1.Write a Java program that accepts a list of product sales and

returns the number of products sold within specific price ranges

(e.g., $0-50, $50-100, $100-200, etc.), and the total revenue

generated within each price range.

Source code

import java.util.ArrayList;

import java.util.List;

class ProductSale {

double price;

int quantity;

public ProductSale(double price, int quantity) {

this.price = price;

this.quantity = quantity;

}

public double getPrice() {

return price;

}

public int getQuantity() {

return quantity;

}

}

public class SalesReport {

public static void main(String[] args) {

// Sample list of product sales

List<ProductSale> sales = new ArrayList<>();

sales.add(new ProductSale(45.0, 10));

sales.add(new ProductSale(75.0, 5));

sales.add(new ProductSale(120.0, 4));

sales.add(new ProductSale(60.0, 8));

sales.add(new ProductSale(30.0, 15));

sales.add(new ProductSale(200.0, 3));

// Price ranges

double[][] ranges = {{0, 50}, {50, 100}, {100, 200}};

// Calculate and print sales report

calculateAndPrintSalesReport(sales, ranges);

}

public static void calculateAndPrintSalesReport(List<ProductSale> sales, double[][] ranges) {

for (double[] range : ranges) {

int count = 0;

double revenue = 0;

for (ProductSale sale : sales) {

if (sale.getPrice() >= range[0] && sale.getPrice() < range[1]) {

count += sale.getQuantity();

revenue += sale.getPrice() \* sale.getQuantity();

}

}

System.out.println("Price Range $" + range[0] + " - $" + range[1] + ":");

System.out.println("Number of Products Sold: " + count);

System.out.println("Total Revenue Generated: $" + revenue);

System.out.println();

}

}

}

2.Write a Java program that accepts a list of weather

data (e.g., temperature, humidity) and returns the

number of days with temperatures within certain ranges

(e.g., <0°C, 0-10°C, 10-20°C, etc.), and the average

humidity for each temperature range.

Source code:

import java.util.ArrayList;

import java.util.List;

class WeatherData {

double temperature;

double humidity;

public WeatherData(double temperature, double humidity) {

this.temperature = temperature;

this.humidity = humidity;

}

public double getTemperature() {

return temperature;

}

public double getHumidity() {

return humidity;

}

}

public class WeatherReport {

public static void main(String[] args) {

// Sample list of weather data

List<WeatherData> weatherDataList = new ArrayList<>();

weatherDataList.add(new WeatherData(-5, 80));

weatherDataList.add(new WeatherData(5, 60));

weatherDataList.add(new WeatherData(15, 55));

weatherDataList.add(new WeatherData(25, 45));

weatherDataList.add(new WeatherData(3, 70));

weatherDataList.add(new WeatherData(18, 50));

// Temperature ranges

double[][] ranges = {{Double.NEGATIVE\_INFINITY, 0}, {0, 10}, {10, 20}, {20, Double.POSITIVE\_INFINITY}};

// Calculate and print weather report

calculateAndPrintWeatherReport(weatherDataList, ranges);

}

public static void calculateAndPrintWeatherReport(List<WeatherData> weatherData, double[][] ranges) {

for (double[] range : ranges) {

int count = 0;

double totalHumidity = 0;

for (WeatherData data : weatherData) {

if (data.getTemperature() >= range[0] && data.getTemperature() < range[1]) {

count++;

totalHumidity += data.getHumidity();

}

}

if (count > 0) { // Avoid division by zero

double averageHumidity = totalHumidity / count;

System.out.println("Temperature Range " + range[0] + "°C to " + range[1] + "°C:");

System.out.println("Number of Days: " + count);

System.out.println("Average Humidity: " + averageHumidity + "%");

System.out.println();

}

}

}

}

3.Write a Java program that accepts a list of medical test

results and returns the number of patients with results

falling within specific ranges (e.g., normal, borderline,

high), and the average value for each range.

Source code:

import java.util.ArrayList;

import java.util.List;

class TestResult {

String patientName;

double resultValue;

public TestResult(String patientName, double resultValue) {

this.patientName = patientName;

this.resultValue = resultValue;

}

public String getPatientName() {

return patientName;

}

public double getResultValue() {

return resultValue;

}

}

public class MedicalReport {

static class Range {

String label;

double min;

double max;

public Range(String label, double min, double max) {

this.label = label;

this.min = min;

this.max = max;

}

}

public static void main(String[] args) {

// Sample list of medical test results

List<TestResult> results = new ArrayList<>();

results.add(new TestResult("John Doe", 4.5));

results.add(new TestResult("Jane Smith", 6.2));

results.add(new TestResult("Bob Johnson", 5.9));

results.add(new TestResult("Sarah Brown", 7.1));

results.add(new TestResult("Mike Davis", 3.9));

// Define ranges

List<Range> ranges = new ArrayList<>();

ranges.add(new Range("Normal", 0, 5));

ranges.add(new Range("Borderline", 5, 6.5));

ranges.add(new Range("High", 6.5, Double.POSITIVE\_INFINITY));

// Calculate and print medical report

calculateAndPrintMedicalReport(results, ranges);

}

public static void calculateAndPrintMedicalReport(List<TestResult> results, List<Range> ranges) {

for (Range range : ranges) {

int count = 0;

double totalValue = 0;

for (TestResult result : results) {

if (result.getResultValue() >= range.min && result.getResultValue() < range.max) {

count++;

totalValue += result.getResultValue();

}

}

if (count > 0) { // Avoid division by zero

double averageValue = totalValue / count;

System.out.println(range.label + " Results:");

System.out.println("Number of Patients: " + count);

System.out.println("Average Value: " + averageValue);

System.out.println();

}

}

}

}

4.Write a Java program that accepts a list of employee

work hours and returns the number of employees who

worked more than 40 hours, exactly 40 hours, or less

than 40 hours in a week, and the average hours worked

per day for each group.

Source code:

import java.util.ArrayList;

import java.util.List;

class EmployeeHours {

String employeeName;

double[] hours; // Assuming hours for 5 working days in a week

public EmployeeHours(String employeeName, double[] hours) {

this.employeeName = employeeName;

this.hours = hours;

}

public String getEmployeeName() {

return employeeName;

}

public double[] getHours() {

return hours;

}

public double getTotalHours() {

double total = 0;

for (double hour : hours) {

total += hour;

}

return total;

}

}

public class WorkHoursReport {

static class CategorySummary {

int employeeCount = 0;

double totalHours = 0;

void addEmployee(double totalHours) {

employeeCount++;

this.totalHours += totalHours;

}

double averageHoursPerDay() {

return totalHours / (employeeCount \* 5); // Assuming a 5-day workweek

}

}

public static void main(String[] args) {

// Sample list of employee work hours

List<EmployeeHours> employees = new ArrayList<>();

employees.add(new EmployeeHours("John Doe", new double[]{8, 8, 8, 8, 8}));

employees.add(new EmployeeHours("Jane Smith", new double[]{9, 9, 9, 9, 5}));

employees.add(new EmployeeHours("Bob Johnson", new double[]{7, 8, 7, 8, 10}));

employees.add(new EmployeeHours("Sarah Brown", new double[]{8, 8, 8, 6, 6}));

// Process and print work hours report

printWorkHoursReport(employees);

}

public static void printWorkHoursReport(List<EmployeeHours> employees) {

CategorySummary moreThan40 = new CategorySummary();

CategorySummary exactly40 = new CategorySummary();

CategorySummary lessThan40 = new CategorySummary();

for (EmployeeHours employee : employees) {

double totalHours = employee.getTotalHours();

if (totalHours > 40) {

moreThan40.addEmployee(totalHours);

} else if (totalHours == 40) {

exactly40.addEmployee(totalHours);

} else {

lessThan40.addEmployee(totalHours);

}

}

System.out.println("More than 40 hours:");

System.out.println("Number of Employees: " + moreThan40.employeeCount);

System.out.println("Average hours worked per day: " + moreThan40.averageHoursPerDay());

System.out.println();

System.out.println("Exactly 40 hours:");

System.out.println("Number of Employees: " + exactly40.employeeCount);

System.out.println("Average hours worked per day: " + exactly40.averageHoursPerDay());

System.out.println();

System.out.println("Less than 40 hours:");

System.out.println("Number of Employees: " + lessThan40.employeeCount);

System.out.println("Average hours worked per day: " + lessThan40.averageHoursPerDay());

}

}

5.Write a Java program that accepts a list of housing

prices and returns the number of houses sold within

specific price ranges (e.g., $100,000-200,000,

$200,000-300,000, etc.), and the average square footage

for each price range.

Source code:

import java.util.ArrayList;

import java.util.List;

class House {

double price;

double squareFootage;

public House(double price, double squareFootage) {

this.price = price;

this.squareFootage = squareFootage;

}

public double getPrice() {

return price;

}

public double getSquareFootage() {

return squareFootage;

}

}

public class HousingReport {

static class PriceRange {

double minPrice;

double maxPrice;

List<House> houses = new ArrayList<>();

public PriceRange(double minPrice, double maxPrice) {

this.minPrice = minPrice;

this.maxPrice = maxPrice;

}

public void addHouse(House house) {

houses.add(house);

}

public double averageSquareFootage() {

double totalSquareFootage = 0;

for (House house : houses) {

totalSquareFootage += house.getSquareFootage();

}

return houses.isEmpty() ? 0 : totalSquareFootage / houses.size();

}

}

public static void main(String[] args) {

// Sample list of houses

List<House> houses = new ArrayList<>();

houses.add(new House(150000, 1200));

houses.add(new House(250000, 1500));

houses.add(new House(350000, 1800));

houses.add(new House(450000, 2000));

houses.add(new House(175000, 1100));

houses.add(new House(275000, 1600));

// Define price ranges

List<PriceRange> ranges = new ArrayList<>();

ranges.add(new PriceRange(100000, 200000));

ranges.add(new PriceRange(200000, 300000));

ranges.add(new PriceRange(300000, 400000));

ranges.add(new PriceRange(400000, 500000));

// Process and print housing report

calculateAndPrintHousingReport(houses, ranges);

}

public static void calculateAndPrintHousingReport(List<House> houses, List<PriceRange> ranges) {

for (House house : houses) {

for (PriceRange range : ranges) {

if (house.getPrice() >= range.minPrice && house.getPrice() <= range.maxPrice) {

range.addHouse(house);

break; // Assuming a house fits into only one range

}

}

}

for (PriceRange range : ranges) {

System.out.println("Price Range $" + range.minPrice + " - $" + range.maxPrice + ":");

System.out.println("Number of Houses: " + range.houses.size());

System.out.println("Average Square Footage: " + range.averageSquareFootage());

System.out.println();

}

}

}