Notes of CSE$_{110}$ Fall 2021

Atharv Sule

December 2, 2021

Contents

1 Auditorium suorce code

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;
import java.text.NumberFormat;

public class Auditorium {
    double[][] seats;
    double totalSales;
    int numSold;

    // default constructor
    public Auditorium ()  {
        seats = new double[3][4];
        // to view path with file in pathname and click on the file
        try {
            File inputFile = new File("seatPrices.txt");
            Scanner in = new Scanner(inputFile);
            while (in.hasNextDouble()){
                for(int i = 0; i < 3; i++ ){
                    for(int j = 0; j < 4; j++){
```

```
                    double value = in.nextDouble();
                    seats[i][j] = value;
               }
           }
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    System.out.println();
   totalSales = 0;
   numSold  = 0;
}

// gets the total price of the tickets sold
public String getTotal(){

    NumberFormat fmt = NumberFormat.getCurrencyInstance();
    fmt.format(totalSales);
    return "" +  fmt.format( totalSales);
}

public void displayChart(){
    for(int i = 0; i < 3; i++ ){
        for(int j = 0; j < 4; j++){
            System.out.print( seats[i][j] + "    ");;
        }
        System.out.println("");
    }
}

// used to sell tickets by setting ticket value to zero
public boolean sellTicket(int i, int j){
    for(int l = 0; l < 3; l++ ){
        for (int m = 0 ; m < 4; m++){
            if (( i == l) && (j == m)){
                if(seats[l][m] != 0){
                    totalSales = totalSales + seats[l][m];
                    numSold++;
                    seats[l][m] = 0.0;
                    return true;
                    }
                }
            }
        }
        return false;
    }

    // gets number of tickets sold
    public int numSold(){
        return numSold;
    }

    // checks if tickets are sold out or not
    public boolean soldOut(){
        for(int l = 0; l < 3; l++ ){
            for (int m = 0 ; m < 4; m++){
               if(seats[l][m] != 0){
                   return false;
               }
            }
        }
        return true;
    }
}
```

2 Sorting Algo's

```
import java.io.*;
import java.util.*;

public class Sorting {
    public static void main(String[] args) {
        int arr[ ] =  { 8, 6, 9, 3 ,4, 5 };

                // //  // selection sort
       // selectionSort(arr);

        System.out.println("");
        // //  // insertion sort
    String stringArr2 = Arrays.toString(insertionSort(arr));
    System.out.println(stringArr2);


    System.out.print(bianarySerch(insertionSort(arr), 4));

    System.out.println("");

}

private static int[] insertionSort(int[] arr) {
    for (int i = 1; i < arr.length; i++){
        int j  = i;
        while (j > 0 && (arr[j -1] > arr[j])){
            int tem = arr[j];
            arr[j] = arr[j - 1];
            arr[j - 1] = tem;
            j--;
        }

        String array2 = Arrays.toString(arr);
        System.out.println(array2);

    }


    return arr;
}

private static void selectionSort(int[] arr) {
    for(int j = 0; j < arr.length; j++ ){
        int min = j;
        for (int i = j + 1; i < arr.length; i++){
            if (arr[i] < arr[min]){
                min = i;
            }
        }
        if (min != j){
            int temp = arr[j];
            arr[j] = arr[min];
```

```
                arr[min] = temp;
            }
            String array1 = Arrays.toString(arr);
            System.out.println("Phase" + (j + 1) + ":" +  array1);
        }

        for (int m = 0; m < arr.length; m++){
            System.out.print(arr[m] + " ");
        }
    }

    private static int  bianarySerch(int[] arr, int num){
        int left = 0;
        int right = arr.length - 1;
        while (left <= right ){
            int mid = (left + right)/2;
            if (arr[mid] == num){
                return mid;
            }else if (num < arr[mid]){
                right = mid - 1;
            }else{
                left = mid + 1;
            }
        }

        return -1;
    }
}
```

```
[6, 8, 9, 3, 4, 5]
[6, 8, 9, 3, 4, 5]
[3, 6, 8, 9, 4, 5]
[3, 4, 6, 8, 9, 5]
[3, 4, 5, 6, 8, 9]
[3, 4, 5, 6, 8, 9]
[3, 4, 5, 6, 8, 9]
[3, 4, 5, 6, 8, 9]
[3, 4, 5, 6, 8, 9]
[3, 4, 5, 6, 8, 9]
```

```
[3, 4, 5, 6, 8, 9]
1
```