1. Write a Java program that accepts a list of student projects andreturns the number of projects that were completed on time, late,or early, and the average time it took for each student to completetheir projects

Import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

class Project {

private String studentName;

private int completionTime; // in days

public Project(String studentName, int completionTime) {

this.studentName = studentName;

this.completionTime = completionTime;

}public String getStudentName() {

return studentName;

}public int getCompletionTime() {

return completionTime;

}

}public class StudentProjects {

public static void main(String[] args) {

List<Project> projects = new ArrayList<>();

projects.add(new Project("John", 5));

projects.add(new Project("Alice", 7));

projects.add(new Project("Bob", 3));

projects.add(new Project("Carol", 6));

projects.add(new Project("David", 8));

Map<String, Integer> onTime = new HashMap<>();

Map<String, Integer> late = new HashMap<>();

Map<String, Integer> early = new HashMap<>();

Map<String, Integer> projectCounts = new HashMap<>();

Map<String, Integer> totalTime = new HashMap<>();

for (Project project : projects) {

String studentName = project.getStudentName();

int completionTime = project.getCompletionTime();

projectCounts.put(studentName, projectCounts.getOrDefault(studentName, 0) + 1);

totalTime.put(studentName, totalTime.getOrDefault(studentName, 0) + completionTime);

if (completionTime == 5) { // Assuming 5 days is the on-time threshold

onTime.put(studentName, onTime.getOrDefault(studentName, 0) + 1);

} else if (completionTime > 5) {

late.put(studentName, late.getOrDefault(studentName, 0) + 1);

} else {

early.put(studentName, early.getOrDefault(studentName, 0) + 1);

}

}

System.out.println("Number of projects completed on time:");

printMap(onTime);

System.out.println("\nNumber of projects completed late:");

printMap(late);

System.out.println("\nNumber of projects completed early:");

printMap(early);

System.out.println("\nAverage completion time for each student:");

for (String studentName : projectCounts.keySet()) {

int averageTime = totalTime.get(studentName) / projectCounts.get(studentName);

System.out.println(studentName + ": " + averageTime + " days");

}

}

private static void printMap(Map<String, Integer> map) {

for (Map.Entry<String, Integer> entry : map.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

}

}

1. Write a Java program that accepts a list of movie ratings andreturns the number of movies rated in various categories (e.g.,PG, PG-13, R, etc.), and the average rating for each category.

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

class Movie {

private String category;

private double rating;

public Movie(String category, double rating) {

this.category = category;

this.rating = rating;

}

public String getCategory() {

return category;

}

public double getRating() {

return rating;

}

}

public class MovieRatings {

public static void main(String[] args) {

List<Movie> movies = new ArrayList<>();

movies.add(new Movie("PG", 4.5));

movies.add(new Movie("PG-13", 3.8));

movies.add(new Movie("R", 4.2));

movies.add(new Movie("PG", 4.1));

movies.add(new Movie("PG-13", 3.9));

movies.add(new Movie("R", 4.5));

movies.add(new Movie("PG", 3.7));

Map<String, Integer> movieCounts = new HashMap<>();

Map<String, Double> totalRatings = new HashMap<>();

for (Movie movie : movies) {

String category = movie.getCategory();

double rating = movie.getRating();

movieCounts.put(category, movieCounts.getOrDefault(category, 0) + 1);

totalRatings.put(category, totalRatings.getOrDefault(category, 0.0) + rating);

}

System.out.println("Number of movies rated in each category:");

for (String category : movieCounts.keySet()) {

System.out.println(category + ": " + movieCounts.get(category));

}

System.out.println("\nAverage rating for each category:");

for (String category : totalRatings.keySet()) {

double averageRating = totalRatings.get(category) / movieCounts.get(category);

System.out.println(category + ": " + averageRating);

}

}

}

3.Write a Java program that accepts a list of book reviews andreturns the number of books reviewed within specified ratingranges (e.g., 1-5 stars, 6-10 stars), and the count of books withpositive, neutral, and negative reviews.

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

class BookReview {

private int rating;

public BookReview(int rating) {

this.rating = rating;

}

public int getRating() {

return rating;

}

}

public class BookReviews {

public static void main(String[] args) {

List<BookReview> reviews = new ArrayList<>();

reviews.add(new BookReview(4));

reviews.add(new BookReview(8));

reviews.add(new BookReview(6));

reviews.add(new BookReview(3));

reviews.add(new BookReview(9));

reviews.add(new BookReview(2));

reviews.add(new BookReview(7));

int[] ratingRanges = {1, 5, 6, 10}; // Define rating ranges

Map<String, Integer> countByRatingRange = new HashMap<>();

countByRatingRange.put("1-5 stars", 0);

countByRatingRange.put("6-10 stars", 0);

int positiveCount = 0;

int neutralCount = 0;

int negativeCount = 0;

for (BookReview review : reviews) {

int rating = review.getRating();

// Categorize based on rating ranges

if (rating >= ratingRanges[0] && rating <= ratingRanges[1]) {

countByRatingRange.put("1-5 stars", countByRatingRange.get("1-5 stars") + 1);

} else if (rating >= ratingRanges[2] && rating <= ratingRanges[3]) {

countByRatingRange.put("6-10 stars", countByRatingRange.get("6-10 stars") + 1);

}

// Categorize as positive, neutral, or negative based on rating

if (rating >= 7) {

positiveCount++;

} else if (rating >= 4) {

neutralCount++;

} else {

negativeCount++;

}

}

System.out.println("Number of books reviewed within specified rating ranges:");

for (String range : countByRatingRange.keySet()) {

System.out.println(range + ": " + countByRatingRange.get(range));

}

System.out.println("\nCount of books with positive, neutral, and negative reviews:");

System.out.println("Positive: " + positiveCount);

System.out.println("Neutral: " + neutralCount);

System.out.println("Negative: " + negativeCount);

}

}

1. Write a Java program that accepts a list of student test scoresand returns the number of students who scored above theaverage, at the average, or below the average, and the medianscore for each group.

import java.util.ArrayList;

import java.util.Collections;

1. import java.util.List;

class Student {

1. private String name;
2. private double score;
3. public Student(String name, double score) {
4. this.name = name;
5. this.score = score;
6. }
7. public double getScore() {
8. return score;
9. }
10. }
11. public class StudentScores {
12. public static void main(String[] args) {
13. List<Student> students = new ArrayList<>();
14. students.add(new Student("John", 85));
15. students.add(new Student("Alice", 92));
16. students.add(new Student("Bob", 78));
17. students.add(new Student("Carol", 63));
18. students.add(new Student("David", 75));
19. students.add(new Student("Emma", 80));
20. students.add(new Student("Frank", 88));
21. double totalScore = 0;
22. List<Double> scores = new ArrayList<>();
23. for (Student student : students) {
24. totalScore += student.getScore();
25. scores.add(student.getScore());
26. }
27. double averageScore = totalScore / students.size();
28. Collections.sort(scores);
29. int belowAverageCount = 0;
30. int atAverageCount = 0;
31. int aboveAverageCount = 0;
32. for (double score : scores) {
33. if (score < averageScore) {
34. belowAverageCount++;
35. } else if (score == averageScore) {
36. atAverageCount++;
37. } else {
38. aboveAverageCount++;
39. }
40. }
41. double medianBelowAverage = median(scores.subList(0, belowAverageCount));
42. double medianAtAverage = median(scores.subList(belowAverageCount, belowAverageCount + atAverageCount));
43. double medianAboveAverage = median(scores.subList(belowAverageCount + atAverageCount, students.size()));
44. System.out.println("Number of students who scored:");
45. System.out.println("Above the average: " + aboveAverageCount);
46. System.out.println("At the average: " + atAverageCount);
47. System.out.println("Below the average: " + belowAverageCount);
48. System.out.println("\nMedian score for each group:");
49. System.out.println("Above the average: " + medianAboveAverage);
50. System.out.println("At the average: " + medianAtAverage);
51. System.out.println("Below the average: " + medianBelowAverage);
52. }
53. private static double median(List<Double> numbers) {
54. int middle = numbers.size() / 2;
55. if (numbers.size() % 2 == 1) {
56. return numbers.get(middle);
57. } else {
58. return (numbers.get(middle - 1) + numbers.get(middle)) / 2.0;
59. }
60. }
61. }

5.Write a Java program that accepts a list of restaurant ratingsand returns the number of restaurants rated within specific ranges(e.g., 1-5, 6-10), and the average rating for each range.

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

class RestaurantRating {

private double rating;

public RestaurantRating(double rating) {

this.rating = rating;

}

public double getRating() {

return rating;

}

}

public class RestaurantRatings {

public static void main(String[] args) {

List<RestaurantRating> ratings = new ArrayList<>();

ratings.add(new RestaurantRating(4.5));

ratings.add(new RestaurantRating(7.8));

ratings.add(new RestaurantRating(6.2));

ratings.add(new RestaurantRating(3.5));

ratings.add(new RestaurantRating(9.1));

ratings.add(new RestaurantRating(2.9));

ratings.add(new RestaurantRating(8.4));

int[] ratingRanges = {1, 5, 6, 10}; // Define rating ranges

Map<String, Integer> countByRange = new HashMap<>();

Map<String, Double> totalRatingByRange = new HashMap<>();

// Initialize count and total rating for each range

for (int i = 0; i < ratingRanges.length - 1; i++) {

String range = ratingRanges[i] + "-" + ratingRanges[i + 1];

countByRange.put(range, 0);

totalRatingByRange.put(range, 0.0);

}

// Count restaurants within each range and accumulate ratings

for (RestaurantRating rating : ratings) {

double score = rating.getRating();

for (int i = 0; i < ratingRanges.length - 1; i++) {

if (score >= ratingRanges[i] && score <= ratingRanges[i + 1]) {

String range = ratingRanges[i] + "-" + ratingRanges[i + 1];

countByRange.put(range, countByRange.get(range) + 1);

totalRatingByRange.put(range, totalRatingByRange.get(range) + score);

break;

}

}

}

// Calculate average rating for each range and print results

System.out.println("Number of restaurants rated within specific ranges and their average ratings:");

for (String range : countByRange.keySet()) {

int count = countByRange.get(range);

double totalRating = totalRatingByRange.get(range);

double averageRating = count == 0 ? 0 : totalRating / count;

System.out.println(range + ": " + count + " restaurants, Average rating: " + averageRating);

}

}

}