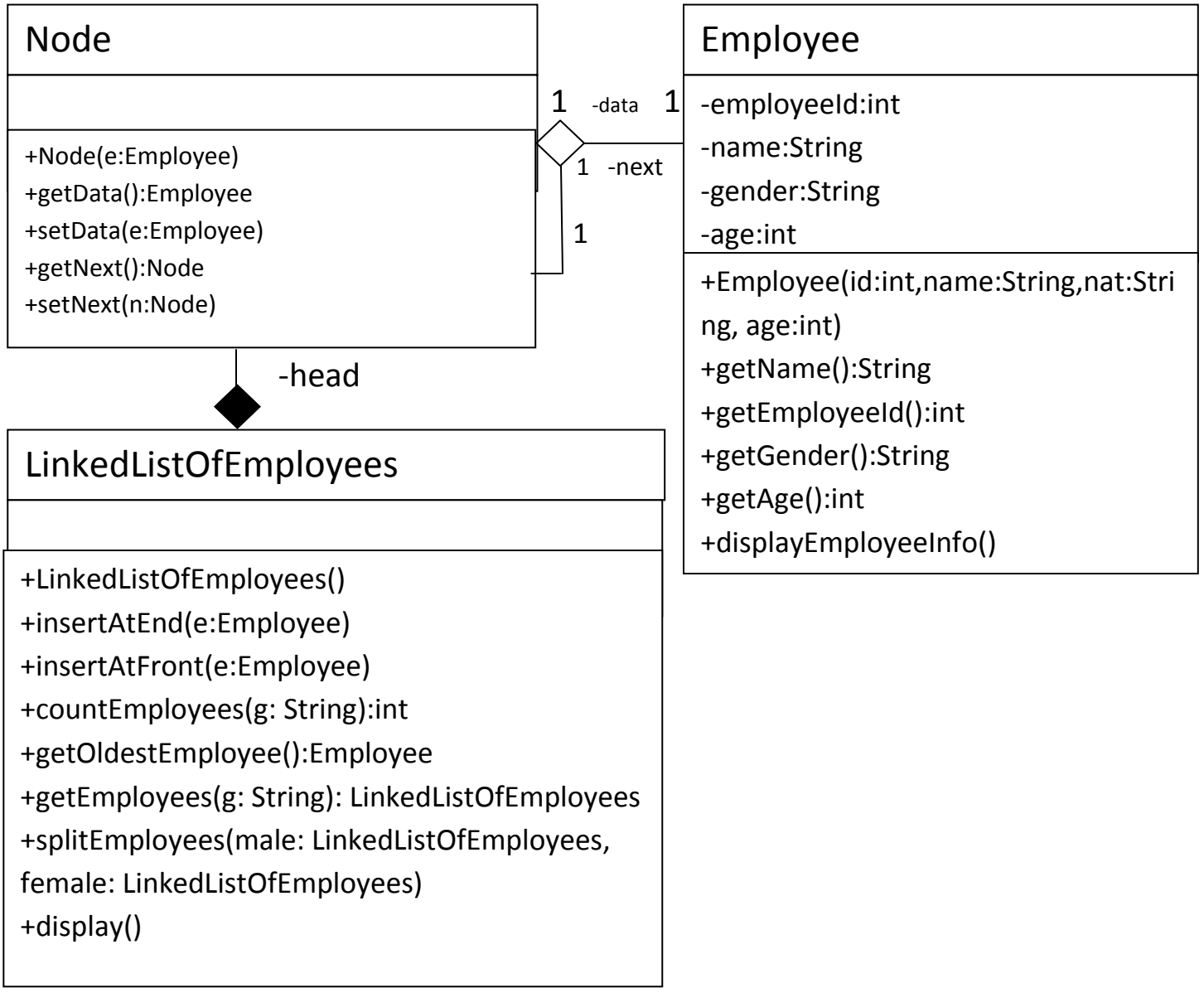
Create the classes along with the functionality given in the following UML Diagram. To understand the problem, please refer to the description given after the diagram.



Employee Class:

- Attributes:
  - employeeId: unique id of the employee.
  - ***name***: the name of the employee.
  - ***gender***: The gender of the employee.
  - ***Age***: The age of the employee
- Methods:
  - ***Employee(id:int,name: string, gender: String, age: int)***: constructor
  - ***displayEmployeeinfo()***: this method displays all the attributes of the employee.
  - ***Getters***

Node Class:

- Attributes:
- Methods:
  - ***Node(e: Employee)***: constructor
  - ***Getters/Setters***

LinkedListOfEmployees Class:

- Attributes:

- Methods:
  - ***LinkedListOfEmployees***: constructor
  - ***insertAtFront(g:Employee):*** this method inserts an employee at the front of linked list.
  - ***insertAtEnd(g:Employee):*** this method inserts an employee at the end of linked list.
  - ***countEmployee(g:String)***: this method returns the number of employees who are of gender ***g***.
  - ***getOldestEmployee():*** this method returns the oldest employee.
  - ***getEmployees(g:String):*** this method returns a linked list of all employees who are of gender ***g***.
  - ***splitEmployees(male: LinkedListOfEmployees, female: LinkedListOfEmployees):*** This method inserts all Male employees linked list male and Female in to female linked list.
  - ***display():*** displays the data of each employee in the linked list

Write a java program that will display following Menu to execute different member functions of these classes.

To add a new employee, Enter 1

To get the number of employees of a given gender, Enter 2

To get and display the oldest employee, Enter 3

To get and display all employees of a given gender, enter 4

To view all Male employees in the Linked List, Enter 5

To Exit, Enter 0

Enter Your Option:____

```
public class Employee {
	private int empId;
	private String name;
	private String gender;
	private int age;



	public Employee(int empId, String name, String gender, int age) {

		this.empId = empId;
		this.name = name;
		this.gender = gender;
		this.age = age;
	}


	public int getEmpId() {
		return empId;
	}


	public void setEmpId(int empId) {
		this.empId = empId;
	}


	public String getName() {
		return name;
	}


	public void setName(String name) {
		this.name = name;
	}


	public String getGender() {
		return gender;
	}


	public void setGender(String gender) {
		this.gender = gender;
	}
```

```
	public int getAge() {
		return age;
	}


	public void setAge(int age) {
		this.age = age;
	}

	public void displayEmployeeInfo()
	{
		System.out.println("Employee Id: "+empId);
		System.out.println("Employee Name: "+name);
		System.out.println("Employee Gender: "+gender);
		System.out.println("Employee Age: "+age);

	}

}
```

//

```
public class Node {

	private Employee data;
	private Node next;
	public Node(Employee data) {

		this.data = data;
		this.next = null;
	}
	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;
	}
```

```
}

//////////////////////////////////////////////////////////////


public class LinkedListOfEmployees {
	private Node head;


	public LinkedListOfEmployees() {
		head = null;
	}

	public void insertAtFront ( Employee e) {

		Node newNode = new Node(e);
		newNode.setNext(head);
		head = newNode;

	}

	public void insertAtBack ( Employee e) {


		Node newNode = new Node(e);
		if(head==null) {
			head = newNode;
			}
			else {
			Node curr = head;
			while (curr.getNext()!=null){
			curr = curr.getNext();
			}
			curr.setNext(newNode);
			}
	}

	public int countEmployees(String g) {
		int n = 0;
		Node current = head;
```

```
                while (current != null) {
                        if
(current.getData().getGender().equalsIgnoreCase(g))
                                n++;
                        current = current.getNext();
                }

        return n;
    }


    Employee getOldestEmployee()
    {
        if(head==null)
            return null;

        Node current = head;
        Employee oldest=current.getData();
        current = current.getNext();

        while (current != null) {
            if (current.getData().getAge()>oldest.getAge())
                oldest=current.getData();

            current = current.getNext();
        }


        return oldest;


    }



    public LinkedListOfEmployees getEmployees(String g)
    {
        LinkedListOfEmployees newList=new LinkedListOfEmployees();

        Node current = head;

        while (current != null) {

            if (current.getData().getGender().equalsIgnoreCase(g))
                newList.insertAtBack(current.getData());

            current = current.getNext();
        }

        return newList;
```

```
	}

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

	public void display()
	{
		Node current = head;

		while (current != null) {
			current.getData().displayEmployeeInfo();
			current = current.getNext();
		}
	}

}
```

//

```
import java.util.Scanner;
public class Main {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//This program assumes single word String input.

		Scanner input=new Scanner(System.in);
		LinkedListOfEmployees list=new LinkedListOfEmployees();
		int choice;
		String gender;
		int nEmployees;

		do
		{

		System.out.println("To Add a new Employee Enter 1 ");
		System.out.println("To get the number of Employees of given
gender Enter 2 ");
		System.out.println("To get and display oldest Employee Enter 3
");
		System.out.println("To get and display all Employees of a given
gender Enter 4 ");
		System.out.println("To view all Male Employees Enter 5 ");
		System.out.println("To Exit Enter 0 ");
		System.out.println("Enter Option___");
		choice=input.nextInt();

		switch(choice)
		{
		case 1:
			System.out.println("Enter Id,Name,Gender and Age of
Employee ");
			Employee e=new
Employee(input.nextInt(),input.next(),input.next(),input.nextInt());
			list.insertAtBack(e);
			System.out.println("Guest Added Successfully");

			break;
		case 2:
			System.out.println("Enter Gender");
			gender=input.next();
			System.out.println("No Of "+gender+" Employees are
"+list.countEmployees(gender));
			break;
```

```
                case 3:
                      Employee oldest=list.getOldestEmployee();
                      if(oldest!=null)
                      {
                            System.out.println("Oldest Employee in the List is
");
                            oldest.displayEmployeeInfo();
                      }
                      else
                            System.out.println("List is Empty");

                      break;
                case 4:
                      System.out.println("Enter Gender");
                      gender=input.next();
                      nEmployees=list.countEmployees(gender);
                      if(nEmployees==0)
                            System.out.println("No Employee of this Gender
Found");
                      else
                      {
                            LinkedListOfEmployees l=list.getEmployees(gender);
                            l.display();

                      }
                      break;

                case 5:
                      nEmployees=list.countEmployees("Male");
                      if(nEmployees==0)
                            System.out.println("No Employee of Male Gender
Found");
                      else
                      {
                            LinkedListOfEmployees l=list.getEmployees("Male");
                            l.display();
                      }
                      break;




                }

                }

                while(choice !=0);


          }
```

```
}
```