JOBSHEET 6

SORTING

Nama   :   Yuma Akhunza Kausar Putra

NIM   :   2341720259

A. Practicum 1 _Create Student Class_

```java
package Jobsheet6;

public class Students {
    String name;
    int entranceYear, age;
    double gpa;

    Students(String n, int y, int a, double g) {
        name = n;
        entranceYear = y;
        age = a;
        gpa = g;
    }

    void print() {
        System.out.println("Name = " + name);
        System.out.println("Entrance Year = " + entranceYear);
        System.out.println("Age = " + age);
        System.out.println("GPA = " + gpa);
    }
}
```

B. Practicum 2 _Create HighAchieverStudent Class_

```java
package Jobsheet6;

public class HighAchieverStudent {
    Students list[] = new Students[5];
    int idx;

    // add method
    void add(Students std) {
        if (idx < list.length) {
            list[idx] = std;
            idx++;
        } else {
            System.out.println(x:"The student list is already full-filled");
        }
    }

    // print method
    void print() {
        for (Students s : list) {
            s.print();
            System.out.println(x:"====================");
        }
    }

    // bubble sort method
    void bubblesort() {
        for (int i = 0; i < list.length - 1; i++) {
            for (int j = 1; j < list.length - i; j++) {
                if (list[j].gpa > list[j - 1].gpa) {

                    Students tmp = list[j];
                    list[j] = list[j - 1];
                    list[j - 1] = tmp;

                }
            }
        }
```

C. Practicum 3 *Create Main Class*

```java
package Jobsheet6;

import java.util.Scanner;

public class Main {
    Run | Debug
    public static void main(String[] args) {

        Scanner s1 = new Scanner(System.in);
        Scanner s2 = new Scanner(System.in);
        HighAchieverStudent data = new HighAchieverStudent();
        int n = 5;

        for (int i = 0; i < n; i++) {
            System.out.print(s:"Name = ");
            String name = s2.nextLine();
            System.out.print(s:"Entrance year = ");
            int year = s1.nextInt();
            System.out.print(s:"Age = ");
            int age = s1.nextInt();
            System.out.print(s:"GPA = ");
            double gpa = s1.nextDouble();

            Students s = new Students(name, year, age, gpa);
            data.add(s);

            System.out.println(x:"Unsorted student list: ");
            data.print();
            System.out.println(x:"Data mahasiswa setelah sorting desc berdasar ipk=");
            data.bubblesort();
            data.print();
            System.out.println(x:"Ascending Sorted student list");
            data.selectionSort();
            data.print();
        }
}
```

D. Practicum 4 *Add Selection Sort process in HighAchieverStudent Class*

```java
// selection sort method
void selectionSort() {
    for (int i = 0; i < list.length - 1; i++) {
        int idxMin = i;
        for (int j = i + 1; j < list.length; j++) {
            if (list[j].gpa < list[idxMin].gpa) {
                idxMin = j;
            }
        }
        // Swap
        Students tmp = list[i];
        list[idxMin] = list[i];
        list[i] = tmp;
```

Main

```java
System.out.println(x:"Ascending Sorted student list");
data.selectionSort();
data.print();
```

E. Practicum 5 *Add Insertion Sort process in HighAchieverStudent Class*

```
// Insertion sort method
void insertionsort() {
    for (int i - 1; i < idx; i++) {
        Students key = list[i];
        int j = i - 1;
        while (j >= 0 && list[j].gpa > key.gpa) {
            list[j + 1] = list[j];
            j = j -1;
        }
        list[j+1]= key;
    }
}
```

Main

```
System.out.println(x:"Ascending Sorted student list ");
data.insertionsort();
data.print();
```

Questions

1. In which class we have a function to do sorting with bubble sort approach?

```
17    void display() {
18        for (Students std : list) {
19            std.print();
20        }
21    }
22
23    // print method
24 >  void print() { ⋯
30
31    // bubble sort method
32    void bubblesort() {
33        for (int i = 0; i < list.length - 1; i++) {
34            for (int j = 1; j < list.length - i; j++) {
35                if (list[j].gpa > list[j - 1].gpa) {
36
37                    Students tmp = list[j];
38                    list[j] = list[j - 1];
39                    list[j - 1] = tmp;
40
41                }
42            }
43        }
44    }
45
```

1. In which class we have a function to do sorting with insertion sort approach?

```
// Insertion sort method
void insertionsort() {
    for (int i - 1; i < idx; i++) {
        Students key = list[i];
        int j = i - 1;
        while (j >= 0 && list[j].gpa > key.gpa) {
            list[j + 1] = list[j];
            j = j -1;
        }
        list[j+1]= key;
    }
}
```

2. What is the meaning of swapping process? Write the code to do the swapping process in the program above!

Swapping refers to the process of exchanging the positions of two elements in a list or array. This is done to reorder the elements based on a certain condition (e.g., in ascending or descending order).

```
// SWAP
Students tmp = list[j];
list[j] = list[j-1];
list[j-1] = tmp;
```

3. In bubbleSort(), there is these lines of code, what's the function of it?

```
29            if(list[j].gpa > list[j-1].gpa){
30                Student tmp = list[j];
31                list[j] = list[j-1];
32                list[j-1] = tmp;
33            }
```

It compares the GPA of two students and swaps their positions if the GPA of the student at index j is less than the GPA of the student at index j-1.

4. Look at the loops inside the bubbleSort() method:

```
27        for(int i=0; i<list.length-1; i++){
28            for(int j=1; j<list.length-i; j++){
```

   a. What's the difference of loop i and loop j?
      The loop indexed by i traverses each element within the array, commencing from the initial element and proceeding up to the secondto-last element. It governs the number of iterations or passes made through the array.

   b. Why is the criteria of loop i is i<listStd.length-1 ?
      The condition i < list.length - 1 prevents the loop from exceeding the array's last element. As each iteration correctly positions the larges unsorted element, there's no necessity to compare the final element with others.

   c. Why is the criteria of loop j is j <listStd.length-i ?
      The loop variable "j" traverses through the unsorted segment of the array. It initiates from index 1, which represents the second element, and continues until list.length - i. This approach ensures that comparisons are made solely within the unsorted section of the array during each iteration.

   d. If the data in listStd is 50, how many loop I will happen ? And how many bubble sort step will be ?
      If there are 50 data elements in listStd, the loop will iterate 49 times, and each iteration will involve one step of the bubble sort. This is because each iteration of the loop moves one element to its proper position from the end of the array.

5. In selection sort method, there is these lines of code, what's that for?

```
41        int idxMin = i;
42        for(int j=i+1; j<list.length; j++){
43            if(list[j].gpa < list[idxMin].gpa){
44                idxMin = j;
45            }
46        }
```

   This code snippet is finding the index of the minimum GPA in a list of objects. It iterates through the list, compares each GPA value, and updates the index of the minimum GPA found so far.

6. Change the insertionSort method so that the user has options to sort in either ascending or descending order. You can do it by adding a parameter, and this parameter's value will be assigned through function calling in main class

```
void insertionSort(boolean asc){
    for (int i = 0; i < list.length; i++) {
        Students temp = list[i];
        int j = i;
        if(asc){
            //Ascending algorithm here
        }else{
            //Descending algorithm here
        }
        list[j] = temp;
    }
}
```

Code:

```java
void insertionSort(boolean asc) {
    for (int i = 1; i < list.length; i++) {
        Students tmp = list[i];
        int j = i;
        if (asc) {
            while (j > 0 && list[j - 1].gpa > tmp.gpa) {
                list[j] = list[j - 1];
                j--;
            }
        } else {
            while (j > 0 && list[j - 1].gpa < tmp.gpa) {
                list[j] = list[j - 1];
                j--;
            }
        }
        list[j] = tmp;
    }
}
```

```java
System.out.println(x:"Sorted Student List based on GPA using insertion sort");
System.out.print(s:"1. Ascending 2. Descending: ");
int input = s1.nextInt();
data.insertionSort(
    input == 1 ? true : false
);
data.display();
```

Output:

```
Age = 290
GPA = 3.8065636085948222
Sorted Student List based on GPA using insertion sort
1. Ascending 2. Descending: 2
Name = 3a3ad0d6-6d52-4344-a39d-407653e56aa7
Entrance Year = 293
Age = 290
GPA = 3.8065636085948222
Name = b7a60b23-46fa-4124-8f07-3fc516123237
Entrance Year = 797
Age = 888
GPA = 3.0203804855641425
Name = f4c36b0d-98e7-4190-826b-a775c83a2088
Entrance Year = 981
Age = 214
GPA = 2.5919942668709415
Name = c3b232be-532a-4df8-8ef3-c9e3bf5def62
Entrance Year = 246
Age = 456
GPA = 2.532172885543012
Name = 4c81d18e-186b-4696-841d-f39edea26f0a
Entrance Year = 113
Age = 380
GPA = 1.2688803502024386
PS D:\Algoritma-Jobsheet>
```

Assignment

1. There is a company that provide services in airplane ticket sales, they are developing a backend system for ticket reservation. One of its features is to display all available tickets based on filter from user. The ticket list must able to be sorted by the price in ascending and descending order. Implement these class diagrams in java program and create the sorting algorithm with bubble sort and selection sort.

   Class :
   https://github.com/akhunzakp/semester-2/blob/main/jobsheet6/Tickets24.java

   https://github.com/akhunzakp/semester-2/blob/main/jobsheet6/TicketService24.java
   Main :
   https://github.com/akhunzakp/semester-2/blob/main/jobsheet6/MainTicket24.java

2. Premiere League in 2020 is already in half-season. In this season, Liverpool is the top of the list, the full list is displayed below Change the standings list above to class diagram that has sorting club function based on highest to smallest points (in ascending order) with insertion sort algorithm. Take these following class diagrams as your reference:

Class   :

https://github.com/akhunzakp/semester-2/blob/main/jobsheet6/PremierLeague24.java

https://github.com/akhunzakp/semester-2/blob/main/jobsheet6/PremierLeagueService24.java

Main   :

https://github.com/akhunzakp/semester-2/blob/main/jobsheet6/MainPremierLeague24.java