**Employee Class:**

public class Employee {  
 String EmployeeID;  
 String Name;  
 int age;  
 int salary;  
 public Employee(String employeeID, String name, int age, int salary) {  
 EmployeeID = employeeID;  
 Name = name;  
 this.age = age;  
 this.salary = salary;  
 }  
 public String getEmployeeID() {  
 return EmployeeID;  
 }  
 public String getName() {  
 return Name;  
 }  
 public int getAge() {  
 return age;  
 }  
 public int getSalary() {  
 return salary;  
 }  
}

**EmployeeList Class:**

public class EmployeeList {  
 private int maxSize;  
 private int position;  
 private static Employee[] *ListEntry*;  
 EmployeeList(int size) {  
 maxSize = size;  
 *ListEntry* = new Employee[maxSize];  
 position = -1;  
 }  
 boolean IsListEmpty() {  
 return position == 1;  
 }  
 boolean IsListFull() {  
 return (position == maxSize - 1);  
 }  
 int ListSize() {  
 return (position + 1);  
 }  
 void InsertLast(Employee x) {  
 if (IsListFull())  
 System.*out*.println("Attempt to insert at the end of a full list");  
 else  
 *ListEntry*[++position] = x;  
 }  
 Employee RetrieveList(int p) {  
 int i;  
 Employee element;  
 if (IsListEmpty()) {  
 System.*out*.println("Attempt to retrieve an entry from an empty");  
 return null;  
 } else if (p < 0 || p >= ListSize()) {  
 System.*out*.println("attempt to retrieve an entry at a position not in the");  
 return null;  
 } else {  
 element = *ListEntry*[p];  
 return element;  
 }  
 }  
 void TraverselList() {  
 int i;  
 System.*out*.println("|\tEmployeeId\t|\tName\t\t\t|\tAge\t|\tSalary\t|");  
 for (i = 0; i < position + 1; i++) {  
 System.*out*.println("|\t" + *ListEntry*[i].getEmployeeID() +  
 "\t\t|\t" + *ListEntry*[i].getName() + "\t\t|\t" + *ListEntry*[i].getAge() +  
 "\t|\t" + *ListEntry*[i].getSalary() + "\t|");  
 }  
 }  
 void BubbleSort() {  
 int n = *ListEntry*.length;  
 Employee temp = null;  
 for (int i = 0; i < n; i++) {  
 for (int j = 1; j < (n - i); j++) {  
 if (*ListEntry*[j - 1].getSalary() >  
 *ListEntry*[j].getSalary()) {  
 temp = *ListEntry*[j - 1];  
 *ListEntry*[j - 1] = *ListEntry*[j];  
 *ListEntry*[j] = temp;  
 }  
 }  
 }  
 }  
 public Employee[] searchEmployeesByAge(int targetAge) {  
 Employee[] result = new Employee[maxSize];  
 int resultIndex = 0;  
 for (int i = 0; i <= position; i++) {  
 if (*ListEntry*[i].getAge() == targetAge) {  
 result[resultIndex++] = *ListEntry*[i];  
 }  
 }  
 if (resultIndex > 0) {  
 Employee[] trimmedResult = new Employee[resultIndex];  
 System.*arraycopy*(result, 0, trimmedResult, 0, resultIndex);  
 return trimmedResult;  
 } else {  
 return null;  
 }  
 }  
}

**Main Class:**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc =new Scanner(System.*in*);  
 System.*out*.print("Enter Employee count: ");  
 int count = sc.nextInt();  
 EmployeeList list = new EmployeeList(count);  
 sc.nextLine();  
 for (int i=0;i<count;i++) {  
 System.*out*.print("Enter Employee Id : ");  
 String employeeID = sc.nextLine();  
 System.*out*.print("Enter Name: ");  
 String name = sc.nextLine();  
 System.*out*.print("Enter Age: ");  
 int age = sc.nextInt();  
 sc.nextLine();  
 System.*out*.print("Enter Salary: ");  
 int salary = sc.nextInt();  
 sc.nextLine();  
 list.InsertLast(new Employee(employeeID, name, age, salary));  
 }  
 list.TraverselList();  
 list.BubbleSort();  
 System.*out*.print("\nAfter Sorting: ");  
 list.TraverselList();  
 System.*out*.println();  
 System.*out*.print("Enter Age of Employee: ");  
 int age = sc.nextInt();  
 Employee[] result = list.searchEmployeesByAge(age);  
 if (result != null) {  
 System.*out*.println("Employee with Age " + age + " : ");  
 for (Employee employee : result) {  
  
 System.*out*.println("|\t"+employee.getEmployeeID()+"\t\t|\t"+employee.getName( )+"\t\t|\t"+employee.getAge()+"\t|\t"+employee.getSalary()+"\t|");  
 }  
 } else {  
 System.*out*.println("No Employee found with age " + age);  
 }  
 }  
}

**Output:**

