StudentInfo

```
package studentinfo;
import java.util.Scanner;

public class StudentInfo {
     String name;
     int matricNo;
    StudentInfo [] stud;

    public StudentInfo(int s){
        stud = new StudentInfo[s];
    }

    public StudentInfo (String name, int matricNo) {
        this.name = name;
        this.matricNo = matricNo;
    }

    public void setName (String name){
        this.name = name;
    }
    public String getName(){
        return name;
    }

    public void setMatricNo (int matricNo){
        this.matricNo = matricNo;
    }
    public int getMatricNo(){
        return matricNo;
    }

}
```

Student

```
package studentinfo;

import java.util.InputMismatchException;
import java.util.Scanner;
import static studentinfo.sorting.quickSorting;
import static studentinfo.sorting.selectionSort;
import static studentinfo.sorting.shellSort;

public class Student {

    static Scanner scan = new Scanner(System.in);
    static int count = 0;
    static int searchValue;

    public static void main(String[] args) {

        int choice;

        System.out.print("Please enter the number of student : ");
        int num = scan.nextInt();

        StudentInfo info = new StudentInfo(num);
        StudentInfo[] stud = new StudentInfo[num];

        do {

            menu();

            System.out.print("\nPlease enter your choice (1-4) or 0 to exit : ");
            choice = scan.nextInt();

            if (choice == 0) {
                System.out.println("Thank you and Bye.");
            } else {
                switch (choice) {
                    case 1:
                        count = add(stud, count);
                        break;

                    case 2:
                        delete(stud);
                        break;

                    case 3:
                        search(stud, count);
```

```
                break;

            case 4:
                display(stud);
                break;

            default:
                System.out.println("Sorry your choice is not in the list. Please enter again.");
        }
      }

   } while (choice != 0);

}

public static void menu() {
   System.out.println();
   System.out.println("********************MENU*********************");
   System.out.println("1. Add Student Data");
   System.out.println("2. Delete Student Data");
   System.out.println("3. Search Student Data");
   System.out.println("4. Display Student Data");
}

public static int add(StudentInfo[] stud, int count) {
   if (count < stud.length) {
      System.out.print("\nPlease enter student's name : ");
      scan.nextLine();
      String name = scan.nextLine();
      System.out.print("Please enter student's matric no : ");
      int matricNo = scan.nextInt();

      StudentInfo st = new StudentInfo(name, matricNo);
      stud[count++] = st;
   } else {
      System.out.println("List is Full.");
   }
   return count;
}

public static void delete(StudentInfo[] stud) {

   if (count == 0) {
      System.out.println("The list is empty.");
   } else {
```

```
        System.out.println("Which student you want to delete (in number 1 until " +
stud.length + ")?");
        int noList = scan.nextInt();
        System.out.println();

        if (stud[noList - 1] != null) {
            for (int x = noList - 1; x < count; x++) {
                stud[x] = stud[x + 1];
            }
            count--;
            System.out.println("Successful deleted.");
        } else {
            System.out.println("Sorry, the data is not found.");
        }
    }

}

public static void search(StudentInfo[] stud, int count) {
    int i;
    boolean searching = false;

    if (count == 0) {
        System.out.println("The list is empty.");
    } else {
        System.out.print("\nPlease enter the Matric Number you want to search: ");
        searchValue = scan.nextInt();

        for (i = 0; i < count; i++) {
            if (stud[i].getMatricNo() == searchValue) {
                searching = true;
                System.out.println("\nName: " + stud[i].getName());
                System.out.println("Matric No : " + stud[i].getMatricNo());
                break;
            }
        }
        if (searching == false) {
            System.out.println("Sorry the Matric No or the Name enter is not in the list.");

        }
    }
}

public static void display(StudentInfo[] stud) {

    if (count == 0) {
```

```
            System.out.println("The list is empty.");
        } else {
            System.out.println("<Before Sorting>");
            for (int i = 0; i < count; i++) {

                System.out.println();
                System.out.println("----------------");
                System.out.println("Student " + (i + 1));
                System.out.println("----------------");
                System.out.println(stud[i].getName());
                System.out.println(stud[i].getMatricNo());
                System.out.println();
            }
            System.out.println("**********************");
            System.out.println("1. Selection Sort");
            System.out.println("2. Shell Sort");
            System.out.println("3. Quick Sort");
            System.out.println("**********************");
            System.out.print("Please select the sorting techniques : ");
            String sort = scan.next();

            if (null == sort) {
                System.out.println("The number you enter is INVALID.");
            } else {
                switch (sort) {
                    case "1":
                        sorting.showSort(stud);
                        selectionSort(stud);
                        System.out.println("< After Sorting >");
                        for (int i = 0; i < count; i++) {
                            System.out.println();
                            System.out.println("----------------");
                            System.out.println("Student " + (i + 1));
                            System.out.println("----------------");
                            System.out.println(stud[i].getName());
                            System.out.println(stud[i].getMatricNo());
                            System.out.println();
                        }
                        break;

                    case "2":
                        shellSort(stud);
                        System.out.println("< After Sorting >");
                        for (int i = 0; i < count; i++) {

                            System.out.println();
```

```
                    System.out.println("----------------");
                    System.out.println("Student " + (i + 1));
                    System.out.println("----------------");
                    System.out.println(stud[i].getName());
                    System.out.println(stud[i].getMatricNo());
                    System.out.println();
                }
                break;
            case "3":
                sorting.showSort(stud);
                quickSorting(stud, 0, count);
                System.out.println("< After Sorting >");
                for (int i = 0; i < count; i++) {
                    System.out.println();
                    System.out.println("----------------");
                    System.out.println("Student " + (i + 1));
                    System.out.println("----------------");
                    System.out.println(stud[i].getName());
                    System.out.println(stud[i].getMatricNo());
                    System.out.println();
                }
                break;
            default:
                System.out.println("The number you enter is INVALID.");
                break;
        }
    }


  }
}

}
```

Sorting

```
package studentinfo;

import java.util.Scanner;
import static studentinfo.Student.count;

public class sorting {

   static Scanner scan = new Scanner(System.in);

   public static void selectionSort(StudentInfo[] stud) {

      int minIndex;
      int tempM;
      String tempN;
      for (int x = 0; x < count; x++) {
         minIndex = x;
         for (int y = x + 1; y < count; y++) {

            if (stud[y].getMatricNo() < stud[minIndex].getMatricNo()) {
               minIndex = y;
            }
         }
         tempM = stud[x].getMatricNo();
         stud[x].setMatricNo(stud[minIndex].getMatricNo());
         stud[minIndex].setMatricNo(tempM);

         tempN = stud[x].getName();
         stud[x].setName(stud[minIndex].getName());
         stud[minIndex].setName(tempN);

         for (int z = 0; z < count; z++) {
            System.out.print(stud[z].getMatricNo() + " ");
         }
         System.out.println();

      }

   } // selectionSort

   public static void shellSort(StudentInfo[] stud) {

      sorting.showSort(stud);
      for (int gap = count / 2; gap > 0; gap /= 2) {

         if (gap == 6) {
```

```
            int n = 1;
            System.out.println("\nSeperate to distance 6 (Before Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 6;
                }
                n++;
            }
            gap = 6;
        } else if (gap == 5) {
            int n = 1;
            System.out.println("\nSeperate to distance 5 (Before Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 5;
                }
                n++;
            }
            gap = 5;
        } else if (gap == 4) {
            int n = 1;
            System.out.println("\nSeperate to distance 4 (Before Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 4;
                }
                n++;
            }
            gap = 4;
        } else if (gap == 3) {
            int n = 1;
            System.out.println("\nSeperate to distance 3 (Before Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 3;
                }
                n++;
            }
```

```
            gap = 3;
        } else if (gap == 2) {
            int n = 1;
            System.out.println("\nSeperate to distance 2 (Before Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 2;
                }
                n++;
            }
            gap = 2;
        } else if (gap == 1) {
            System.out.println("\n Distance 1 (Before Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
            }
        } //display

        for (int i = gap; i < count; i++) {
            int temp = stud[i].getMatricNo();
            String temp2 = stud[i].getName();
            int j;
            for (j = i; j >= gap && stud[j - gap].getMatricNo() > temp; j -= gap) {
                stud[j].setMatricNo(stud[j - gap].getMatricNo());
                stud[j].setName(stud[j - gap].getName());
            }
            stud[j].setMatricNo(temp);
            stud[j].setName(temp2);

        }

        if (gap == 6) {
            int n = 1;
            System.out.println("\nRearrange by distance 6 (After Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 6;
                }
                n++;
            }
            gap = 6;
            System.out.println("----------------------------------------");
```

```
            sorting.showSort(stud);
        } else if (gap == 5) {
            int n = 1;
            System.out.println("\nRearrange by distance 5 (After Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 5;
                }
                n++;
            }
            gap = 5;
            System.out.println("-----------------------------------------");
            sorting.showSort(stud);
        } else if (gap == 4) {
            int n = 1;
            System.out.println("\nRearrange by distance 4 (After Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 4;
                }
                n++;
            }
            gap = 4;
            System.out.println("-----------------------------------------");
            sorting.showSort(stud);
        } else if (gap == 3) {
            int n = 1;
            System.out.println("\nRearrange by distance 3 (After Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 3;
                }
                n++;
            }
            gap = 3;
            System.out.println("-----------------------------------------");
            sorting.showSort(stud);
        } else if (gap == 2) {
            int n = 1;
            System.out.println("\nRearrange by distance 2 (After Sorting) :");
```

```
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
                if (n >= gap) {
                    System.out.println();
                    gap += 2;
                }
                n++;
            }
            gap = 2;
            System.out.println("----------------------------------------");
            sorting.showSort(stud);
        } else if (gap == 1) {
            System.out.println("\nRearrange by distance 1 (After Sorting) :");
            for (int k = 0; k < count; k++) {
                System.out.print(stud[k].getMatricNo() + "\t");
            }
            System.out.println("\n----------------------------------------");
            sorting.showSort(stud);
        }//display
    }
} //shellSort

public static void quickSorting(StudentInfo[] stud, int first, int last) {
    if (first < last) {
        int indeksPivot = choosePivot(stud, first, last);
        quickSorting(stud, first, indeksPivot - 1);
        quickSorting(stud, indeksPivot + 1, last);
    }
}//quickSorting

public static int choosePivot(StudentInfo[] stud, int first, int last) {

    int p = first;
    int pivot = stud[first].getMatricNo();
    String pivot2 = stud[first].getName();

    System.out.println("Pivot : " + stud[p].getMatricNo());
    for (int i = first + 1; i <= last; i++) {
        try {
            if (stud[i].getMatricNo() < pivot) {
                stud[p].setMatricNo(stud[i].getMatricNo());
                stud[i].setMatricNo(stud[p + 1].getMatricNo());
                stud[p + 1].setMatricNo(pivot);

                stud[p].setName(stud[i].getName());
                stud[i].setName(stud[p + 1].getName());
```

```
                stud[p + 1].setName(pivot2);

                p++;
              }
          } catch (NullPointerException ex) {
          } catch (ArrayIndexOutOfBoundsException ex) {
          }
       }

       for (int x = 0; x < count; x++) {
          if (stud[x].getMatricNo() < pivot) {
             System.out.print(stud[x].getMatricNo() + " ");
          }
       }
       System.out.print("\t");
       for (int x = first; x < count; x++) {
          if (stud[x].getMatricNo() > pivot) {
             System.out.print(stud[x].getMatricNo() + " ");
          }
       }
       System.out.println();
       return p;
    }// choosePivot

    public static void showSort(StudentInfo[] stud) {
       for (int z = 0; z < count; z++) {
          System.out.print(+stud[z].getMatricNo() + " ");
       }
       System.out.println();
    }
}
```

Display Student Data (Before Sorting)

```
4. Display Student Data

Please enter your choice (1-4) or 0 to exit : 4
<Before Sorting>

----------------
Student 1
----------------
a
5


----------------
Student 2
----------------
b
8


----------------
Student 3
----------------
c
2


----------------
Student 4
----------------
d
10


----------------
Student 5
----------------
e
6


----------------
Student 6
----------------
f
4


----------------
Student 7
----------------
g
3


----------------
Student 8
----------------
h
7


----------------
Student 9
----------------
i
1


----------------
Student 10
----------------
j
9

**********************
1. Selection Sort
2. Shell Sort
3. Quick Sort
**********************
```

The Steps of Selection Sort

```
Please select the sorting techniques : 1
5 8 2 10 6 4 3 7 1 9
1 8 2 10 6 4 3 7 5 9
1 2 8 10 6 4 3 7 5 9
1 2 3 10 6 4 8 7 5 9
1 2 3 4 6 10 8 7 5 9
1 2 3 4 5 10 8 7 6 9
1 2 3 4 5 6 8 7 10 9
1 2 3 4 5 6 7 8 10 9
1 2 3 4 5 6 7 8 10 9
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
```

The steps of Shell Sort

```
Please select the sorting techniques : 2
5 8 2 10 6 4 3 7 1 9

Seperate to distance 5 (Before Sorting) :
5       8       2       10      6
4       3       7       1       9

Rearrange by distance 5 (After Sorting) :
4       3       2       1       6
5       8       7       10      9
-----------------------------------------
4 3 2 1 6 5 8 7 10 9

Seperate to distance 2 (Before Sorting) :
4       3
2       1
6       5
8       7
10      9

Rearrange by distance 2 (After Sorting) :
2       1
4       3
6       5
8       7
10      9
-----------------------------------------
2 1 4 3 6 5 8 7 10 9

 Distance 1 (Before Sorting) :
2       1       4       3       6       5       8       7       10      9
Rearrange by distance 1 (After Sorting) :
1       2       3       4       5       6       7       8       9       10
-----------------------------------------
1 2 3 4 5 6 7 8 9 10
```

The Steps of Quick Sort

```
Please select the sorting techniques : 3
5 8 2 10 6 4 3 7 1 9
Pivot : 5
2 4 3 1          8 10 7 6 9
Pivot : 2
1        3 4 5 8 10 7 6 9
Pivot : 3
1 2      4 5 8 10 7 6 9
Pivot : 8
1 2 3 4 5 7 6    10 9
Pivot : 7
1 2 3 4 5 6      8 10 9
Pivot : 10
1 2 3 4 5 6 7 8 9
```

Example Output (After Sorting)

```
< After Sorting >

----------------
Student 1
----------------
i
1


----------------
Student 2
----------------
c
2


----------------
Student 3
----------------
g
3


----------------
Student 4
----------------
f
4


----------------
Student 5
----------------
a
5


----------------
Student 6
----------------
e
6


----------------
Student 7
----------------
h
7


----------------
Student 8
----------------
b
8


----------------
Student 9
----------------
j
9


----------------
Student 10
----------------
d
10
```