

There are the following JAVA classes implementing the multi-frontal direct solver algorithm for finite difference problem:

- *A.java*
- *A1.java*
- *A2.java*
- *AN.java*
- *Aroot.java*
- *BS.java*
- *E2.java*
- *Eroot.java*
- *Executor.java*
- *Main.java*
- *P1.java*
- *P2.java*
- *P3.java*
- *Production.java*
- *Vertex.java*

Let us focus on the classes *A2.java*, *E2.java*, *Aroot.java*, *Eroot.java*, *BS.java*. The class *Aroot* performs merging at root node.

```
1 class Aroot extends A2 {
2     Aroot(Vertex Vert, CyclicBarrier Barrier) {
3         super(Vert, Barrier);
4     }
5 }
```

The class *Eroot* solves the problem at the root node. This is done by the standard Gaussian elimination for 3 times 3 matrix

```

1  divider = A(0,0)
2  for icol=0,2
3      A(0,icol)=A(0,icol)/divider
4  end loop
5  b(0)=b(0)/divider
6  for irow=1,2
7      multiplier = A(irow,0)
8      A(irow,0)=0
9      for icol=1,2
10         A(irow,icol) = A(irow,icol) - A(irow,0) * A(0,icol)
11     end loop
12     b(irow)=b(irow)-A(irow,0)*b(0)
13 end loop
14 divider = A(1,1)
15 for icol=1,2
16     A(1,icol)=A(1,icol)/divider
17 end loop
18 b(1)=b(1)/divider
19 for irow=2,2
20     multiplier = A(irow,1)
21     A(irow,1)=1
22     for icol=1,2
23         A(irow,icol) = A(irow,icol) - A(irow,1) * A(1,icol)
24     end loop
25     b(irow)=b(irow)-A(irow,1)*b(1)
26 end loop
27 divider = A(2,2)
28 A(2,2)=1
29 b(2) = b(2)/divider
30 //backward substitutions
31 x(2)=b(2)/A(2,2)
32 x(1)=(b(1)-A(1,2)*x(2))/a(1,1)
33 x(0)=(b(0)-A(0,1)*x(1)-A(0,2)*x(2))/a(0,0)

```

Please implement the production *Eroot*