

图的创建练习

```
package 图的创建;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;

public class Graph {

    ArrayList<String> namesArrayList;//储存元素的集合
    private String[] name;//元素名字
    private int n;//元素的个数
    private int m;//边的个数
    private int[][] square;
    public static void main(String[] args) {
        Graph graph=new Graph();
        graph.clear_f();
        graph.initialize_graph();
        graph.insert();
        graph.show_graph();
    }

    public void clear_f(){
        Scanner s=new Scanner(System.in);
        n=s.nextInt();
        name=new String[n];
        for(int i=0;i<n;++i){
            name[i]=s.next();
        }
    }

    public void initialize_graph(){
        square=new int[n][n];

        namesArrayList=new ArrayList<String>(n);

        for(int i=0;i<n;++i){
            namesArrayList.add(name[i]);
        }
    }

    public void insert(){
        System.out.println("How many edges?");
        Scanner s=new Scanner(System.in);
        m=s.nextInt();
        for(int i=0;i<m;++i){
            String k1=s.next();
            String k2=s.next();
        }
    }
}
```

```

int h3=s.nextInt();
insert_square(k1, k2, h3);
    }
    s.close();
}

public void insert_square(String k1,String k2,int h3){
int v1=namesArrayList.indexOf(k1);
int v2=namesArrayList.indexOf(k2);
square[v1][v2]=h3;
square[v2][v1]=h3;
}

public void show_graph(){
    for(int[] i:square){
        System.out.println(Arrays.toString(i));
    }
}

}

```

用java写图还是遇到了好多问题

可以巩固java和数据结构

q1: 为啥clear_f中可以直接用n, 但在main中就不行

在Java中, 静态方法 (如main方法) 和非静态方法 (如clear_f方法) 之间有一些区别, 这也涉及到了对类成员的访问方式。

1. 静态方法 (包括main方法) 是与类本身相关联的, 它们不依赖于任何特定的对象实例。在静态方法中, 无法直接访问非静态成员 (例如实例变量或实例方法), 因为静态方法在对象创建之前就可以被调用, 而实例变量必须在对象创建后才能被初始化。
2. 非静态方法可以直接访问类的实例变量和其他实例方法, 因为它们是在对象的上下文中执行的, 可以使用当前对象的状态和属性。

在您的代码中, clear_f方法是一个非静态方法, 因此它可以直接访问类的实例变量n。但是, main方法是一个静态方法, 不能直接访问实例变量n。要在静态方法中访问实例变量, 您需要首先创建类的一个实例, 然后使用该实例来访问实例变量。

q2: 为什么不能用foreach

foreach看的是是否有下一组是否为null, 数字还是要用普通for

q3:关于next()

nextLine()字符串

next()字符

nextInt()整型

其它的看看代码。