Maddison Kiefer

Dr. Schwartz

Advanced Java Programming

11/28/2023

**Project 6-1 Sorting and ArrayList**

**Source Code:**

// @author Maddison Kiefer

// This class represents a Student with basic information

public class Student {

// Attributes to store student information

// Roll number of the student

int rollno;

// Name of the student

String name;

// Address of the student

String address;

// Constructor to initialize a new Student object with the given parameters

public Student(int rollno, String name, String address) {

this.rollno = rollno;

this.name = name;

this.address = address;

}

// Override the toString() method to provide a custom string representation of the Student object

@Override

public String toString() {

return "Student [rollno = " + rollno + ", name = " + name + ", address = " + address + "]\n";

}

}

import java.util.Comparator;

// This class implements the Comparator interface to provide a custom comparison for sorting students by name

public class NameComparator implements Comparator<Student> {

// Override the compare method to define the custom comparison logic based on student names

@Override

public int compare(Student s1, Student s2) {

// Compare student names using the compareTo method of the String class

return s1.name.compareTo(s2.name);

}

}

import java.util.Comparator;

// This class implements the Comparator interface to provide a custom comparison for sorting students by roll number

public class RollnoComparator implements Comparator<Student> {

// Override the compare method to define the custom comparison logic based on student roll numbers

@Override

public int compare(Student s1, Student s2) {

// Compare student roll numbers by subtracting s2.rollno from s1.rollno

return s1.rollno - s2.rollno;

}

}

import java.util.ArrayList;

import java.util.Comparator;

// This class provides a generic implementation of the Selection Sort algorithm

public class SelectionSort {

// The sort method performs in-place selection sort on the input ArrayList using the provided comparator

public static <T> void sort(ArrayList<T> list, Comparator<? super T> comparator) {

// Get the size of the list

int n = list.size();

// Iterate through the list, up to the second-to-last element

for (int i = 0; i < n - 1; i++) {

// Assume the current index is the minimum index

int minIndex = i;

// Iterate through the unsorted portion of the list to find the index of the minimum element

for (int j = i + 1; j < n; j++) {

// Compare elements using the provided comparator

if (comparator.compare(list.get(j), list.get(minIndex)) < 0) {

// If the current element is smaller, update the minimum index

minIndex = j;

}

}

// Swap the minimum element with the element at the current index

T temp = list.get(minIndex);

list.set(minIndex, list.get(i));

list.set(i, temp);

}

}

}

import java.util.ArrayList;

// Main class for the program

public class Main {

public static void main(String[] args) {

// Create an ArrayList to store instances of the Student class

ArrayList<Student> students = new ArrayList<>();

// Add student objects to the ArrayList with different information

students.add(new Student(1, "Bob", "Los Angeles"));

students.add(new Student(3, "Evan", "Chicago"));

students.add(new Student(5, "Alice", "New York"));

students.add(new Student(2, "David", "Houston"));

students.add(new Student(4, "Charlie", "Miami"));

students.add(new Student(6, "Kaylee", "Detroit"));

students.add(new Student(8, "John", "Plymouth"));

students.add(new Student(7, "Adam", "Austin"));

students.add(new Student(9, "Taylor", "Orlando"));

students.add(new Student(10, "Sam", "San Diego"));

// Display the original list of students

System.out.println("Original List:");

System.out.println(students);

// Sort the list of students by name using the NameComparator

System.out.println("\nSorted by Name:");

SelectionSort.sort(students, new NameComparator());

System.out.println(students);

// Sort the list of students by roll number using the RollnoComparator

System.out.println("\nSorted by Roll Number:");

SelectionSort.sort(students, new RollnoComparator());

System.out.println(students);

}

}

**Executing the Application:**

