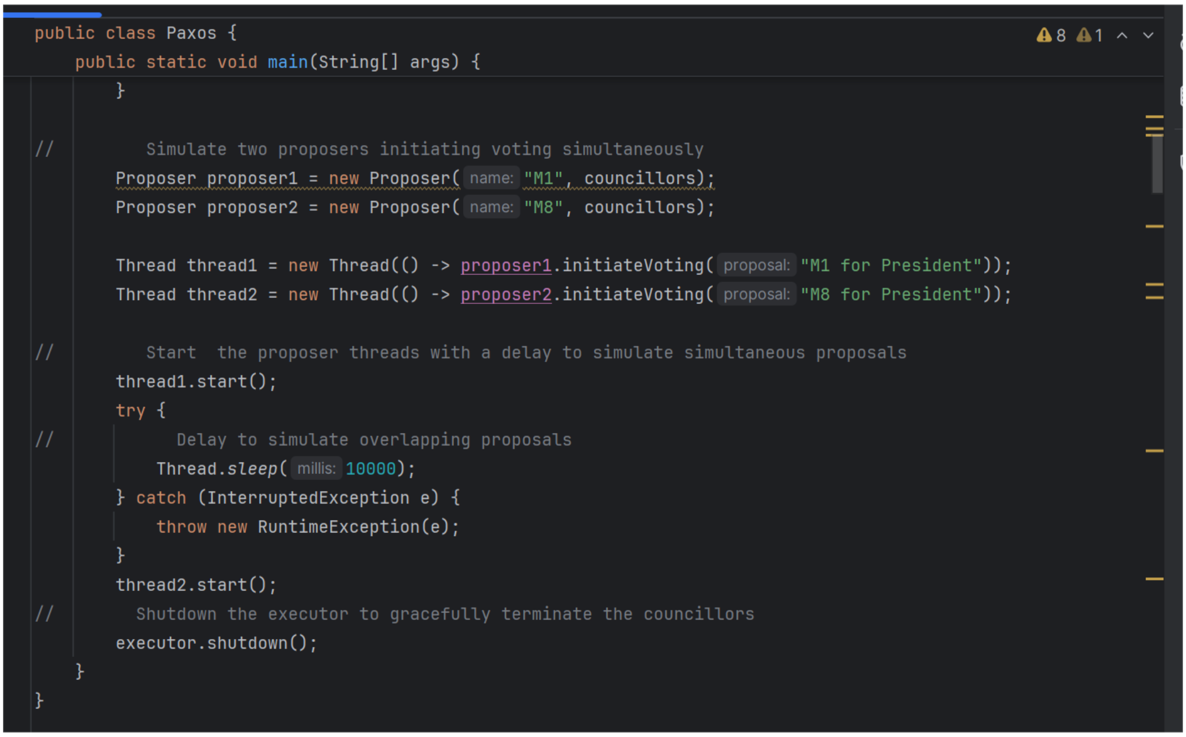
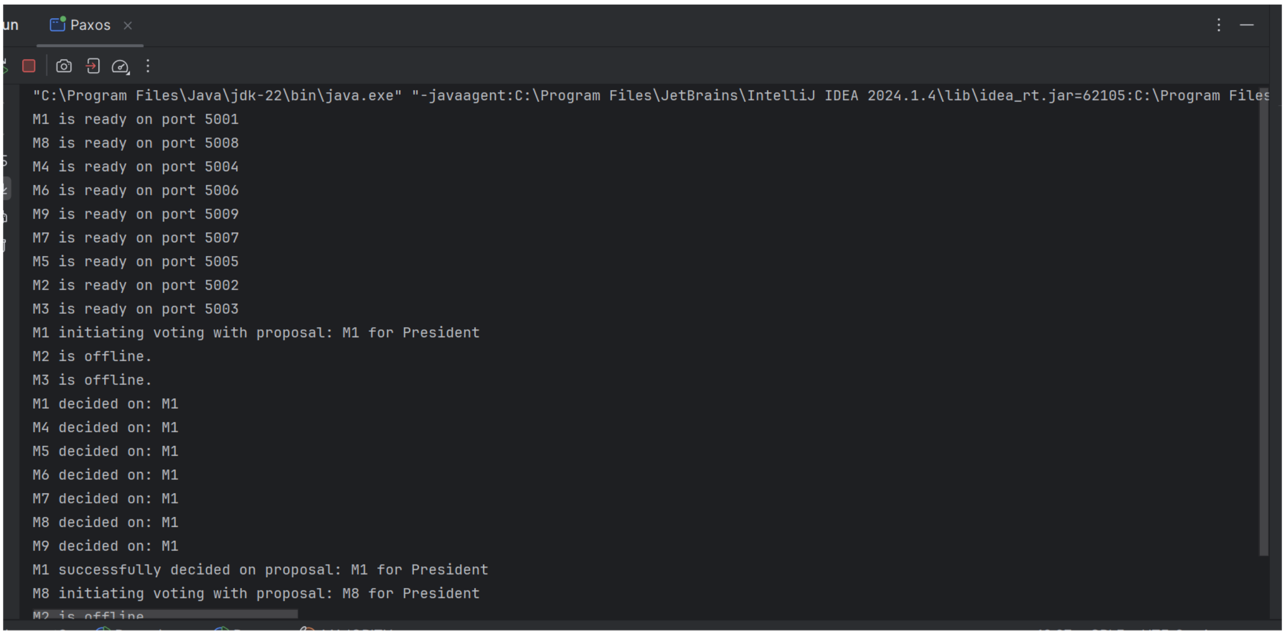
ASSIGNMENT-3 OUTPUT SCREENSHOTS

1.Paxos implementation works when two councillors send voting proposals at the same time.

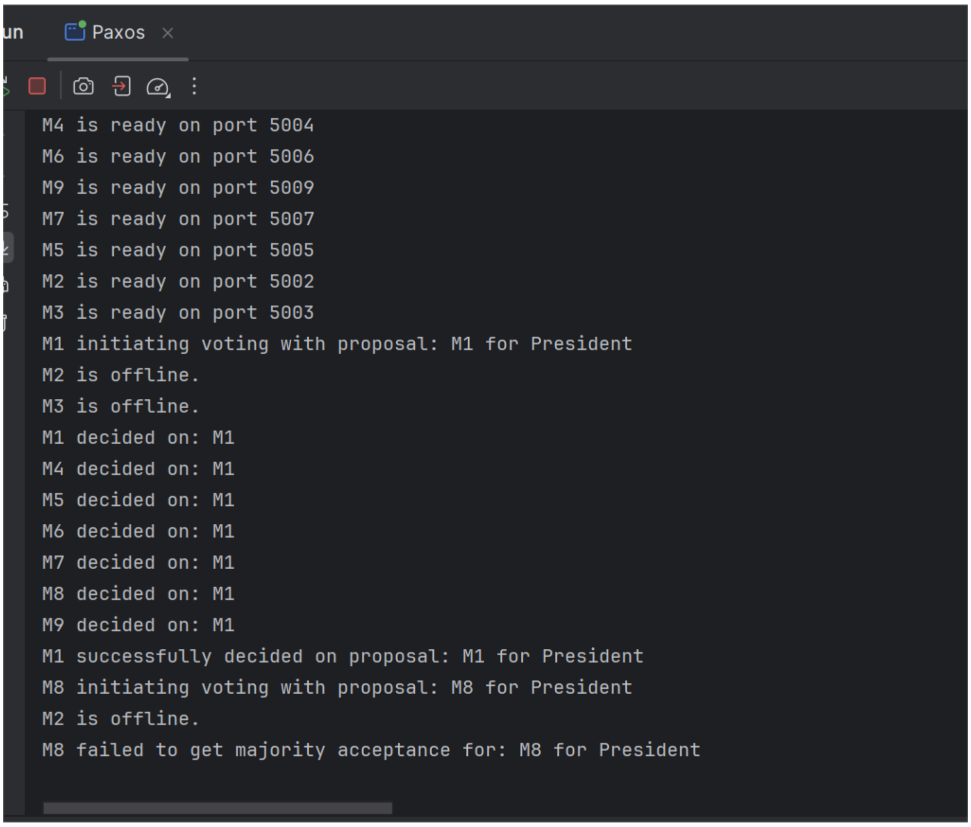


```
public class Paxos {
    public static void main(String[] args) {
        }

//         Simulate two proposers initiating voting simultaneously
        Proposer proposer1 = new Proposer( name: "M1", councillors);
        Proposer proposer2 = new Proposer( name: "M8", councillors);

        Thread thread1 = new Thread(() -> proposer1.initiateVoting( proposal: "M1 for President"));
        Thread thread2 = new Thread(() -> proposer2.initiateVoting( proposal: "M8 for President"));

//         Start  the proposer threads with a delay to simulate simultaneous proposals
        thread1.start();
        try {
//             Delay to simulate overlapping proposals
            Thread.sleep( millis: 10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        thread2.start();
//        Shutdown the executor to gracefully terminate the councillors
        executor.shutdown();
    }
}
```

OUTPUT:

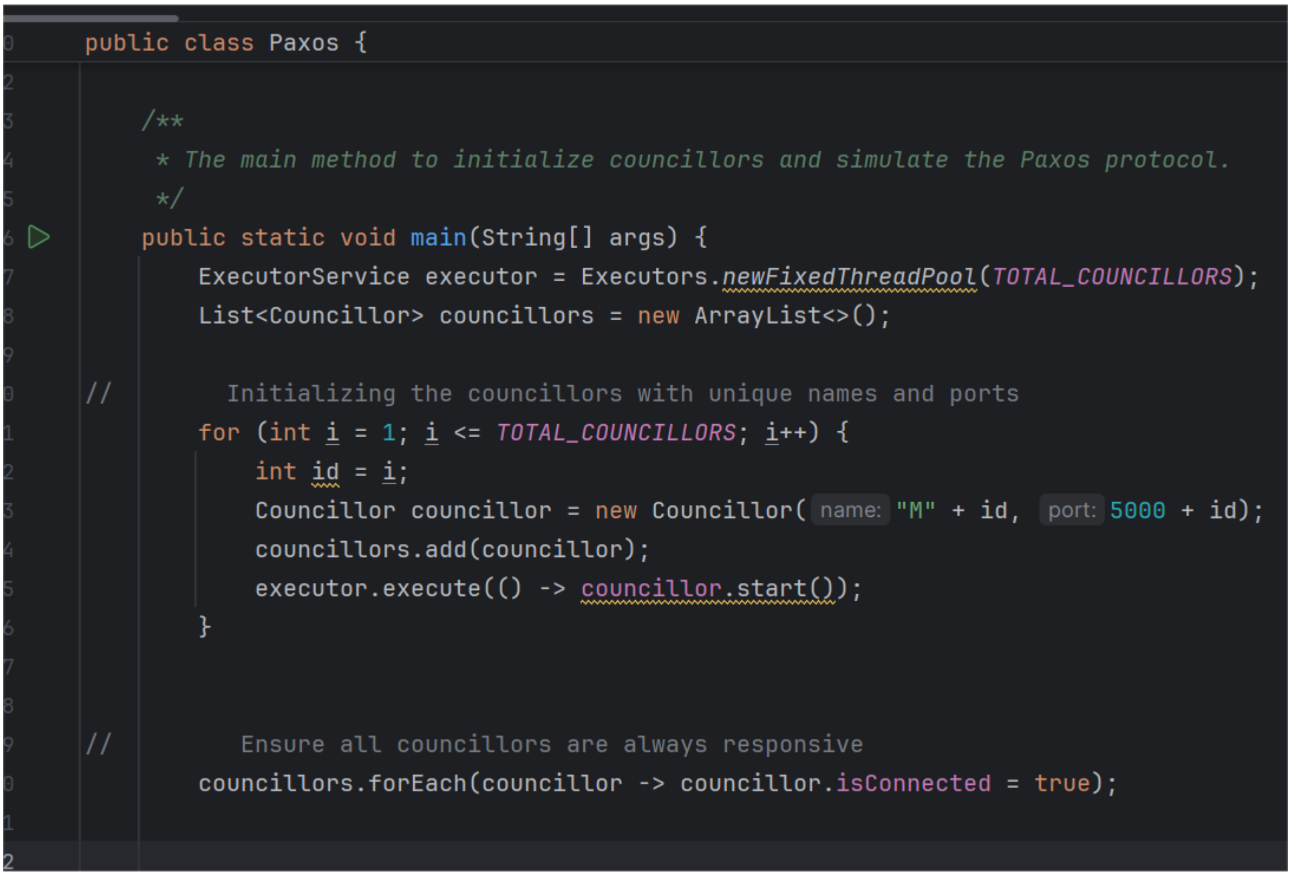


```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1.4\lib\idea_rt.jar=62105:C:\Program Files
M1 is ready on port 5001
M8 is ready on port 5008
M4 is ready on port 5004
M6 is ready on port 5006
M9 is ready on port 5009
M7 is ready on port 5007
M5 is ready on port 5005
M2 is ready on port 5002
M3 is ready on port 5003
M1 initiating voting with proposal: M1 for President
M2 is offline.
M3 is offline.
M1 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1 for President
M8 initiating voting with proposal: M8 for President
M2 is offline
```



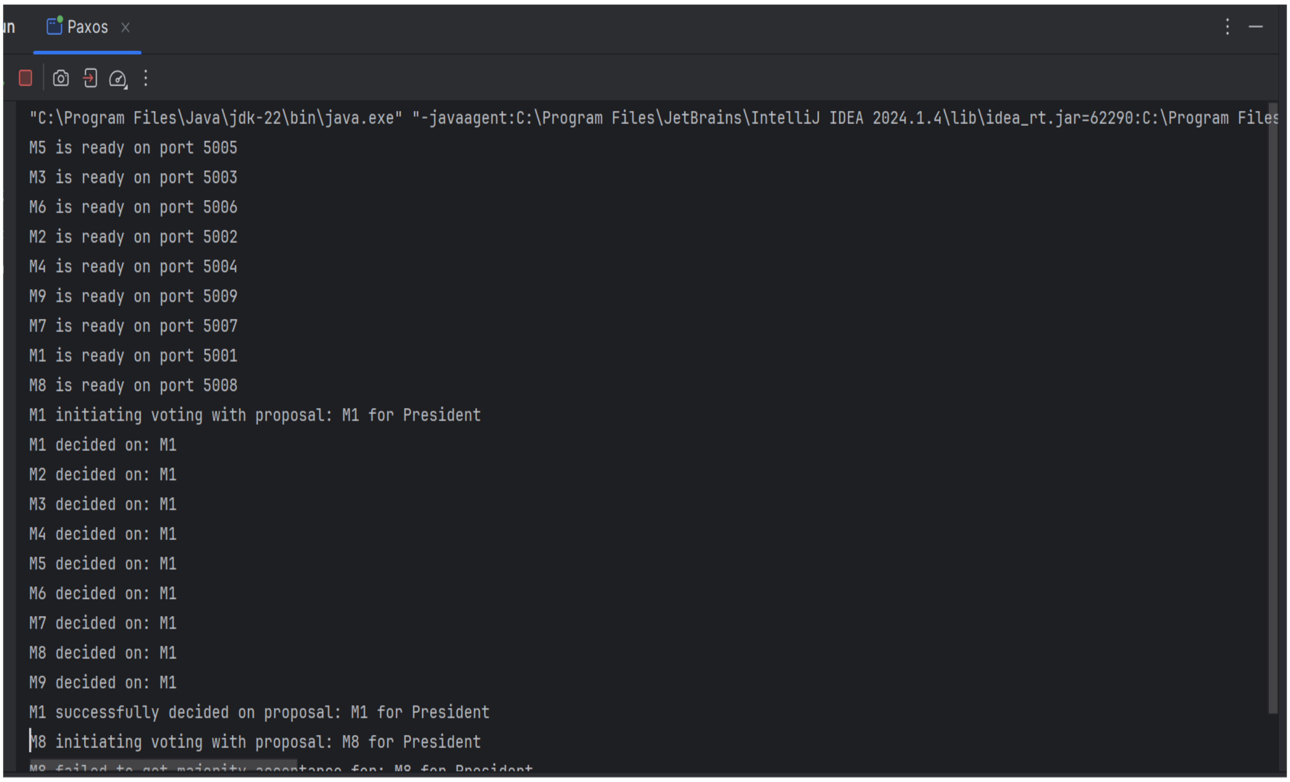
```
M4 is ready on port 5004
M6 is ready on port 5006
M9 is ready on port 5009
M7 is ready on port 5007
M5 is ready on port 5005
M2 is ready on port 5002
M3 is ready on port 5003
M1 initiating voting with proposal: M1 for President
M2 is offline.
M3 is offline.
M1 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1 for President
M8 initiating voting with proposal: M8 for President
M2 is offline.
M8 failed to get majority acceptance for: M8 for President
```

2. Paxos implementation works in the case where all M1-M9 have immediate responses to voting queries.

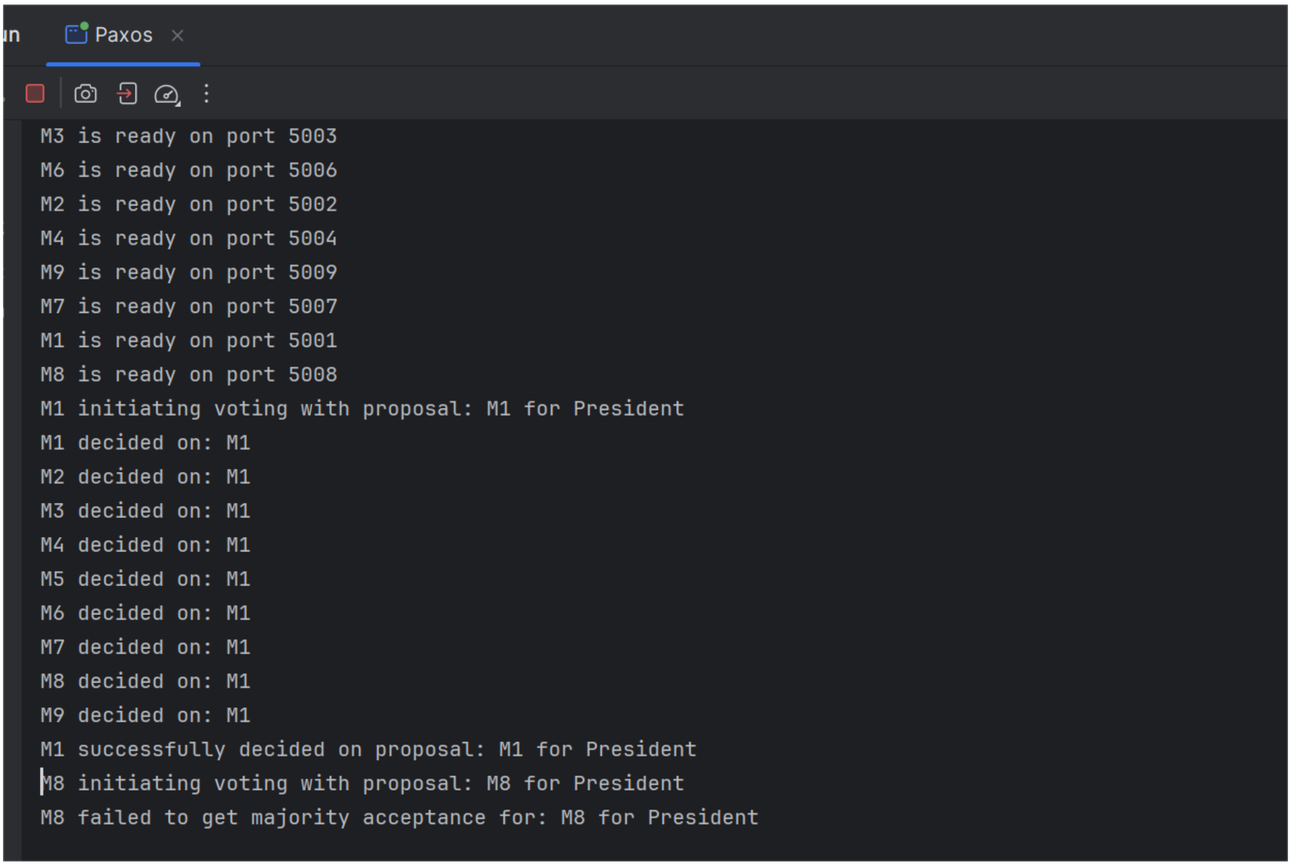


```
public class Paxos {

    /**
     * The main method to initialize councillors and simulate the Paxos protocol.
     */
    public static void main(String[] args) {
        ExecutorService executor = Executors.newFixedThreadPool(TOTAL_COUNCILLORS);
        List<Councillor> councillors = new ArrayList<>();

//          Initializing the councillors with unique names and ports
        for (int i = 1; i <= TOTAL_COUNCILLORS; i++) {
            int id = i;
            Councillor councillor = new Councillor( name: "M" + id,  port: 5000 + id);
            councillors.add(councillor);
            executor.execute(() -> councillor.start());
        }


//           Ensure all councillors are always responsive
        councillors.forEach(councillor -> councillor.isConnected = true);

```

OUTPUT:

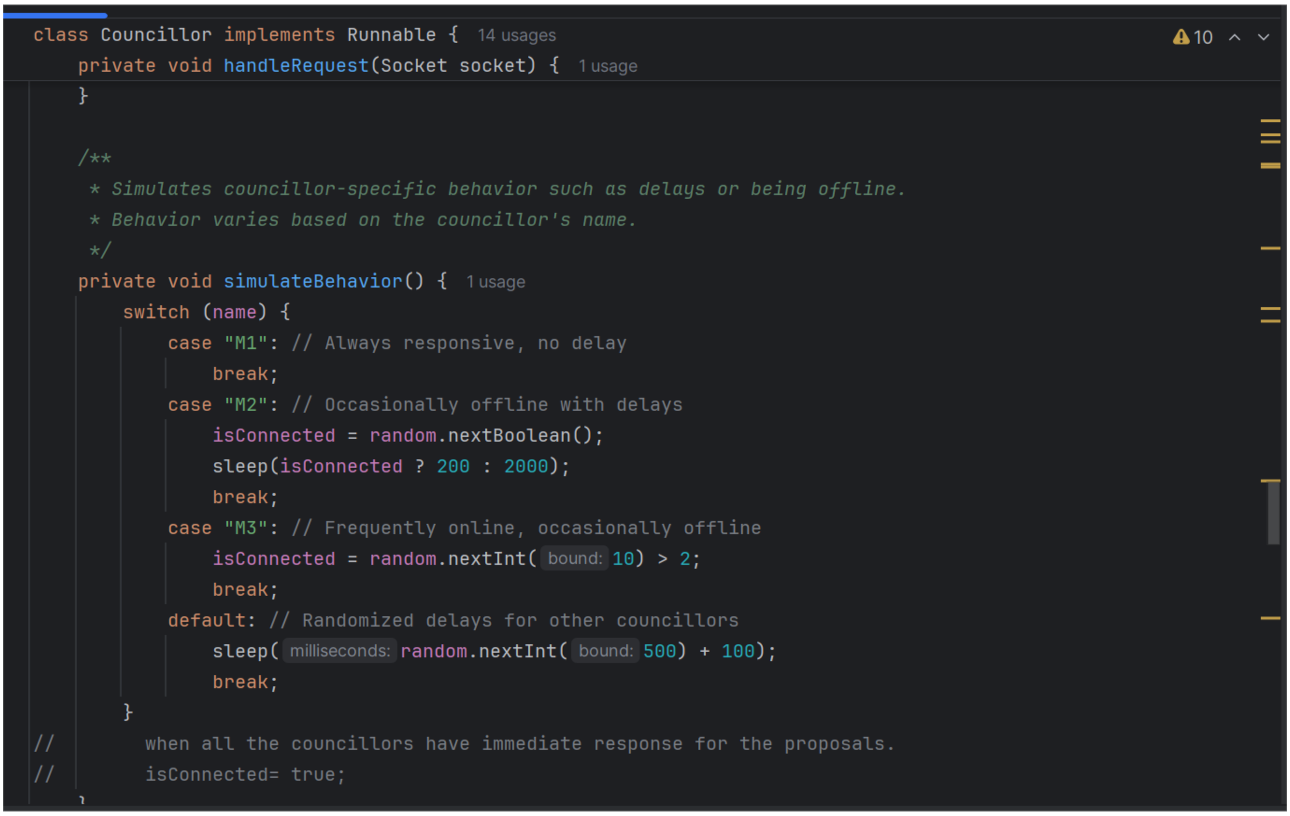


```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1.4\lib\idea_rt.jar=62290:C:\Program Files
M5 is ready on port 5005
M3 is ready on port 5003
M6 is ready on port 5006
M2 is ready on port 5002
M4 is ready on port 5004
M9 is ready on port 5009
M7 is ready on port 5007
M1 is ready on port 5001
M8 is ready on port 5008
M1 initiating voting with proposal: M1 for President
M1 decided on: M1
M2 decided on: M1
M3 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1 for President
M8 initiating voting with proposal: M8 for President
M8 failed to get majority acceptance for: M8 for President
```

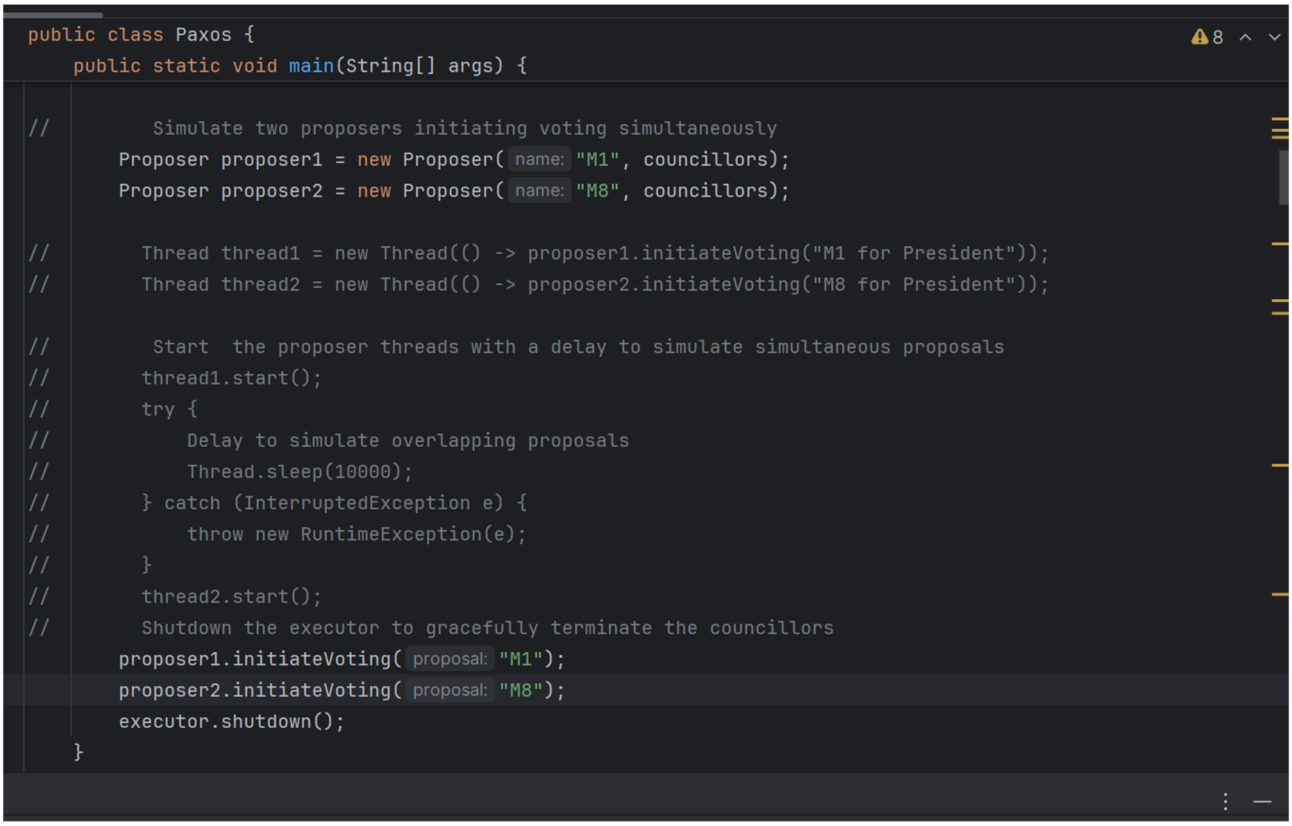


```
M3 is ready on port 5003
M6 is ready on port 5006
M2 is ready on port 5002
M4 is ready on port 5004
M9 is ready on port 5009
M7 is ready on port 5007
M1 is ready on port 5001
M8 is ready on port 5008
M1 initiating voting with proposal: M1 for President
M1 decided on: M1
M2 decided on: M1
M3 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1 for President
M8 initiating voting with proposal: M8 for President
M8 failed to get majority acceptance for: M8 for President
```

3.Paxos implementation works when M1 – M9 have responses to voting queries suggested by several profiles (immediate response, small delay, large delay and no response), including when M2 or M3 propose and then go offline.

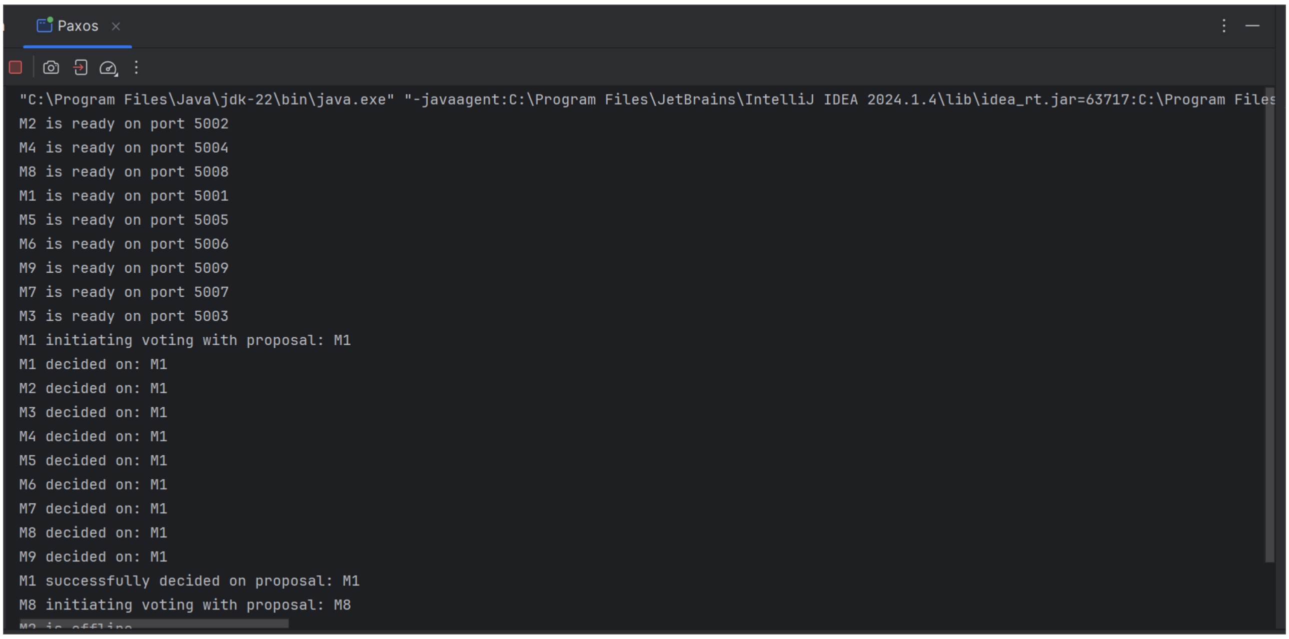


```
class Councillor implements Runnable {  14 usages
    private void handleRequest(Socket socket) {  1 usage
    }

    /**
     * Simulates councillor-specific behavior such as delays or being offline.
     * Behavior varies based on the councillor's name.
     */
    private void simulateBehavior() {  1 usage
        switch (name) {
            case "M1": // Always responsive, no delay
                break;
            case "M2": // Occasionally offline with delays
                isConnected = random.nextBoolean();
                sleep(isConnected ? 200 : 2000);
                break;
            case "M3": // Frequently online, occasionally offline
                isConnected = random.nextInt( bound: 10) > 2;
                break;
            default: // Randomized delays for other councillors
                sleep( milliseconds: random.nextInt( bound: 500) + 100);
                break;
        }
//          when all the councillors have immediate response for the proposals.
//          isConnected= true;
    }
```

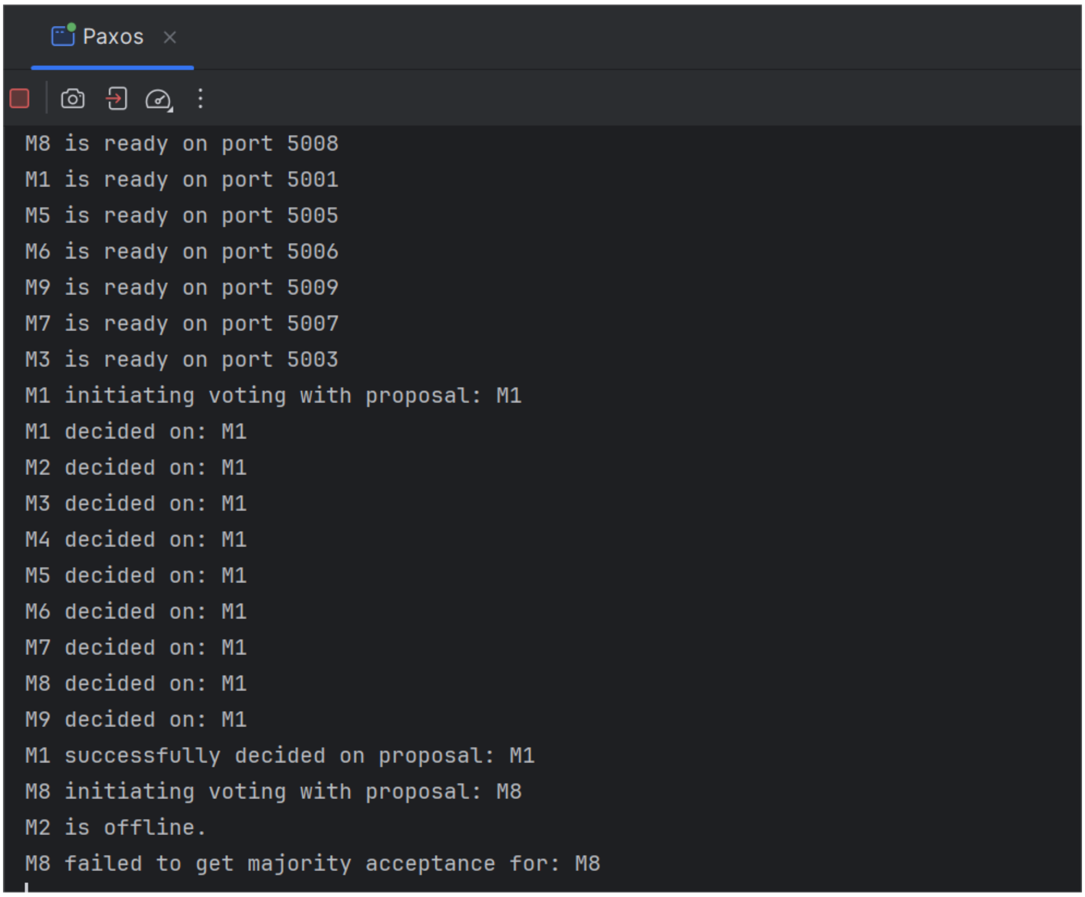


```
public class Paxos {
    public static void main(String[] args) {

//           Simulate two proposers initiating voting simultaneously
        Proposer proposer1 = new Proposer( name: "M1", councillors);
        Proposer proposer2 = new Proposer( name: "M8", councillors);

//          Thread thread1 = new Thread(() -> proposer1.initiateVoting("M1 for President"));
//          Thread thread2 = new Thread(() -> proposer2.initiateVoting("M8 for President"));

//           Start  the proposer threads with a delay to simulate simultaneous proposals
//          thread1.start();
//          try {
//              Delay to simulate overlapping proposals
//              Thread.sleep(10000);
//          } catch (InterruptedException e) {
//              throw new RuntimeException(e);
//          }
//          thread2.start();
//          Shutdown the executor to gracefully terminate the councillors
        proposer1.initiateVoting( proposal: "M1");
        proposer2.initiateVoting( proposal: "M8");
        executor.shutdown();
    }
```

OUTPUT:

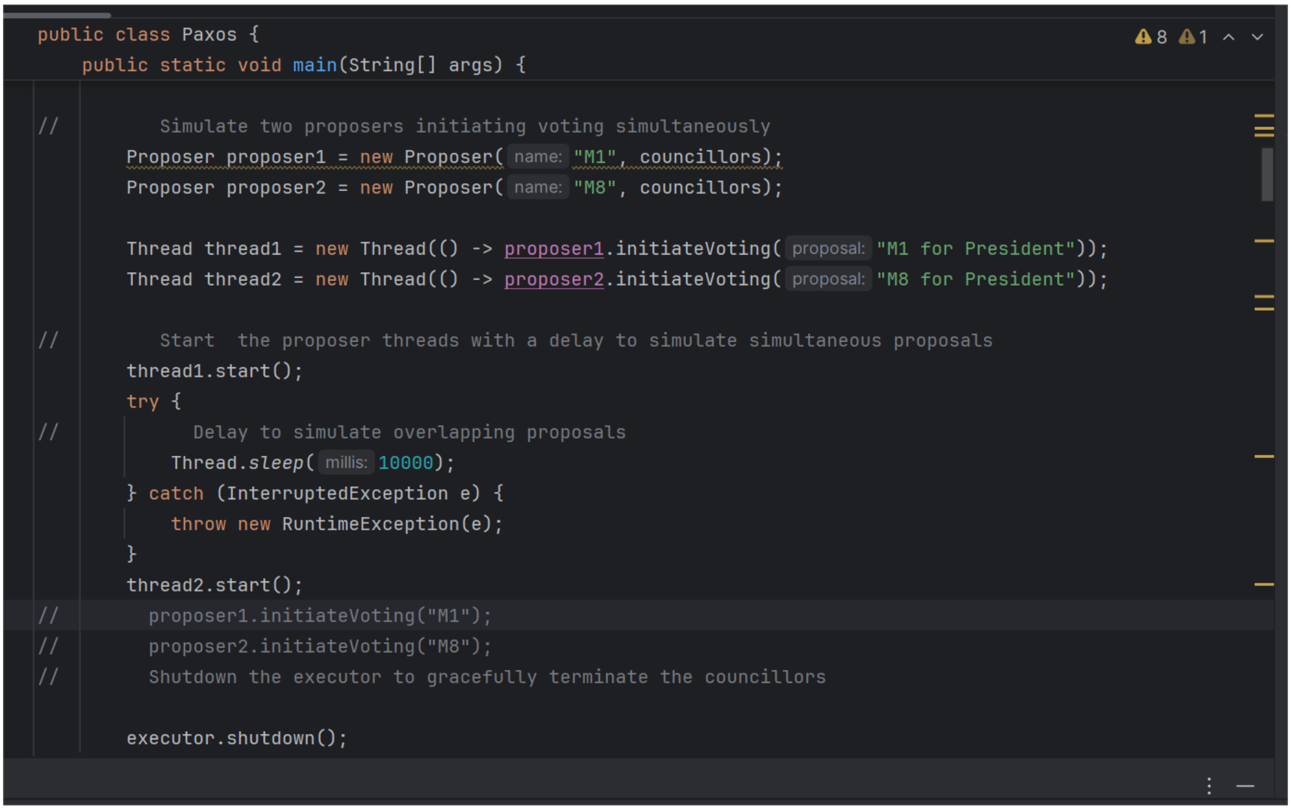


```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1.4\lib\idea_rt.jar=63717:C:\Program Files
M2 is ready on port 5002
M4 is ready on port 5004
M8 is ready on port 5008
M1 is ready on port 5001
M5 is ready on port 5005
M6 is ready on port 5006
M9 is ready on port 5009
M7 is ready on port 5007
M3 is ready on port 5003
M1 initiating voting with proposal: M1
M1 decided on: M1
M2 decided on: M1
M3 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1
M8 initiating voting with proposal: M8
M2 is offline
```



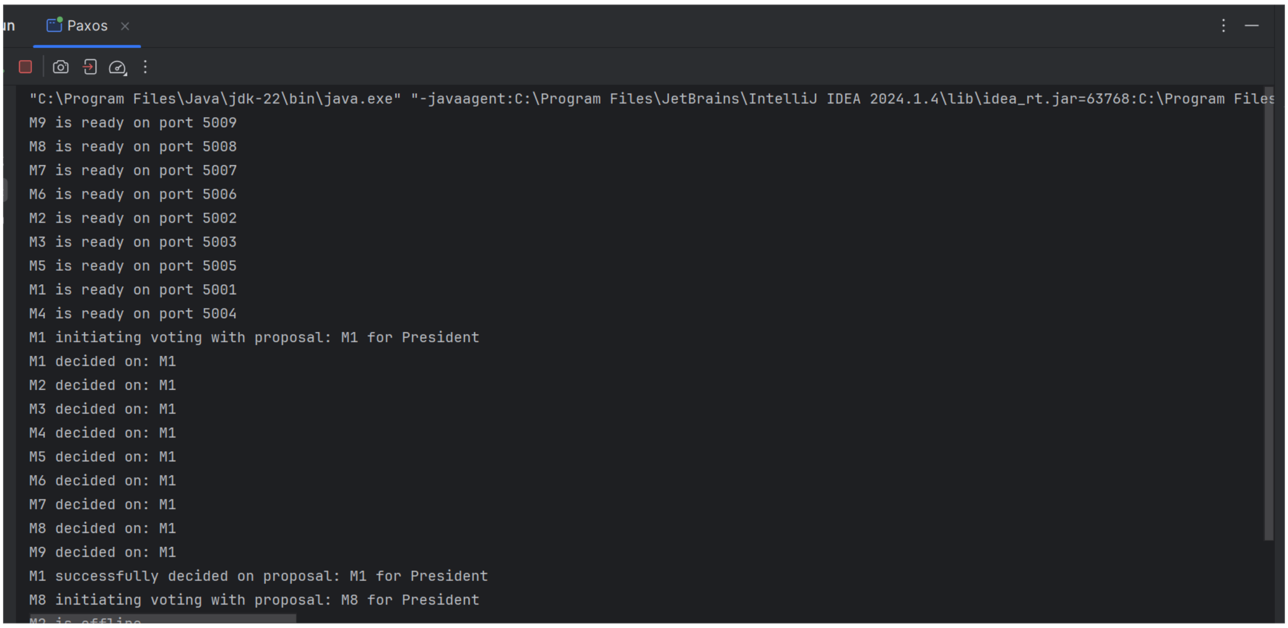
```
M8 is ready on port 5008
M1 is ready on port 5001
M5 is ready on port 5005
M6 is ready on port 5006
M9 is ready on port 5009
M7 is ready on port 5007
M3 is ready on port 5003
M1 initiating voting with proposal: M1
M1 decided on: M1
M2 decided on: M1
M3 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1
M8 initiating voting with proposal: M8
M2 is offline.
M8 failed to get majority acceptance for: M8
```

3.1

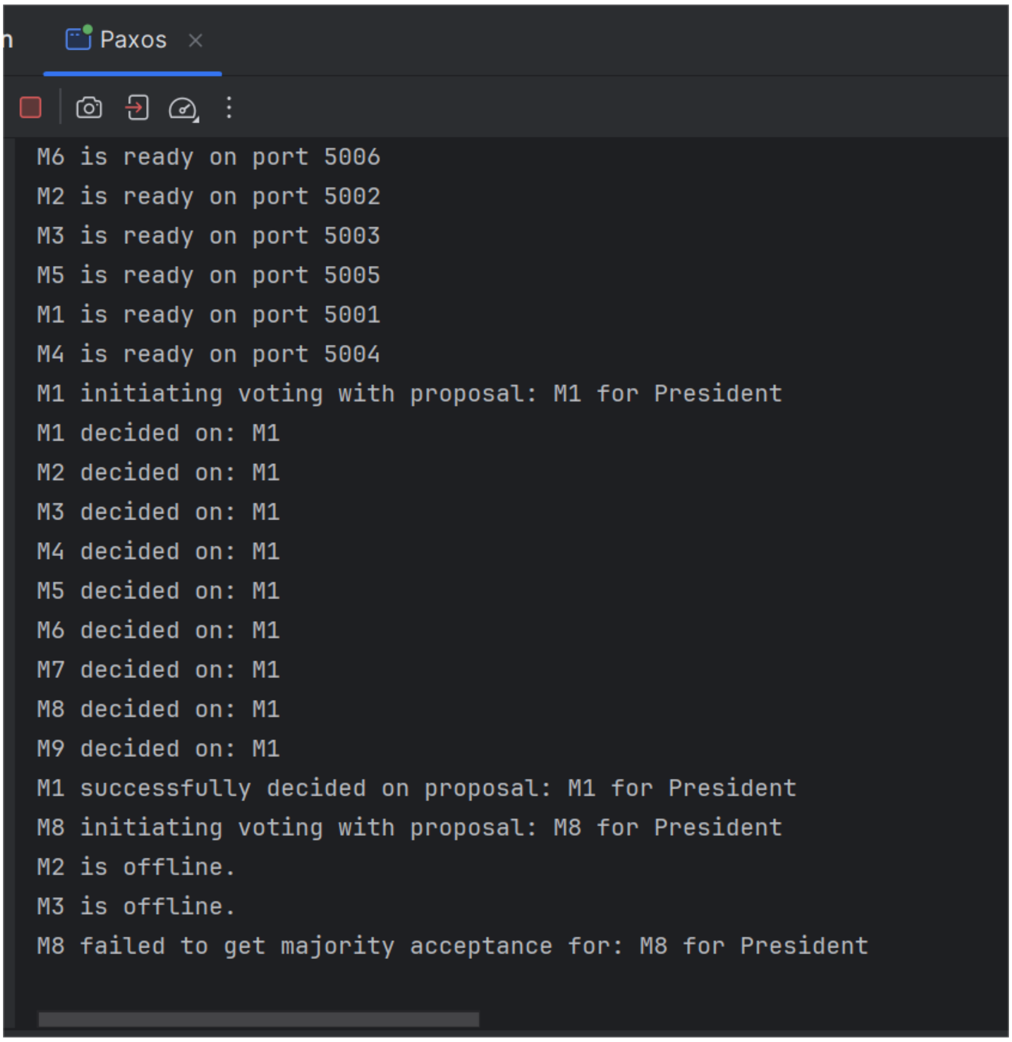


```
public class Paxos {
    public static void main(String[] args) {

//        Simulate two proposers initiating voting simultaneously
        Proposer proposer1 = new Proposer("M1", councillors);
        Proposer proposer2 = new Proposer("M8", councillors);

        Thread thread1 = new Thread(() -> proposer1.initiateVoting("M1 for President"));
        Thread thread2 = new Thread(() -> proposer2.initiateVoting("M8 for President"));

//        Start  the proposer threads with a delay to simulate simultaneous proposals
        thread1.start();
        try {
//            Delay to simulate overlapping proposals
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        thread2.start();
//      proposer1.initiateVoting("M1");
//      proposer2.initiateVoting("M8");
//      Shutdown the executor to gracefully terminate the councillors

        executor.shutdown();
```

OUTPUT:

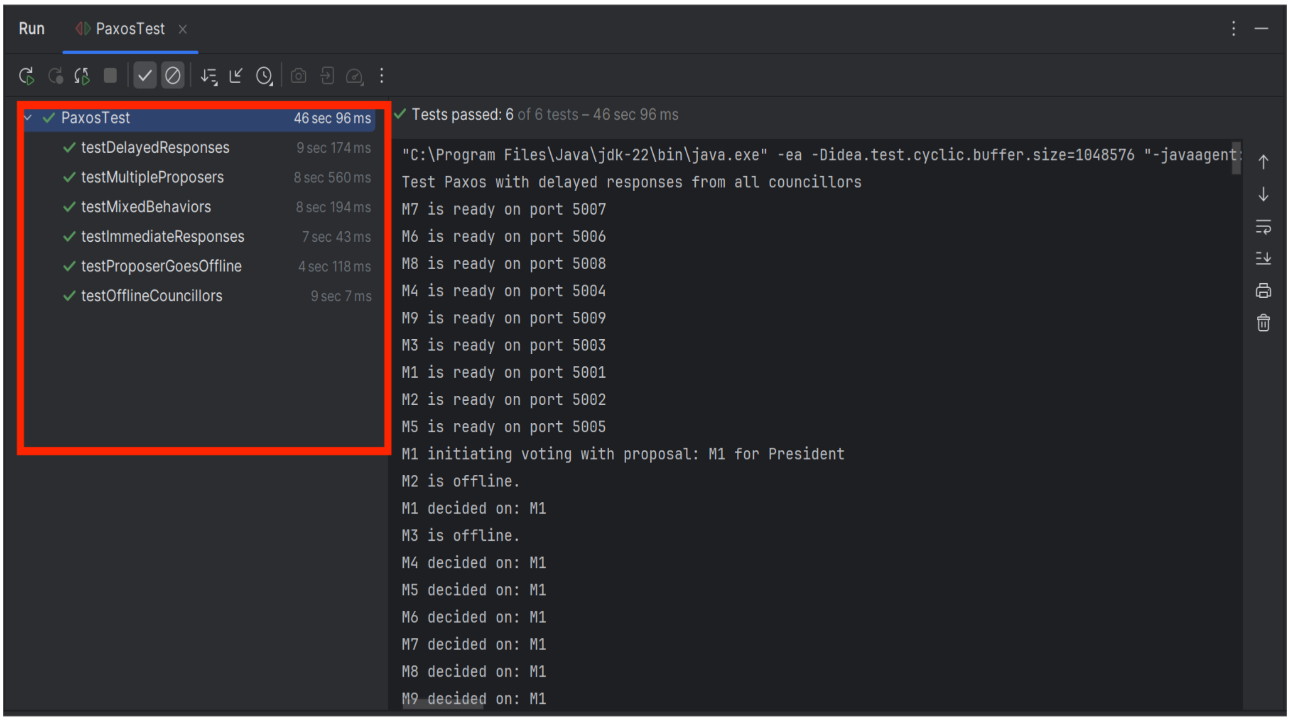


```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1.4\lib\idea_rt.jar=63768:C:\Program Files
M9 is ready on port 5009
M8 is ready on port 5008
M7 is ready on port 5007
M6 is ready on port 5006
M2 is ready on port 5002
M3 is ready on port 5003
M5 is ready on port 5005
M1 is ready on port 5001
M4 is ready on port 5004
M1 initiating voting with proposal: M1 for President
M1 decided on: M1
M2 decided on: M1
M3 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1 for President
M8 initiating voting with proposal: M8 for President
M2 is offline
```



```
M6 is ready on port 5006
M2 is ready on port 5002
M3 is ready on port 5003
M5 is ready on port 5005
M1 is ready on port 5001
M4 is ready on port 5004
M1 initiating voting with proposal: M1 for President
M1 decided on: M1
M2 decided on: M1
M3 decided on: M1
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
M1 successfully decided on proposal: M1 for President
M8 initiating voting with proposal: M8 for President
M2 is offline.
M3 is offline.
M8 failed to get majority acceptance for: M8 for President
```

TESTING:

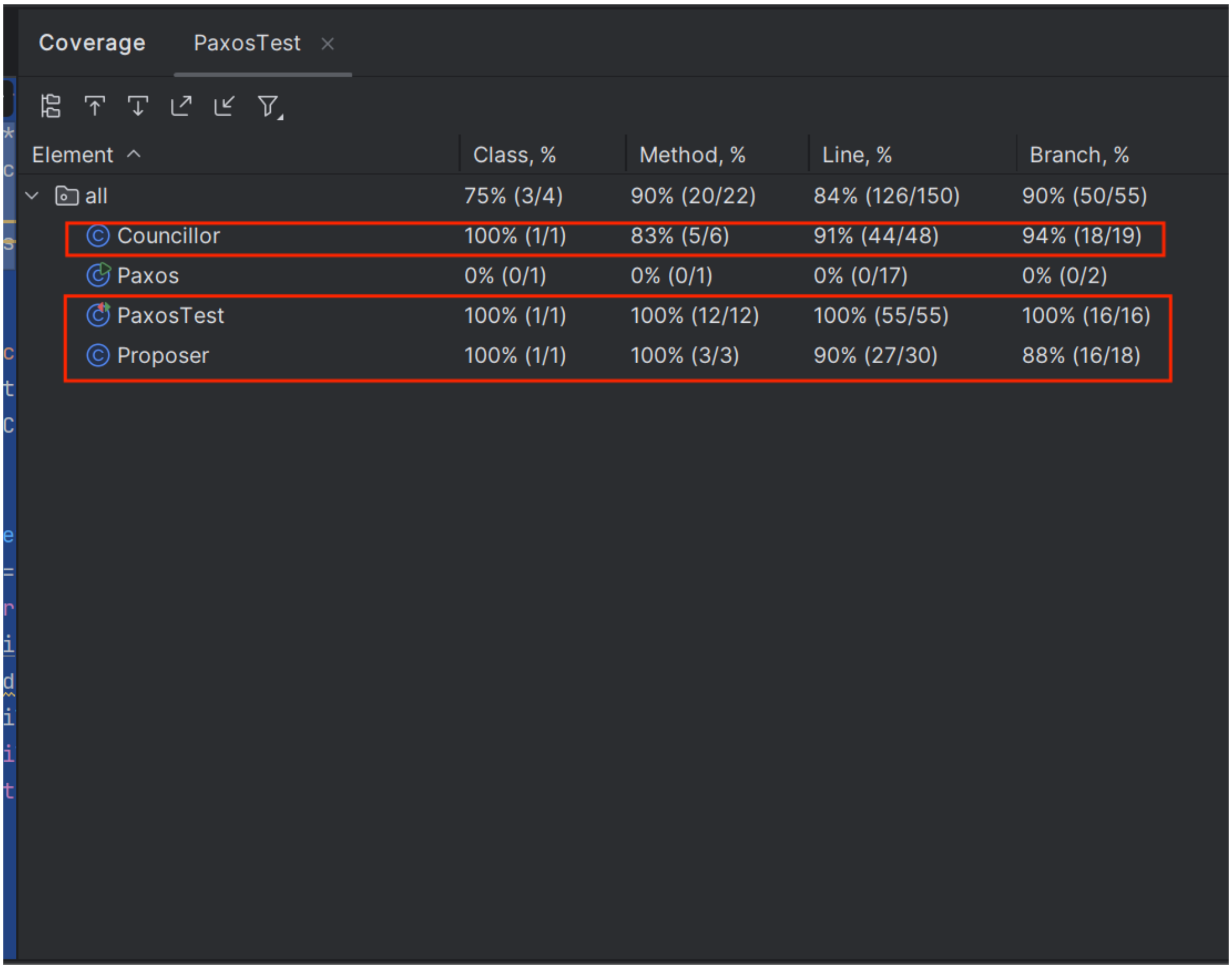


```
✓ PaxosTest                       46 sec 96 ms
    ✓ testDelayedResponses         9 sec 174 ms
    ✓ testMultipleProposers        8 sec 560 ms
    ✓ testMixedBehaviors           8 sec 194 ms
    ✓ testImmediateResponses       7 sec 43 ms
    ✓ testProposerGoesOffline      4 sec 118 ms
    ✓ testOfflineCouncillors       9 sec 7 ms
```

✓ Tests passed: 6 of 6 tests – 46 sec 96 ms

```
"C:\Program Files\Java\jdk-22\bin\java.exe" -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:
Test Paxos with delayed responses from all councillors
M7 is ready on port 5007
M6 is ready on port 5006
M8 is ready on port 5008
M4 is ready on port 5004
M9 is ready on port 5009
M3 is ready on port 5003
M1 is ready on port 5001
M2 is ready on port 5002
M5 is ready on port 5005
M1 initiating voting with proposal: M1 for President
M2 is offline.
M1 decided on: M1
M3 is offline.
M4 decided on: M1
M5 decided on: M1
M6 decided on: M1
M7 decided on: M1
M8 decided on: M1
M9 decided on: M1
```

TEST COVERAGE:



Element ^	Class, %	Method, %	Line, %	Branch, %
all	75% (3/4)	90% (20/22)	84% (126/150)	90% (50/55)
Councillor	100% (1/1)	83% (5/6)	91% (44/48)	94% (18/19)
Paxos	0% (0/1)	0% (0/1)	0% (0/17)	0% (0/2)
PaxosTest	100% (1/1)	100% (12/12)	100% (55/55)	100% (16/16)
Proposer	100% (1/1)	100% (3/3)	90% (27/30)	88% (16/18)