**package** com;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** SortComparators {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter number of Employee records you want to store");

**int** n = sc.nextInt();

**boolean** flag = **true**;

**//Driver class**

List<Employee> employee = **new** ArrayList<>();

System.***out***.println("Enter employee details :");

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

employee.add(**new** Employee(0, **null**, 0));

}

**do** {

System.***out***.println("choose how you want to sort : ");

sc.nextLine();

System.***out***.println("1. ById\n2. ByName\n3. BySalary\n4. Exit");

**int** choice = sc.nextInt();

**switch** (choice) {

**case** 1:

System.***out***.println("in 1. Ascending OR 2. Descending");

**int** ch = sc.nextInt();

**if** (ch == 1)

Collections.*sort*(employee, **new** SortByIdAsc());

**else** **if** (ch == 2) **//Sorting By Id**

Collections.*sort*(employee, **new** SortByIdDsc());

**else** {

System.***out***.println("Wrong choice");

Continue;

}

**break**;

**case** 2:

System.***out***.println("in 1. Ascending OR 2. Descending");

**int** ch1 = sc.nextInt();

**if** (ch1 == 1)

Collections.*sort*(employee, **new** SortByNameAsc()); **//Sorting By Name**

**else** **if** (ch1 == 2)

Collections.*sort*(employee, **new** SortByNameDsc());

**else** {

System.***out***.println("Wrong choice");

Continue;

}

**break**;

**case** 3:

System.***out***.println("in 1. Ascending OR 2. Descending");

**int** ch2 = sc.nextInt();

**if** (ch2 == 1)

Collections.*sort*(employee, **new** SortBySalaryAsc()); **//Sorting By Salary**

**else** **if** (ch2 == 2)

Collections.*sort*(employee, **new** SortBySalaryDsc());

**else** {

System.***out***.println("Wrong choice");

Continue;

}

**break**;

**case** 4:

System.***out***.println("Thank you !! Have a Nice Day ");

value = **false**;

System.*exit*(0);

**break**;

**default**:

System.***out***.println("Wrong choice,choose proper option");

}

employee.forEach(System.***out***::println);

} **while** (flag);

}

}

**class** SortByIdAsc **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** o1.getId() - o2.getId();

}

}

**class** SortByIdDsc **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** o2.getId() - o1.getId();

}

}

**class** SortByNameAsc **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** o1.getName().compareTo(o2.getName());

}

}

**class** SortByNameDsc **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** o2.getName().compareTo(o1.getName());

}

}

**class** SortBySalaryAsc **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** (**int**) (o1.getSalary() - o2.getSalary());

}

}

**class** SortBySalaryDsc **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** (**int**) (o2.getSalary() - o1.getSalary());

}

}