

Class Employee

```
public class Employee {

    private int id;
    private String name;
    private String gender;
    private int age;

    public Employee(int id, String name, String gender, int age) {
        this.id = id;
        this.name = name;
        this.gender = gender;
        this.age = age;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getGender() {
        return gender;
    }

    public int getAge() {
        return age;
    }

    public void display() {
        System.out.println("Employee id: " + id);
        System.out.println("Employee name: " + name);
        System.out.println("Employee gender: " + gender);
        System.out.println("Employee age: " + age);
    }
}
```

Class Node

```
public class Node {  
  
    private Employee data;  
    private Node next;  
  
    public Node(Employee data) {  
        this.data = data;  
    }  
  
    public Employee getData() {  
        return data;  
    }  
  
    public void setData(Employee data) {  
        this.data = data;  
    }  
  
    public Node getNext() {  
        return next;  
    }  
  
    public void setNext(Node next) {  
        this.next = next;  
    }  
  
}
```

Class LinkedListOfEmployee

```
public class LinkedListOfEmployee {

    private Node head;

    public LinkedListOfEmployee(){
        head = null;
    }

    public void insertAtEnd(Employee e){
        Node newNode = new Node(e);
        if(head == null){
            head = newNode;
            return;
        }
        Node current = head;
        while(current.getNext() != null){
            current = current.getNext();
        }
        current.setNext(newNode);
    }

    public void insertAtFront(Employee e){
        Node newNode = new Node(e);
        newNode.setNext(head);
        head = newNode;
    }

    public int countEmployees(String g){
        int count = 0;
        Node current = head;
        while(current != null){
            if(current.getData().getGender().equalsIgnoreCase(g))
                count++;
            current = current.getNext();
        }
        return count;
    }
}
```

```

public Employee getOldest(){
    if(head == null)
        return null;
    Employee oldest = head.getData();
    Node current = head;
    while(current != null){

        if(current.getData().getAge() > oldest.getAge())
            oldest = current.getData();

        current = current.getNext();
    }

    return oldest;
}

public LinkedListOfEmployee getEmployees(String g){
    LinkedListOfEmployee temp = new LinkedListOfEmployee();
    Node current = head;
    while(current != null){
        if(current.getData().getGender().equalsIgnoreCase(g))
            temp.insertAtEnd(current.getData());

        current = current.getNext();
    }

    return temp;
}

public void split(LinkedListOfEmployee male, LinkedListOfEmployee female){
    Node current = head;
    while(current != null){
        if(current.getData().getGender().equalsIgnoreCase("Male"))
            male.insertAtEnd(current.getData());
        else
            female.insertAtFront(current.getData());
        current = current.getNext();
    }
}

public void display(){
    Node current = head;
    while(current != null){
        current.getData().display();
        current = current.getNext();
    }
}
}

```

Class test

```
import java.util.Scanner;

public class test {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        LinkedListOfEmployee list = new LinkedListOfEmployee();
        int choice = 0;
        do {
            System.out.println("1- Add a new emmployee.");
            System.out.println("2- Get number of employees of a given gender.");
            System.out.println("3- Display the oldest employee.");
            System.out.println("4- To display all employees of a given gender.");
            System.out.println("5- To view all Male employees in the LinkedList.");
            System.out.println("0- Exit.");
            System.out.print("Enter choice: ");
            choice = input.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter employee id, name, gender, and age: ");
                    Employee e = new Employee(input.nextInt(),
input.next(), input.next(), input.nextInt());
                    list.insertAtEnd(e);
                    System.out.println("Employee Added Successfully");
                    break;
                case 2:
                    System.out.print("Enter the given gender: ");
                    String g = input.next();
                    System.out.println("Number of " + g + " employees: "
+ list.countEmployees(g));
                    break;
                case 3:
                    Employee emp = list.getOldest();
                    if (emp == null)
                        System.out.println("The list is empty");
                    else {
                        System.out.println("The oldest employee is:");
                        emp.display();
                    }
                    break;
                case 4:
                    System.out.print("Enter the given gender: ");
                    String gen = input.next();
                    int count = list.countEmployees(gen);
                    if (count == 0) {
                        System.out.println("There are no employees of the given gender!");
                    } else {
                        LinkedListOfEmployee tempList = list.getEmployees(gen);
                        System.out.println("All " + gen + " employees: ");
                    }
            }
        } while (choice != 0);
    }
}
```

```

templList.display();
    }
    break;
    case 5:
        int countM = list.countEmployees("Male");
        if (countM == 0) {
System.out.println("There are no male employees!");
        } else {
LinkedListOfEmployee male = new LinkedListOfEmployee();
LinkedListOfEmployee female = new LinkedListOfEmployee();
list.split(male, female);
System.out.println("All Male employees: ");
male.display();
        }
        break;
    case 0:
        System.out.println("GoodBye!");
        break;
    default:
        System.out.println("Invalid choice!");
    }
} while (choice != 0);
}
}

```