**Programming Assignment Unit 4**

**University of the People**

**CS 1102-01 - AY2024-T5: Programming 1**

**Avernell Christopher Melikhov**

**July 18th, 2024**

**Summary of Code**

The code is for a class named StockPriceAnalysis that analyzes stock prices. It contains methods to calculate averages, find maximum prices, count occurrences, and compute cumulative sums for stock price data. There are two versions of some methods: one for an arrays and one for an ArrayList.

### **Methods:**

1. **calculateAveragePrice(float[] array)**:
   * **Purpose**: Calculates the average price from an array of stock prices.
   * **How it Works**: Adds up all the prices in the array, then divides by the number of prices to find the average.
2. **calculateAveragePrice(ArrayList<Float> stockPrices)**:
   * **Purpose**: Calculates the average price from an ArrayList of stock prices.
   * **How it Works**: Adds up all the prices in the ArrayList, then divides by the number of prices to find the average.
3. **findMaximumPrice(float[] array)**:
   * **Purpose**: Finds the highest stock price in an array.
   * **How it Works**: Compares each price in the array to find the largest value.
4. **findMaximumPrice(ArrayList<Float> stockPrices)**:
   * **Purpose**: Finds the highest stock price in an ArrayList.
   * **How it Works**: Compares each price in the ArrayList to find the largest value.
5. **countOccurrences(float[] array, float targetPrice)**:
   * **Purpose**: Counts how many times a specific price appears in an array.
   * **How it Works**: Checks each price in the array and counts the number of times it matches the target price.
6. **computeCumulativeSum(ArrayList<Float> stockPrices)**:
   * **Purpose**: Calculates a running total of stock prices from an ArrayList.
   * **How it Works**: Adds up prices progressively and stores each cumulative sum in a new ArrayList.

The main method shows a sample array and arrayList with float values.

It then outputs:

* Average stock prices for both array and ArrayList.
* Maximum price for both array and ArrayList.
* Count of occurrences for a specific price in the array.
* Cumulative sum of prices from the ArrayList.

**Code:**

/\* This program provides methods to calculate averages, find maximum values,count occurrences,

\* and compute cumulative sums for stock prices using both arrays and ArrayLists. \*/

**package** application;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**public** **class** StockPriceAnalysis {

//\*\* Calculate Average Method for ARRAY\*\*

**public** **static** **float** calculateAveragePrice(**float**[] array) {

**float** arraySum=0;

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

arraySum += array[i];

}

**float** averagePrice = (**float**) arraySum/array.length; //Average is calculated by the total sum of the array

**return** averagePrice; //and then divided by the length.Average returned.

}//average array end

//\*\* Calculate Average Method for ARRAYLIST\*\*

**public** **static** **float** calculateAveragePrice(ArrayList<Float> stockPrices) {

**float** sum = 0;

**for** (**int** i = 0 ; i <stockPrices.size();i++ ) {

sum += stockPrices.get(i); // method for getting all the elements in the list

}

**return** sum / stockPrices.size(); // average for arrayList diving the sum by the size() method.

}//average (arrayList) end

//\*\*Maximum Price for ARRAY \*\*

**public** **static** **float** findMaximumPrice(**float**[]array) {

**float** maxPrice=array[0]; // starting with the first element

**for**(**int** i=1; i < array.length; i++) {

**if**(array[i]> maxPrice) { //After looping through the array, the highest price is found

maxPrice=array[i]; // assigns the highest number to maxPrice variable

}

}

**return** maxPrice;

}//max Price (array) end

// \*\*Maximum Price for ArrayList\*\*

**public** **static** **float** findMaximumPrice(ArrayList<Float> stockPrices) {

**float** maxPrice = stockPrices.get(0); // Initialize maxPrice with the first element

**for** (**int** i = 1; i < stockPrices.size(); i++) {

**if** (stockPrices.get(i) > maxPrice) { //After looping through the array, the highest price is found

maxPrice = stockPrices.get(i); //assigning to the max variable

}

}

**return** maxPrice;

}//max Price (arrayList) end

// \*\* Count Occurrences for ARRAY \*\*

**public** **static** **float** countOccurences(**float**[] array,**float** targetPrice) {

**float** count=0; //initializing counter variable

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

**if** (array[i] == targetPrice) { //incrementing counter whenever an element matches the target price

count++;

}

}

**return** count;

}//count end

//\*\* Computer Cumulative Sum for ArrayList\*\*

**public** **static** ArrayList <Float> computeCumulativeSum(ArrayList<Float>stockPrices){

ArrayList<Float> cumulativeSumPrices = **new** ArrayList<>(); //initializing new Array

**float** sum=0;

**for**(**int** i=0; i< stockPrices.size(); i++ ) {

sum += stockPrices.get(i); // Calculating the cumulative sum

cumulativeSumPrices.add(sum); //Adding the sum to the arrayList

}

**return** cumulativeSumPrices;

}//ArrayList end

//\*\* MAIN METHOD \*\*//

**public** **static** **void** main(String[]args) {

//\*\*Sample Data

// Daily Stock Price Array

**float**[] arrayStockPrices = {432.2f, 876.5f, 109.7f, 543.6f, 234.0f, 543.6f, 321.0f, 654.5f, 987.7f, 543.6f};

// Stock Price Array List

ArrayList<Float> stockPricesList= **new** ArrayList<>(Arrays.*asList*(258.4f,345.4f,563.5f,527.4f,356.7f,546.24f,431.5f,323.11f,654.32f,654.32f));

//Displaying results

System.***out***.println("Average Stock Prices for Daily Stock Prices: " + *calculateAveragePrice*(arrayStockPrices));

System.***out***.println();

System.***out***.println("Average Stock Prices for Price List: " + *calculateAveragePrice*(stockPricesList));

System.***out***.println();

System.***out***.println("Maximum Price for Daily Stock Prices : " + *findMaximumPrice*(arrayStockPrices));

System.***out***.println();

System.***out***.println("Maximum Price for Price List: " + *findMaximumPrice*(stockPricesList));

System.***out***.println();

System.***out***.println("The count occurrences of the number 543.6 is: " + *countOccurences*(arrayStockPrices,543.6f));

System.***out***.println();

System.***out***.println("The Cumulative Sum is: " + *computeCumulativeSum*(stockPricesList));

}

}

**References**

Coding with John. (2021, February 24). *Array vs. ArrayList in Java Tutorial - What’s The Difference?* [Video]. YouTube.

<https://www.youtube.com/watch?v=NbYgm0r7u6o>

Eck, D. J. (2022). *Introduction to programming using java version 9, JavaFX edition.* Licensed under CC 4.0.

<https://math.hws.edu/javanotes/>