**TASK DESCRIPTION**

This is a program to calculate statistics based on complete recorded scores in the season. Scores are recorded in external text file which needs to be imported and all the data of scores are stored in array. There are total 18 football teams and 22 round in each season with scores of all the team in all the matches are stored in files. Statistics like the lowest score, highest score, range, average, median and mode needs to be calculated. Whole program is completed by following steps:

1. The file in which all the score data is stored is read.
2. Total number of scores are counted by calculating teams and matches in season.
3. Then all the score data is stored in array which is then passed to many methods to calculate statistics.
4. Statistics of lowest number, highest number, range, average, median and mode is calculated with the scores data recorded in array.
5. Then all the calculated data is displayed.

**TASK OUTPUT**

|  |  |
| --- | --- |
| **Test Data** | **Screenshot** |
| Reads data from text file containing list of numbers (scores). |  |
| Gives error if program is not able to locate the file. (wrong file path) |  |

**TASK CODE**

// Program starts from here

**import** java.util.\*;

**import** java.io.\*;

**public** **class** ProcessScores {

**public** **static** **void** main(String[] args) **throws** Exception

{

// connects the file from which score data needs to be read

File scoreFile = **new** File("afl.txt");

// reading score data from the file

Scanner getScore = **new** Scanner(scoreFile);

**int** totalTeams = 18;

**int** totalRounds = 22;

// according to given data of 18 teams and 22 matches, total number of score data in file is calculated

**int** totalNumberScores = totalTeams \* totalRounds;

// declaring array to store the score data

**int**[] scores = **new** **int**[totalNumberScores];

**int** count=0;

// looping through all the score data, read from the file

**while**(getScore.hasNext())

{

// storing score data into the array

scores[count] = getScore.nextInt();

count++;

}

getScore.close();

// calling methods by passing array of scores as an argument to all the methods to calculate required statistics

System.***out***.println("The lowest score is " + *LowestScore*(scores));

System.***out***.println("The highest score is " + *HighestScore*(scores));

System.***out***.println("The range value is " + *Range*(scores));

System.***out***.println("The average score is " + *AverageScore*(scores));

System.***out***.println("The median value is " + *Median*(scores));

System.***out***.println("The mode is " + *Mode*(scores));

}

// this is the method to calculate the lowest score

**public** **static** **int** LowestScore(**int**[] scores)

{

**int** lowestScore = scores[0];

// looping through all the recorded score data in array to find the lowest score

**for**(**int** item=0; item < scores.length; item++)

{

**if**(scores[item] < lowestScore)

{

lowestScore = scores[item];

}

}

**return** lowestScore;

}

// looping through all the recorded score data in array to find the highest score

**public** **static** **int** HighestScore(**int**[] scores)

{

**int** highestScore = scores[0];

// looping through all the recorded score data in array to find the highest score

**for**(**int** item=0; item < scores.length; item++)

{

**if**(scores[item] > highestScore)

{

highestScore = scores[item];

}

}

**return** highestScore;

}

// looping through all the recorded score data in array to find the range score

// range is calculated by finding difference between highest score and lowest score

**public** **static** **int** Range(**int**[] scores)

{

**int** lowestScore = scores[0];

**int** highestScore = scores[0];

**int** range;

// looping through all the recorded score data in array to find the lowest score

**for**(**int** item=0; item < scores.length; item++)

{

**if**(scores[item] < lowestScore)

{

lowestScore = scores[item];

}

}

// looping through all the recorded score data in array to find the highest score

**for**(**int** item=0; item < scores.length; item++)

{

**if**(scores[item] > highestScore)

{

highestScore = scores[item];

}

}

// calculating the range by getting difference of the highest and lowest number

range = highestScore - lowestScore;

**return** range;

}

// this is a method to calculate average score

// average score is calculated by adding all the numbers and then divide by the total number of data

**public** **static** **int** AverageScore(**int**[] scores)

{

**int** totalScore = 0;

**int** average;

**for**(**int** item=0; item < scores.length; item++)

{

totalScore = totalScore + scores[item];

}

average = totalScore/scores.length;

**return** average;

}

// this is a method to calculate median score

// median number is calculated by finding the value of middle number index

**public** **static** **int** Median(**int**[] scores)

{

**int** median;

// check if the middle number is even

**if**(scores.length % 2 == 0)

{

// formula to calculate middle number if total number of data is even

median = (scores[(scores.length/2) - 1] + scores[scores.length/2])/2;

}

**else**

{

// formula to calculate middle number if total number of data is odd

median = scores[scores.length/2];

}

**return** median;

}

// this is a method to calculate mode

// mode is the number which is repeated maximum number of times in the list of numbers

**public** **static** **int** Mode(**int**[] scores)

{

**int** mode;

**int** count = 0;

**int** tempItem;

**int** repeatedTimes = 0;

**int** repeatedScore = 0;

// looping through the list of numbers and increase the count for each number if it is repeated again

**for**(**int** i=0; i<scores.length; i++)

{

tempItem = scores[i];

count = 1;

**for**(**int** j=i+1; j<scores.length; j++)

{

**int** repeatedNumber = scores[j];

**if**(repeatedNumber == tempItem)

{

count++;

}

}

**if**(count > repeatedTimes)

{

repeatedTimes = count;

repeatedScore = scores[i];

}

}

mode = repeatedScore;

**return** mode;

}

}