test.bowl("10");
        test.bowl("10","10","10");
        assertEquals(300, test.totalScore());
    }
    @Test
    public void allThree() throws Exception {
        System.out.println("All Three");
        Bowling test = new Bowling();
        for(int i = 0; i < 10; i++)
            test.bowl("3", "3");
        assertEquals(60, test.totalScore());
    }
    @Test
    public void oneSpare() throws Exception {
        System.out.println("One Spare");
        Bowling test = new Bowling();
        for(int i = 0; i < 10; i++) {
            if(i == 4)
                test.bowl("4", "6");
            else
                test.bowl("3", "3");
        }
        assertEquals(67, test.totalScore());
    }
    @Test
    public void twoSpare() throws Exception {
        System.out.println("Two Spares");
        Bowling test = new Bowling();
        for(int i = 0; i < 10; i++) {
            if(i == 4 || i == 5)
                test.bowl("4", "6");
            else
                test.bowl("3", "3");
        }
        assertEquals(75, test.totalScore());
    }
    @Test
    public void oneStrike() throws Exception {
        System.out.println("One Strike");
        Bowling test = new Bowling();
        for(int i = 0; i < 10; i++) {
            if(i == 4)
                test.bowl("10");
            else
                test.bowl("3", "3");
        }
        assertEquals(70, test.totalScore());
    }
    @Test
    public void twoStrikes() throws Exception {
        System.out.println("Two Strikes");
        Bowling test = new Bowling();
        for(int i = 0; i < 10; i++) {
            if(i == 4 || i == 5)
                test.bowl("10");
            else
                test.bowl("3", "3");
        }
        assertEquals(87, test.totalScore());
    }
    @Test
    public void oneStrikeTenth() throws Exception {
        System.out.println("One Strike in Tenth");
        Bowling test = new Bowling();
        for(int i = 0; i < 9; i++)
            test.bowl("3", "3");
        test.bowl("10","3","3");
        assertEquals(70, test.totalScore());
    }
    @Test
    public void threeStrikeTenth() throws Exception {
        System.out.println("Three Strikes in Tenth");
        Bowling test = new Bowling();
        for(int i = 0; i < 9; i++)
            test.bowl("3", "3");
        test.bowl("10","10","10");
        assertEquals(84, test.totalScore());
    }
    @Test
    public void allSpares() throws Exception {
        System.out.println("One Spare in Tenth");
        Bowling test = new Bowling();
        for(int i = 0; i < 9; i++) {
            test.bowl("3","7");
        }
        test.bowl("3","7","3");
        assertEquals(130, test.totalScore());
    }
}
```

Screenshots

simple game with two continued strikes in the middle and others are 3 pins down, total score is 87

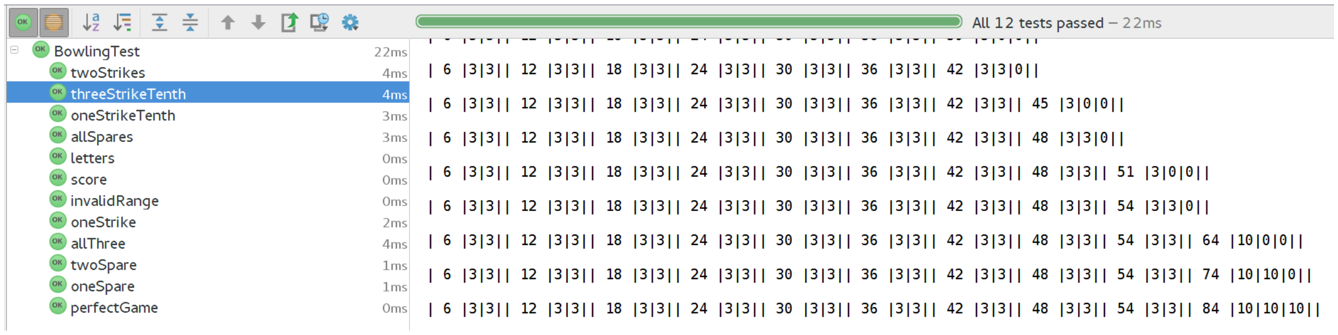


```
BowlingTest
All 12 tests passed – 22ms

BowlingTest          22ms
  twoStrikes          4ms
  threeStrikeTenth    4ms
  oneStrikeTenth      3ms
  allSpares           3ms
  letters             0ms
  score               0ms
  invalidRange        0ms
  oneStrike           2ms
  allThree            4ms
  twoSpare            1ms
  oneSpare            1ms
  perfectGame         0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 60 |10|0|| 63 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|| 72 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|| 75 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|| 75 |3|3|| 78 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|| 75 |3|3|| 81 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|| 75 |3|3|| 81 |3|3|| 84 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 47 |10|0|| 63 |10|0|| 69 |3|3|| 75 |3|3|| 81 |3|3|| 87 |3|3|0||
```

simple game with one strike in the tenth frame, each bonus is 10 and others are 3 pins down, total score is 84

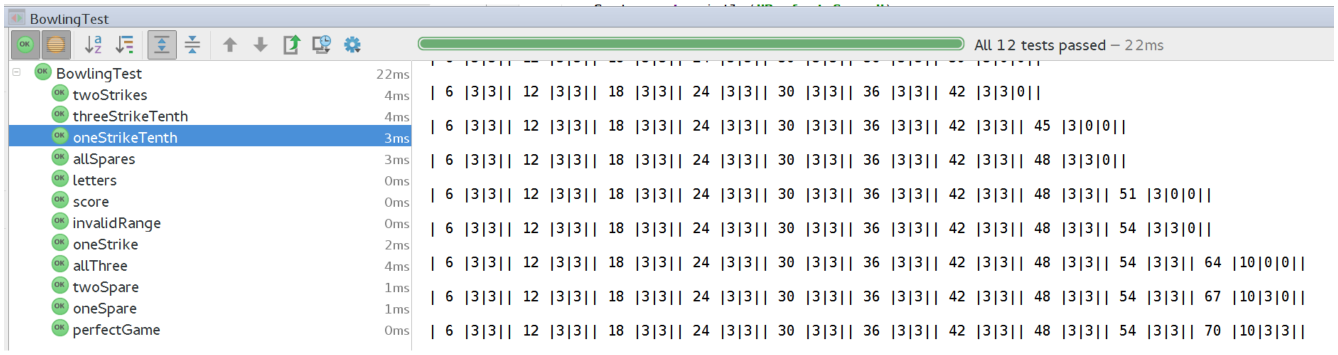


```
All 12 tests passed – 22ms

BowlingTest          22ms
  twoStrikes          4ms
  threeStrikeTenth    4ms
  oneStrikeTenth      3ms
  allSpares           3ms
  letters             0ms
  score               0ms
  invalidRange        0ms
  oneStrike           2ms
  allThree            4ms
  twoSpare            1ms
  oneSpare            1ms
  perfectGame         0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 45 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 51 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 64 |10|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 74 |10|10|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 84 |10|10|10||
```

simple game with one **strike in the tenth frame, bonus is 10, and others are 3 pins down, total score is 70 (tenth frame is 10, then 3, then 3)**

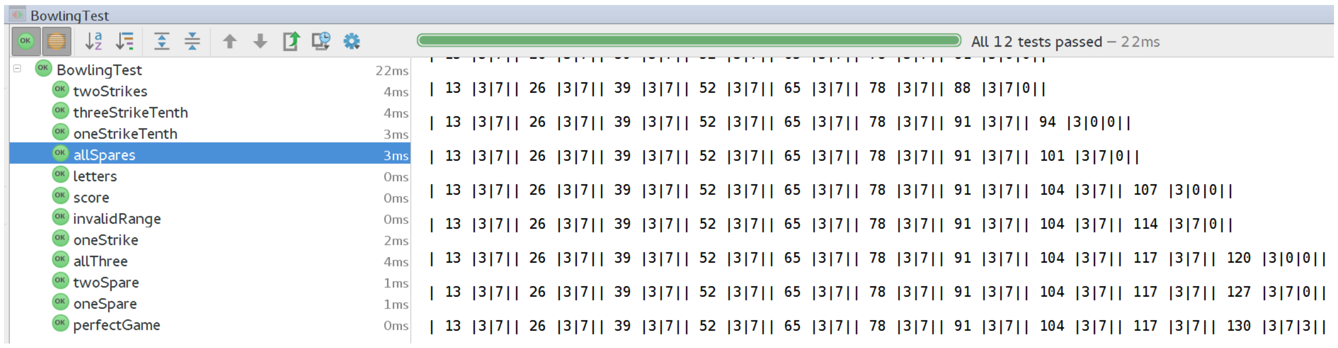


```
BowlingTest
All 12 tests passed – 22ms

BowlingTest          22ms
  twoStrikes          4ms
  threeStrikeTenth    4ms
  oneStrikeTenth      3ms
  allSpares           3ms
  letters             0ms
  score               0ms
  invalidRange        0ms
  oneStrike           2ms
  allThree            4ms
  twoSpare            1ms
  oneSpare            1ms
  perfectGame         0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 45 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 51 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 64 |10|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 67 |10|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 70 |10|3|3||
```

a game with all spare (3/7) total score is 130

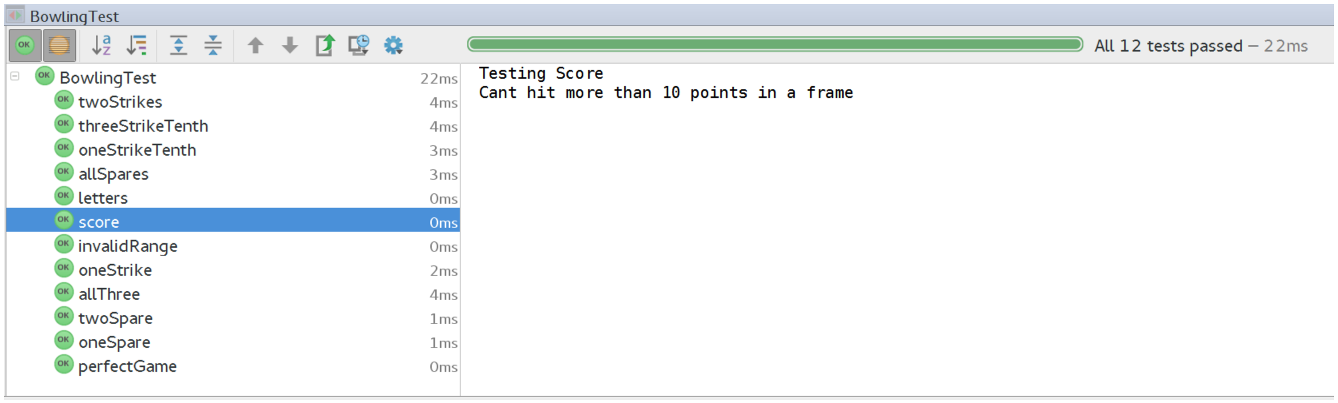


```
BowlingTest
All 12 tests passed – 22ms

BowlingTest          22ms
  twoStrikes          4ms
  threeStrikeTenth    4ms
  oneStrikeTenth      3ms
  allSpares           3ms
  letters             0ms
  score               0ms
  invalidRange        0ms
  oneStrike           2ms
  allThree            4ms
  twoSpare            1ms
  oneSpare            1ms
  perfectGame         0ms
```

```
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 88 |3|7|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 94 |3|0|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 101 |3|7|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 104 |3|7|| 107 |3|0|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 104 |3|7|| 114 |3|7|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 104 |3|7|| 117 |3|7|| 120 |3|0|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 104 |3|7|| 117 |3|7|| 127 |3|7|0||
| 13 |3|7|| 26 |3|7|| 39 |3|7|| 52 |3|7|| 65 |3|7|| 78 |3|7|| 91 |3|7|| 104 |3|7|| 117 |3|7|| 130 |3|7|3||
```

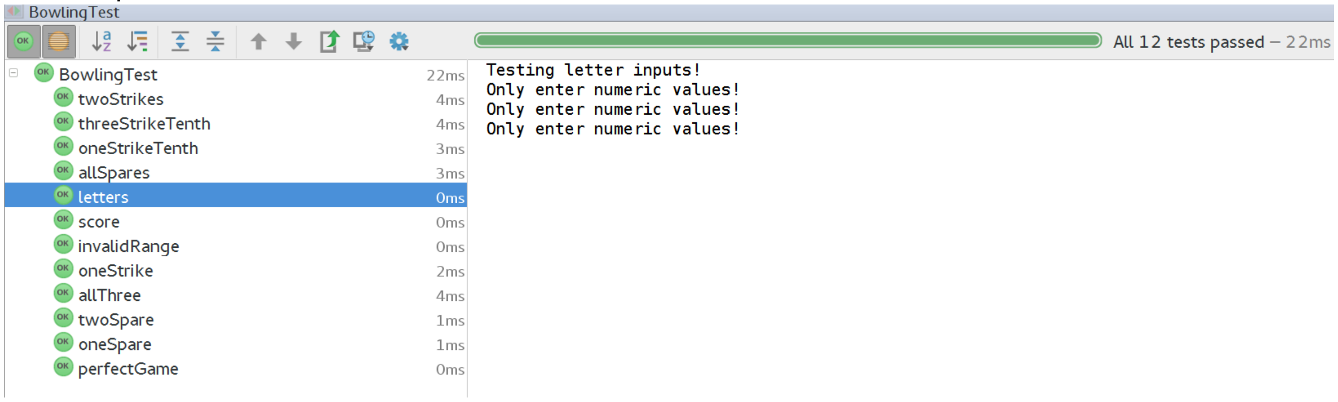
with two throws in the same frame bigger than 10



```
BowlingTest                          All 12 tests passed – 22ms
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
Testing Score
Cant hit more than 10 points in a frame
```

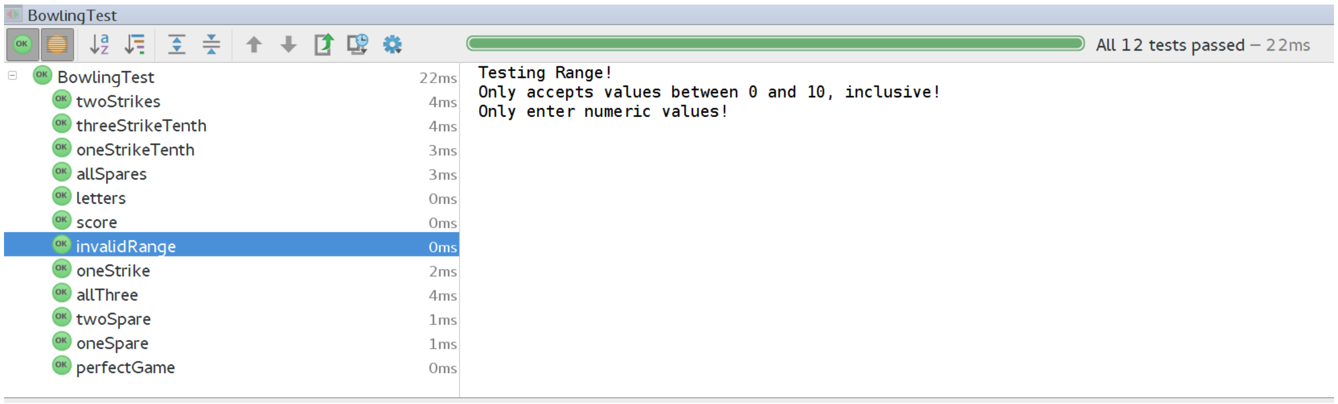
with letter input



```
BowlingTest                          All 12 tests passed – 22ms
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
Testing letter inputs!
Only enter numeric values!
Only enter numeric values!
Only enter numeric values!
```

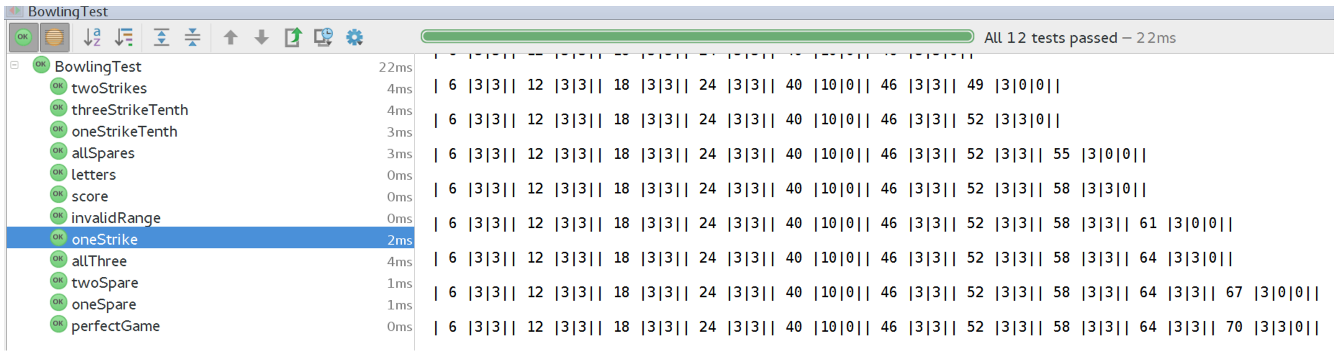
with 15 for one throw



```
BowlingTest                          All 12 tests passed – 22ms
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
Testing Range!
Only accepts values between 0 and 10, inclusive!
Only enter numeric values!
```

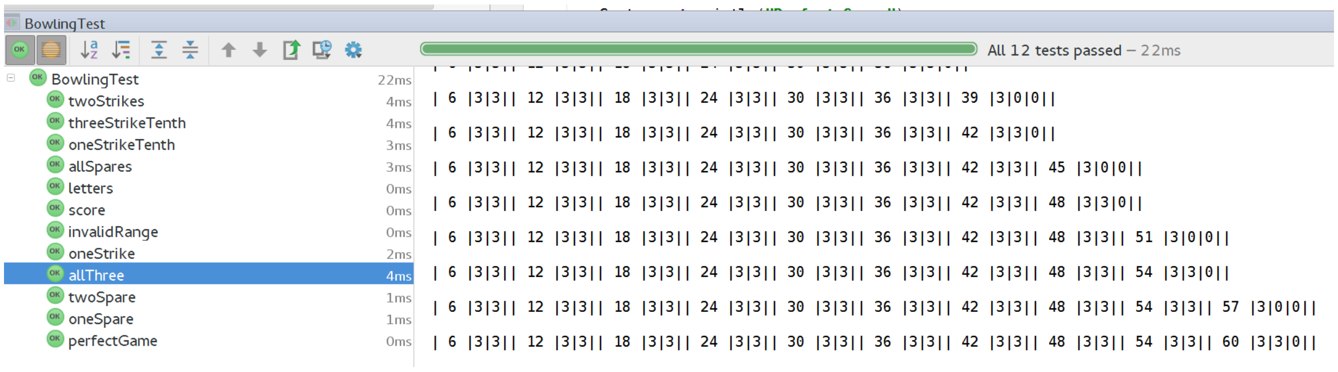
simple game with one strike in the middle and others are 3 pins down, total score is 70



```
BowlingTest                          All 12 tests passed – 22ms
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 49 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|| 55 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|| 58 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|| 58 |3|3|| 61 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|| 58 |3|3|| 64 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|| 58 |3|3|| 64 |3|3|| 67 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 40 |10|0|| 46 |3|3|| 52 |3|3|| 58 |3|3|| 64 |3|3|| 70 |3|3|0||
```

simple game with all throw with 3 pins down, total score is 60



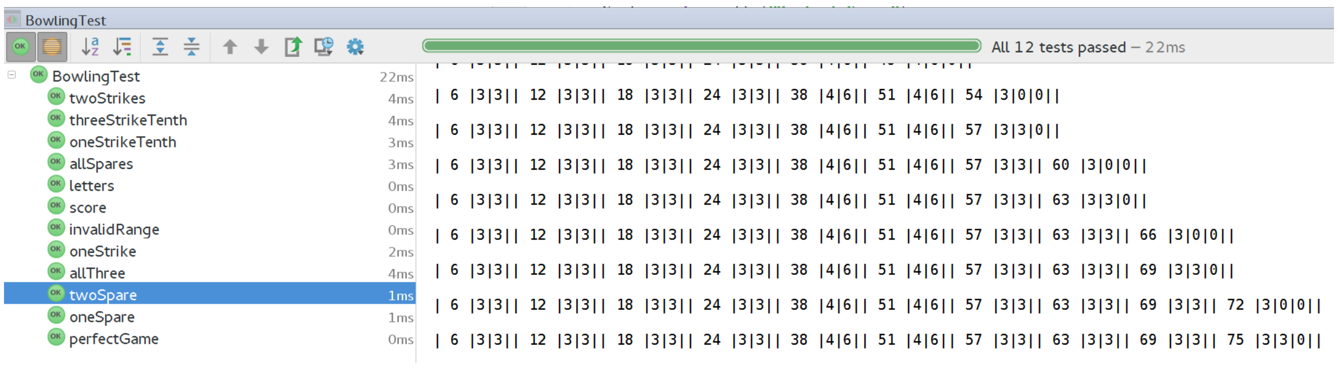
BowlingTest

All 12 tests passed – 22ms

```
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 39 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 45 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 51 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 57 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 30 |3|3|| 36 |3|3|| 42 |3|3|| 48 |3|3|| 54 |3|3|| 60 |3|3|0||
```

simple game with two continued spares (4/6) in the middle and others are 3 pins down, total score is 75



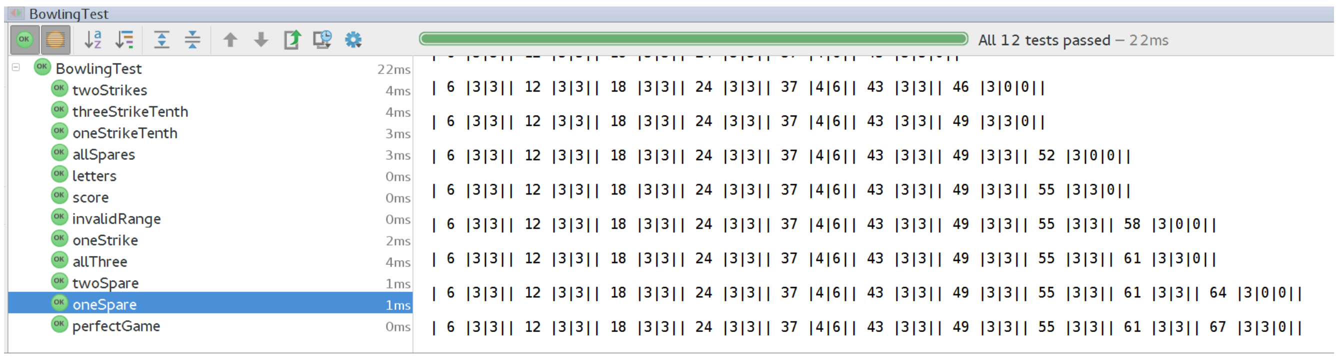
BowlingTest

All 12 tests passed – 22ms

```
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 54 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|| 60 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|| 63 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|| 63 |3|3|| 66 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|| 63 |3|3|| 69 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|| 63 |3|3|| 69 |3|3|| 72 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 38 |4|6|| 51 |4|6|| 57 |3|3|| 63 |3|3|| 69 |3|3|| 75 |3|3|0||
```

simple game with one spare (4/6) in the middle, and others are 3 pins down, total score is 67



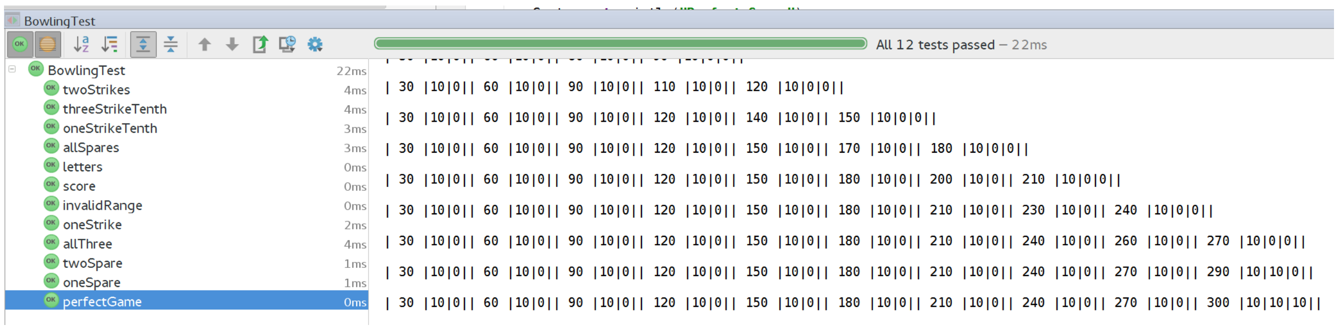
BowlingTest

All 12 tests passed – 22ms

```
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 46 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|| 52 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|| 55 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|| 55 |3|3|| 58 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|| 55 |3|3|| 61 |3|3|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|| 55 |3|3|| 61 |3|3|| 64 |3|0|0||
| 6 |3|3|| 12 |3|3|| 18 |3|3|| 24 |3|3|| 37 |4|6|| 43 |3|3|| 49 |3|3|| 55 |3|3|| 61 |3|3|| 67 |3|3|0||
```

perfect game with all strike, total score is 300



BowlingTest

All 12 tests passed – 22ms

```
BowlingTest        22ms
  twoStrikes        4ms
  threeStrikeTenth  4ms
  oneStrikeTenth    3ms
  allSpares         3ms
  letters           0ms
  score             0ms
  invalidRange      0ms
  oneStrike         2ms
  allThree          4ms
  twoSpare          1ms
  oneSpare          1ms
  perfectGame       0ms
```

```
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 110 |10|0|| 120 |10|0|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 140 |10|0|| 150 |10|0|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 150 |10|0|| 170 |10|0|| 180 |10|0|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 150 |10|0|| 180 |10|0|| 200 |10|0|| 210 |10|0|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 150 |10|0|| 180 |10|0|| 210 |10|0|| 230 |10|0|| 240 |10|0|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 150 |10|0|| 180 |10|0|| 210 |10|0|| 240 |10|0|| 260 |10|0|| 270 |10|0|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 150 |10|0|| 180 |10|0|| 210 |10|0|| 240 |10|0|| 270 |10|0|| 290 |10|10|0||
| 30 |10|0|| 60 |10|0|| 90 |10|0|| 120 |10|0|| 150 |10|0|| 180 |10|0|| 210 |10|0|| 240 |10|0|| 270 |10|0|| 300 |10|10|10||
```