模版·图的实现

```
class Node{
    int val;
    List<Node> neighbors;

    public Node(int val){
        this.val=val;
        this.neighbors=new ArrayList<>();
    }

    public void addNeighbor(Node neighbor){
        neighbors.add(neighbor);
    }
}

class Graph{
    private List<Node> nodes;//邻接表

    public Graph(int numVertices){
        this.nodes=newArrayList<>();
        for(int i=0;i<numVertices;i++){
            nodes.add(new Node(i));
        }
    }

    //添加边
    public void addEdge(int source,int destination){
        Node sourceNode=nodes.get(source);
        Node destinationNode=nodes.get(destination);

        sourceNode.addNeighbor(destinationNode);
        destinationNode.addNeighbor(sourceNode);//无向图
    }

    //删除边
    public void removeEdge(int source,int destination){
        Node sourceNode=nodes.get(source);
        Node destinationNode=nodes.get(destination);

        sourceNode.neighbors.remove(destinationNode);
        destinationNode.neighbors.remove(sourceNode);//无向图
    }

    //判断是否有边
    public boolean hasEdge(int source,int destination){
        Node sourceNode=nodes.get(source);
        Node destinationNode=nodes.get(destination);

        return sourceNode.neighbors.contains(destinationNode);
    }

    //打印邻接表
    public void printAdjacencyList(){
        for(Node node:nodes){
            System.out.print(node.val+":");
            for(Node neighbor:node.neighbors){
                System.out.print(neighbor.val+" ");
            }
            System.out.println();
        }
    }
}
```