**Programming Assignment Unit 4**

University of the People

CS 1102-01: Programming 1 AY2025-T2

Naeem Ahmed

13 December 2024

# Programming Assignment Unit 4

## Problem Statement

You are working on a data analysis project where you need to process an array of opening stock prices containing 10 days of data as float datatype.

**Question:**

Consider a scenario where you have been given an array of integers representing the daily stock prices of a company for a given period. You are also provided with an ArrayList of stock prices for the same period You are required to implement a program that performs the following tasks:

**Calculate the average stock price:**

Write a method, calculateAveragePrice, that takes the array of stock prices as input and returns the average price of the stocks.

**Find the maximum stock price:**

Write a method, findMaximumPrice, that takes the array of stock prices as input and returns the maximum price among all the stocks.

**Determine the occurrence count of a specific price:**

Write a method, countOccurrences, that takes the array of stock prices and a target price as input and returns the number of times the target price occurs in the array.

**Compute the cumulative sum of stock prices:**

Write a method, computeCumulativeSum, that takes the ArrayList of stock prices as input and returns a new ArrayList containing the cumulative sum of prices at each position.

**Solution Code (VSCode Snapshot)**



**Solution Code (VSCode Text)**

import java.util.ArrayList;

public class StockAnalysis {

// Method to calculate the average stock price

public static float calculateAveragePrice(float[] prices) {

// Initialize a variable to store the sum of prices

float sum = 0;

// Loop through each price in the array

for (float price : prices) {

// Add the price to the sum

sum += price;

}

// Calculate the average by dividing the sum by the number of prices

return sum / prices.length;

}

// Method to find the maximum stock price

public static float findMaximumPrice(float[] prices) {

// Initialize a variable to store the maximum price, assuming the first price is the maximum initially

float maxPrice = prices[0];

// Loop through each price in the array

for (float price : prices) {

// Update the maxPrice if the current price is greater

if (price > maxPrice) {

maxPrice = price;

}

}

// Return the maximum price found

return maxPrice;

}

// Method to count the occurrences of a specific price

public static int countOccurrences(float[] prices, float targetPrice) {

// Initialize a variable to store the count of occurrences

int count = 0;

// Loop through each price in the array

for (float price : prices) {

// Increment the count if the current price matches the target price

if (price == targetPrice) {

count++;

}

}

// Return the count of occurrences

return count;

}

// Method to compute the cumulative sum of stock prices

public static ArrayList<Float> computeCumulativeSum(ArrayList<Float> prices) {

// Initialize an ArrayList to store the cumulative sum

ArrayList<Float> cumulativeSum = new ArrayList<>();

// Initialize a variable to store the sum

float sum = 0;

// Loop through each price in the ArrayList

for (float price : prices) {

// Add the current price to the sum

sum += price;

// Add the cumulative sum to the ArrayList

cumulativeSum.add(sum);

}

// Return the ArrayList containing the cumulative sum

return cumulativeSum;

}

public static void main(String[] args) {

// Example usage:

// Define an array of stock prices

float[] pricesArray = {10.5f, 12.3f, 11.8f, 13.2f, 10.5f, 12.9f, 10.5f, 13.5f, 11.0f, 12.7f};

// Create an ArrayList from the array

ArrayList<Float> pricesArrayList = new ArrayList<>();

for (float price : pricesArray) {

pricesArrayList.add(price);

}

// Calculate average price

float averagePrice = calculateAveragePrice(pricesArray);

System.out.println("Average Price: " + averagePrice);

// Find maximum price

float maximumPrice = findMaximumPrice(pricesArray);

System.out.println("Maximum Price: " + maximumPrice);

// Count occurrences of a specific price

float targetPrice = 10.5f;

int occurrences = countOccurrences(pricesArray, targetPrice);

System.out.println("Occurrences of " + targetPrice + ": " + occurrences);

// Compute cumulative sum

ArrayList<Float> cumulativeSum = computeCumulativeSum(pricesArrayList);

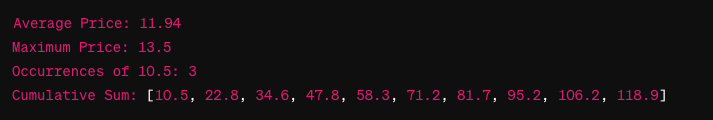
System.out.println("Cumulative Sum: " + cumulativeSum);

}

}

**Documentation**

1. **Class Declaration:** The code begins with the declaration of a Java class named `StockAnalysis`.
2. **Import Statement:** The `import java.util.ArrayList;` statement imports the `ArrayList` class from the `java.util` package, allowing us to use ArrayLists in our code.
3. **Method Definitions:**
   * + **`calculateAveragePrice`:** This method calculates the average of stock prices in an array. It takes a float array `prices` as input and returns the average price as a float.
     + **`findMaximumPrice`:** This method finds the maximum stock price in an array. It takes a float array `prices` as input and returns the maximum price as a float.
     + **`countOccurrences`:** This method counts the occurrences of a specific price in an array. It takes a float array `prices` and a target price `targetPrice` as input and returns the count of occurrences as an integer.
     + **`computeCumulativeSum`:** This method computes the cumulative sum of stock prices stored in an ArrayList. It takes an ArrayList of floats `prices` as input and returns another ArrayList containing the cumulative sum of prices at each position.
4. **Main Method:**
   * + The `main` method is the entry point of the program.
     + It demonstrates how to use the defined methods with example data.
     + It initializes an array of stock prices (`pricesArray`) and converts it into an ArrayList (`pricesArrayList`) for demonstration purposes.
     + It calls each of the defined methods with appropriate arguments and prints the results.
5. **Variable Types:**
   * + All methods use `float` data type for prices since stock prices typically include decimal points.
     + The `countOccurrences` method returns an `int` for the count of occurrences.
     + The `computeCumulativeSum` method returns an `ArrayList<Float>` containing the cumulative sum of prices.
6. **Looping Constructs:**
   * + Loops (`for` loops) are used to iterate over arrays and ArrayLists to perform calculations and operations.
     + They enable the calculation of the average, finding the maximum price, counting occurrences, and computing the cumulative sum efficiently by iterating through the elements.
7. **Conditional Statements:**
   * + Conditional statements (`if` statements) are used within loops to perform specific actions based on certain conditions.
     + They are employed to compare prices, update maximum price, and count occurrences of a target price.
8. **ArrayList Usage:**
   * + The code demonstrates how to work with ArrayLists by initializing an ArrayList from an array and performing cumulative sum operations.

**Output**

This output results of each method call:

* Average Price: The average price of the stock prices array is 11.94.
* Maximum Price: The maximum price among all the stocks is 13.5.
* Occurrences of 10.5: The target price 10.5 occurs 3 times in the stock prices array.
* Cumulative Sum: The cumulative sum of stock prices at each position in the ArrayList is provided as [10.5, 22.8, 34.6, 47.8, 58.3, 71.2, 81.7, 95.2, 106.2, 118.9].

**References**

OpenJDK. (n.d.). Java Platform, *Standard Edition 8 Documentation.*

*https://docs.oracle.com/javase/8/docs/api/*