Assignment Questions on Implementing Comparators

1. ArrayList

1. **Sorting Books by Multiple Criteria**:
    - Write a Java program to create a Book class with properties id, title, and publicationYear. Implement two Comparator classes:
        1. **ByPublicationYearComparator**: Sort books by their publication year in ascending order.
        2. **ByTitleComparator**: Sort books by their titles in alphabetical order.
    - Create an ArrayList of Book objects, add five books with different titles and publication years, and use both comparators to sort the list. Display the sorted lists.

2. **Sorting Employees by Salary and Department**:
    - Write a Java program to create an Employee class with properties employeeId, name, and salary. Implement two Comparator classes:
        1. **BySalaryComparator**: Sort employees by their salary in descending order.
        2. **ByDepartmentComparator**: Sort employees by their department names in alphabetical order.
    - Create an ArrayList of Employee objects, add five employees with different salaries and department names, and use both comparators to sort the list. Display the sorted lists.

2. Set

1. **Sorting Products by Price and Category**:
    - Write a Java program to create a Product class with properties productId, productName, and price. Implement two Comparator classes:
        1. **ByPriceComparator**: Sort products by their prices in ascending order.
        2. **ByCategoryComparator**: Sort products by their categories in alphabetical order.
    - Create a TreeSet of Product objects with a custom comparator to sort products by price. Add five products to the set and display them.
    - Modify the TreeSet to use the ByCategoryComparator and display the products sorted by category.

CompilePanda

2. **Sorting Students by Grade and Name**:

    - Write a Java program to create a Student class with properties id, name, and grade. Implement two Comparator classes:
        1. **ByGradeComparator**: Sort students by their grades in ascending order.
        2. **ByNameComparator**: Sort students by their names in alphabetical order.
    - Create a TreeSet of Student objects with a custom comparator to sort students by grade. Add five students to the set and display them.
    - Modify the TreeSet to use the ByNameComparator and display the students sorted by name.

3. Map

1. **Sorting Employees by Salary and Name**:
    - Write a Java program to create an Employee class with properties employeeId, name, and salary. Implement two Comparator classes:
        1. **BySalaryComparator**: Sort employees by their salaries in ascending order.
        2. **ByNameComparator**: Sort employees by their names in alphabetical order.
    - Create a HashMap with Integer keys (employee IDs) and Employee values. Add five employees to the map.
    - To display the employees sorted by salary, convert the map values to a List and sort it using the BySalaryComparator.
    - To display the employees sorted by name, convert the map values to a List and sort it using the ByNameComparator.

2. **Sorting Products by Price and Name**:
    - Write a Java program to create a Product class with properties productId, productName, and price. Implement two Comparator classes:
        1. **ByPriceComparator**: Sort products by their prices in ascending order.
        2. **ByNameComparator**: Sort products by their names in alphabetical order.
    - Create a HashMap with String keys (product codes) and Product values. Add five products to the map.
    - To display the products sorted by price, convert the map values to a List and sort it using the ByPriceComparator.
    - To display the products sorted by name, convert the map values to a List and sort it using the ByNameComparator.

CompilePanda