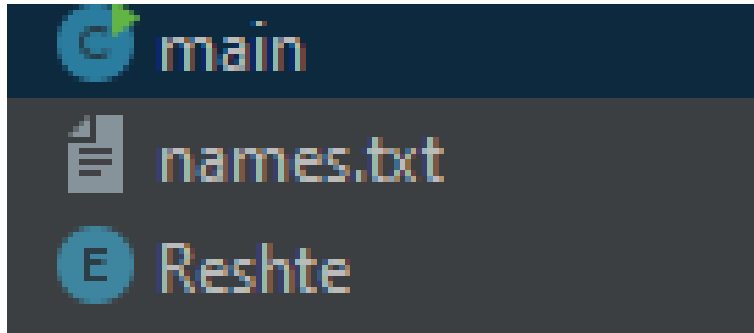


علیرضا سعیدنیا : پروژه چهار توضیحات

۴۰۰۱۰۸۳۳

ابتدا باید بباییم و برای رشته ها اینام و برای اسم ها یک فایل تی اکس تی در نظر بگیریم من هم ادم و همینکار را کردم



```
public Node head=new Node( model: 0, name: "", reshte: null, tedadVahed: 0) ;
```

• گره راسی را کاری میکنم که تا حد امکان همه فیلدهای داده ایش نال باشد و اینکار را کردم چون اینتیجر نمیشد نال باشد در جاوا .
[گرفتم \(منبع استک اورفلو\)](#)

سپس اومدم و کلاس نود را ساختم که شامل فیلد داده ای معدل و تعداد واحد و اسم و رشته است و همچنین هر نود شامل نود قبلی و نود بعدی است ، خب حالا نوبت به مرحله ساختن لیست پیوندی دو طرفه میرسد قبلش کانستراکتور و متد تو استرینگ را میسازیم تا بباییم و همه فیلدهای داده ای که میخواهیم را پرینت کنیم

```

public class Node {

    @Override
    public String toString() {
        return "Node{" + "moadel=" + moadel + ", name=\n\n'"
            + name + '\n' + ", reshte=" + reshte + ", tedadVahed=\n" + tedadVahed;
    }

    double moadel;
    String name;
    Reshte reshte;
    double tedadVahed;
    Node next;
    Node prev;
    public Node(double moadel, String name, Reshte reshte, double tedadVahed) {
        this.moadel = moadel;
        this.name = name;
        this.reshte = reshte;
        this.tedadVahed = tedadVahed;
    }

    public Node() {
    }
}

```

میاییم و از کلاس رندوم استفاده میکنیم و برای خواندن اسم ها از کلاس فایل ریدر استفاده میکنم و با ان از کلاس بافر رد ریدر هم استفاده میکنیم تا فایل تی اکستی را تک تک و خط به خط بخوانیم و میدانیم استرینگ است و در لیست پیوندی دو طرفه ۱۰۰ تا نود داریم پس میاییم از بین ۰ تا ۱۰۰ با استفاده از کلاس رندوم کلمات را میفرستیم به عنوان فیلد داده ای نود، ولی قبلش باید آنها را جایی ذخیره کرده باشیم ، ابتدا آنها را در یک آرایه ای از رشته ها ذخیره میکنیم (فور لوپ اول) و سپس میاییم در فور لوپ دوم آنها را به کانستراکتور میدهیم . برای رشته هم میاییم ان را داخل اری لیست میریزیم و با استفاده از کلاس رندوم میاییم و ان را انتخاب میکنیم من رشته های محدودی را در نظر گرفتم و برای معدل و واحد های برداشته شده هم یک متغیر میسازیم از جنس رندوم و میدانیم واحد کمتر از ۲۴ است و معدل تا ۲۰ جای دارد برود و آنها را میفرستیم به متد ادمون . اینکار هایی که گفتیم را ۹۹+۱ بار انجام میدهیم ، اولین بار برای تعیین همدان و بارهای بعدی برای ادامه دادن درج در لیست پیوندی دو طرفه مان (۹۹ دفعه بعدی)

```

public static void main(String[] args) throws IOException {
    try {
        main cl=new main();
        Random moad=new Random( seed: 20);
        FileReader fileReader=new FileReader( fileName: "src/names.txt");
        BufferedReader bufferedReader=new BufferedReader(fileReader);
        Reshte reshte;
        ArrayList<Reshte>arrayList=new ArrayList<>();
        arrayList.add(Reshte.BARGH);
        arrayList.add(Reshte.COMPUTER);
        arrayList.add(Reshte.ENSANI);
        arrayList.add(Reshte.HONAR);
        arrayList.add(Reshte.MEMARI);
        arrayList.add(Reshte.PHYISIC);
        arrayList.add(Reshte.SHIMI);
        arrayList.add(Reshte.TAJROBI);
        Random resht=new Random( seed: 7);
        Random vahed=new Random( seed: 24);
        String[]s=new String[100];
        for(int i=0;i<100;i++){
            s[i]=bufferedReader.readLine();
        }
        Random name=new Random();
        Node nnn=cl.add(moad.nextInt( origin: 0, bound: 20), s[name.nextInt( bound: 100)],arrayList.get(resht.nextInt( origin: 0, bound: 7)),vahed.nextInt( origin: 0, bound: 24));

        for(int i=0;i<99;i++){
            nnn=cl.add(moad.nextInt( origin: 0, bound: 20), s[name.nextInt( bound: 100)],arrayList.get(resht.nextInt( origin: 0, bound: 7)),vahed.nextInt( origin: 0, bound: 24));
        }
        //cl.display();
        bufferedReader.close();
        cl.display();
    }

    catch (Exception e){

    }
}

```

```

public Node add(double moadel,String name,Reshte reshte,double tedadVahed,Node prev){
    Node newNode = new Node(moadel,name,reshte,tedadVahed);
    if(head.next==null){
        head.prev=null;
        head.next=newNode;
        newNode.prev=head;
        return newNode;
    }
    else{
        prev.next=newNode;
        newNode.prev=prev;
        return newNode;
    }
}

```

در دفعه اول که فقط گره راسی که همان هد است را داریم و بعد از گره راسی نال است پس ند جدید را میسازیم و ان را اضافه میکنیم در دفعات بعدی بعد از گره راسی نال نیست و وارد الس میشود و هر دفعه ما ند جدید را به عنوان ورودی قبلی نود جدیدتر اضافه میکنیم یعنی چی ؟ یعنی مثلا هد اول بعدش نال بوده و به ند اضافه شده ، حالا ند جدید انگار همون نقش هد قبلی مارو بازی میکنه که به عنوان پرو دادمش به تابع که هی داره زیادتر و زیادتر میشه و اینکار هی ادامه پیدا میکنه تا به ۱۰۰ امین ند برسیم

```
public void display() {  
    Node current = head;  
    System.out.println("Nodes of the circular linked list: ");  
    do{  
        System.out.print( current.toString());  
        current = current.next;  
    }while(current != null);  
    return;  
}
```

در آخر هم امدم و از گره راسی همه ۹۹ ند دیگر را پرینت کردم تا زمانی که به انتها برسم و اینکار همان نمایش لیست دو طرفه من هست

نمونه خروجی

```
4.0Node{moade1=17.0, name=
'Agnella', reshte=BARGH, tedadVahed=
21.0Node{moade1=17.0, name=
'Alameda', reshte=ENSANI, tedadVahed=
4.0Node{moade1=6.0, name=
'Adrianna', reshte=SHIMI, tedadVahed=
19.0Node{moade1=1.0, name=
'Aida', reshte=BARGH, tedadVahed=
2.0Node{moade1=6.0, name=
'Adara', reshte=BARGH, tedadVahed=
0.0Node{moade1=15.0, name=
'Ailina', reshte=COMPUTER, tedadVahed=
0.0Node{moade1=8.0, name=
'Adda', reshte=MEMARI, tedadVahed=
14.0Node{moade1=11.0, name=
'Aarika', reshte=SHIMI, tedadVahed=
11.0Node{moade1=10.0, name=
'Adda', reshte=MEMARI, tedadVahed=
1.0Node{moade1=14.0, name=
'Agnese', reshte=HONAR, tedadVahed=
22.0Node{moade1=11.0, name=
'Aga', reshte=MEMARI, tedadVahed=
1.0Node{moade1=7.0, name=
'Ajax', reshte=PHYSISIC, tedadVahed=
22.0
```

