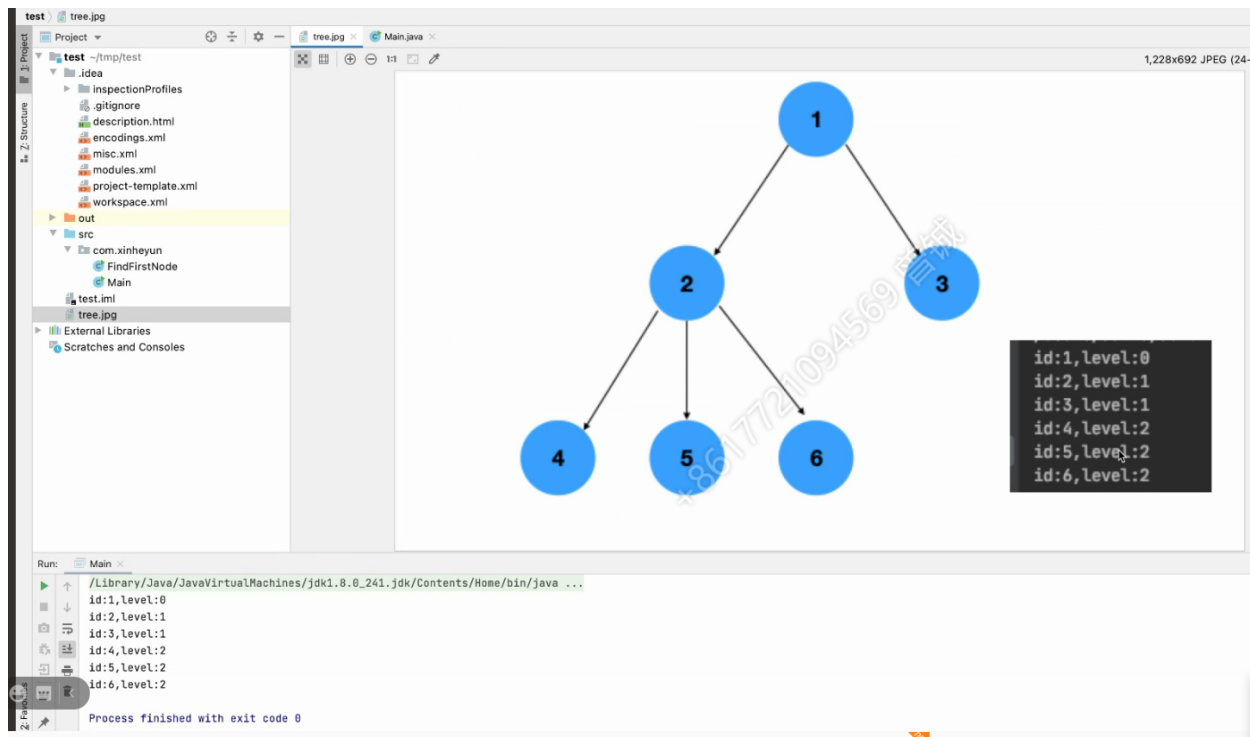


定义:

应用:

问题+代码:



```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

```
# 多叉树的层序遍历问题:
```

```
# 定义多叉树
```

```
class Treenode():
```

```
    def __init__(self, val):
```

```
        self.val = val
```

```
        self.childs = [] #多个孩子节点
```

```
# 建立多叉树之间的关系
```

```
# 首先实例化各个对象
```

```
id1=Treenode(1)
```

```
id2=Treenode(2)
```

```
id3=Treenode(3)
```

```
id4=Treenode(4)
```

```
id5=Treenode(5)
```

```
id6=Treenode(6)
```

```
# 然后, 建立各个对象之间地关系
```

```
id1.childs.append(id2)
```

```
id1.childs.append(id3)
```

```
id2.childs.append(id4)
```

```
id2.childs.append(id5)
```

```
id2.childs.append(id6)
```

```
# 使用广度优先搜索的方法；记得使用双端队列
# 根据题目要求，输出打印每一层的节点
from collections import deque
# 定义一个双端队列，用于输入输出各父节点、子节点的值
queue=deque()
queue.append(id1)
def bfs(root):
    level = 0 #放在函数里面才有值
    while len(queue) > 0:
        # 开始遍历队列中的元素：
        for i in range(len(queue)):
            newRoot=queue.pop()
            print("id:{},level:{}".format(newRoot.val, level))
            for child in newRoot.childs:
                queue.appendleft(child)
        level += 1
bfs(id1)
```