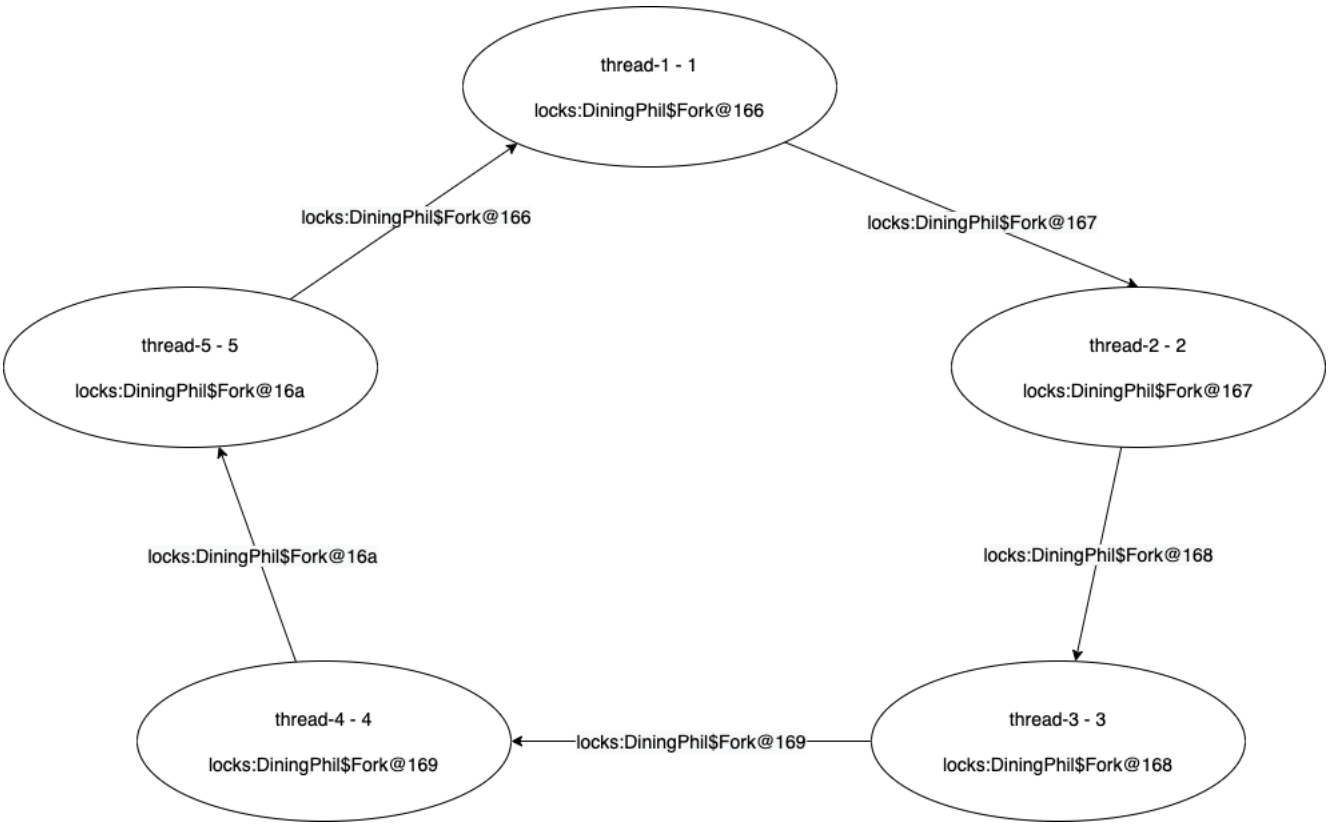
Error Trace of Dining Philosophers

transition \ Thread name - id	main - 0	thread-1 - 1	thread-2 - 2	thread-3 - 3	thread-4 - 4	thread-5 - 5
0	create forks and creates first Philosopher					
1-12	create remaining Philosophers					
13		obtain lock locks:DiningPhil$Fork@166				
14		try to obtain lock DiningPhil$Fork@167				
15			obtain lock locks:DiningPhil$Fork@167			
16			try to obtain lock DiningPhil$Fork@168			
17				obtain lock locks:DiningPhil$Fork@168		
18				try to obtain lock DiningPhil$Fork@169		
19					obtain lock locks:DiningPhil$Fork@169	
20					try to obtain lock DiningPhil$Fork@16a	
21						obtain lock locks:DiningPh
22						try to obtain lo DiningPhil$For

Lock dependency graph



Full error trace

```
====================================================== system under test
DiningPhil.main()

====================================================== search started: 5/5/21 8:56 AM

====================================================== error 1
gov.nasa.jpf.vm.NotDeadlockedProperty
deadlock encountered:
  thread DiningPhil$Philosopher:{id:1,name:Thread-
1,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  thread DiningPhil$Philosopher:{id:2,name:Thread-
2,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  thread DiningPhil$Philosopher:{id:3,name:Thread-
3,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  thread DiningPhil$Philosopher:{id:4,name:Thread-
4,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  thread DiningPhil$Philosopher:{id:5,name:Thread-
5,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}


====================================================== trace #1
------------------------------------------------------ transition #0 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"ROOT" ,1/1,isCascaded:false}
      [3168 insn w/o sources]
  DiningPhil.java:50             : Fork[] forks = new Fork[N];
  DiningPhil.java:51             : for (int i = 0; i < N; i++) {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:23             : static class Fork {
      [1 insn w/o sources]
  DiningPhil.java:23             : static class Fork {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:51             : for (int i = 0; i < N; i++) {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:23             : static class Fork {
      [1 insn w/o sources]
  DiningPhil.java:23             : static class Fork {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:51             : for (int i = 0; i < N; i++) {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:23             : static class Fork {
      [1 insn w/o sources]
  DiningPhil.java:23             : static class Fork {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:51             : for (int i = 0; i < N; i++) {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:23             : static class Fork {
      [1 insn w/o sources]
  DiningPhil.java:23             : static class Fork {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:51             : for (int i = 0; i < N; i++) {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:23             : static class Fork {
      [1 insn w/o sources]
  DiningPhil.java:23             : static class Fork {
  DiningPhil.java:52             : forks[i] = new Fork();
  DiningPhil.java:51             : for (int i = 0; i < N; i++) {
  DiningPhil.java:55             : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:31             : public Philosopher(Fork left, Fork right) {
      [145 insn w/o sources]
  DiningPhil.java:32             : this.left = left;
  DiningPhil.java:33             : this.right = right;
  DiningPhil.java:34             : start();
      [1 insn w/o sources]
------------------------------------------------------ transition #1 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"START" ,1/2,isCascaded:false}
      [2 insn w/o sources]
  DiningPhil.java:35             : }
  DiningPhil.java:55             : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:54             : for (int i = 0; i < N; i++) {
  DiningPhil.java:55             : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:31             : public Philosopher(Fork left, Fork right) {
      [27 insn w/o sources]
------------------------------------------------------ transition #2 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,1/2,isCascaded:false}
      [119 insn w/o sources]
  DiningPhil.java:32             : this.left = left;
  DiningPhil.java:33             : this.right = right;
```

```
------------------------------------------------------ transition #3 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"EXPOSE" ,1/2,isCascaded:false}
  DiningPhil.java:33            : this.right = right;
  DiningPhil.java:34            : start();
      [1 insn w/o sources]
------------------------------------------------------ transition #4 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"START" ,1/3,isCascaded:false}
      [2 insn w/o sources]
  DiningPhil.java:35            : }
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:54            : for (int i = 0; i < N; i++) {
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:31            : public Philosopher(Fork left, Fork right) {
      [27 insn w/o sources]
------------------------------------------------------ transition #5 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,1/3,isCascaded:false}
      [119 insn w/o sources]
  DiningPhil.java:32            : this.left = left;
  DiningPhil.java:33            : this.right = right;
------------------------------------------------------ transition #6 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"EXPOSE" ,1/3,isCascaded:false}
  DiningPhil.java:33            : this.right = right;
  DiningPhil.java:34            : start();
      [1 insn w/o sources]
------------------------------------------------------ transition #7 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"START" ,1/4,isCascaded:false}
      [2 insn w/o sources]
  DiningPhil.java:35            : }
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:54            : for (int i = 0; i < N; i++) {
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:31            : public Philosopher(Fork left, Fork right) {
      [27 insn w/o sources]
------------------------------------------------------ transition #8 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,1/4,isCascaded:false}
      [119 insn w/o sources]
  DiningPhil.java:32            : this.left = left;
  DiningPhil.java:33            : this.right = right;
------------------------------------------------------ transition #9 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"EXPOSE" ,1/4,isCascaded:false}
  DiningPhil.java:33            : this.right = right;
  DiningPhil.java:34            : start();
      [1 insn w/o sources]
------------------------------------------------------ transition #10 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"START" ,1/5,isCascaded:false}
      [2 insn w/o sources]
  DiningPhil.java:35            : }
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:54            : for (int i = 0; i < N; i++) {
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:31            : public Philosopher(Fork left, Fork right) {
      [27 insn w/o sources]
------------------------------------------------------ transition #11 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,1/5,isCascaded:false}
      [119 insn w/o sources]
  DiningPhil.java:32            : this.left = left;
  DiningPhil.java:33            : this.right = right;
  DiningPhil.java:34            : start();
      [1 insn w/o sources]
------------------------------------------------------ transition #12 thread: 0
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"START" ,1/6,isCascaded:false}
      [2 insn w/o sources]
  DiningPhil.java:35            : }
  DiningPhil.java:55            : new Philosopher(forks[i], forks[(i + 1) % N]);
  DiningPhil.java:54            : for (int i = 0; i < N; i++) {
  DiningPhil.java:57            : }
  DiningPhil.java:3             : // Copyright (C) 2006 United States Government as represented by the
------------------------------------------------------ transition #13 thread: 1
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"TERMINATE" ,1/5,isCascaded:false}
      [1 insn w/o sources]
  DiningPhil.java:1             :
  DiningPhil.java:39            : synchronized (left) {
------------------------------------------------------ transition #14 thread: 1
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,1/5,isCascaded:false}
  DiningPhil.java:39            : synchronized (left) {
  DiningPhil.java:40            : synchronized (right) {
------------------------------------------------------ transition #15 thread: 2
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/5,isCascaded:false}
      [1 insn w/o sources]
  DiningPhil.java:1             :
  DiningPhil.java:39            : synchronized (left) {
```

```
------------------------------------------------------ transition #16 thread: 2
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/5,isCascaded:false}
  DiningPhil.java:39            : synchronized (left) {
  DiningPhil.java:40            : synchronized (right) {
------------------------------------------------------ transition #17 thread: 3
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/4,isCascaded:false}
      [1 insn w/o sources]
  DiningPhil.java:1             :
  DiningPhil.java:39            : synchronized (left) {
------------------------------------------------------ transition #18 thread: 3
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/4,isCascaded:false}
  DiningPhil.java:39            : synchronized (left) {
  DiningPhil.java:40            : synchronized (right) {
------------------------------------------------------ transition #19 thread: 4
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/3,isCascaded:false}
      [1 insn w/o sources]
  DiningPhil.java:1             :
  DiningPhil.java:39            : synchronized (left) {
------------------------------------------------------ transition #20 thread: 4
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/3,isCascaded:false}
  DiningPhil.java:39            : synchronized (left) {
  DiningPhil.java:40            : synchronized (right) {
------------------------------------------------------ transition #21 thread: 5
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/2,isCascaded:false}
      [1 insn w/o sources]
  DiningPhil.java:1             :
  DiningPhil.java:39            : synchronized (left) {
------------------------------------------------------ transition #22 thread: 5
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"LOCK" ,2/2,isCascaded:false}
  DiningPhil.java:39            : synchronized (left) {
  DiningPhil.java:40            : synchronized (right) {

====================================================== snapshot #1
thread DiningPhil$Philosopher:{id:1,name:Thread-
1,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  owned locks:DiningPhil$Fork@166
  blocked on: DiningPhil$Fork@167
  call stack:
        at DiningPhil$Philosopher.run(DiningPhil.java:40)

thread DiningPhil$Philosopher:{id:2,name:Thread-
2,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  owned locks:DiningPhil$Fork@167
  blocked on: DiningPhil$Fork@168
  call stack:
        at DiningPhil$Philosopher.run(DiningPhil.java:40)

thread DiningPhil$Philosopher:{id:3,name:Thread-
3,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  owned locks:DiningPhil$Fork@168
  blocked on: DiningPhil$Fork@169
  call stack:
        at DiningPhil$Philosopher.run(DiningPhil.java:40)

thread DiningPhil$Philosopher:{id:4,name:Thread-
4,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  owned locks:DiningPhil$Fork@169
  blocked on: DiningPhil$Fork@16a
  call stack:
        at DiningPhil$Philosopher.run(DiningPhil.java:40)

thread DiningPhil$Philosopher:{id:5,name:Thread-
5,status:BLOCKED,priority:5,isDaemon:false,lockCount:0,suspendCount:0}
  owned locks:DiningPhil$Fork@16a
  blocked on: DiningPhil$Fork@166
  call stack:
        at DiningPhil$Philosopher.run(DiningPhil.java:40)


====================================================== results
error #1: gov.nasa.jpf.vm.NotDeadlockedProperty "deadlock encountered:   thread DiningPhil$Philosop..."

====================================================== statistics
elapsed time:       00:00:00
states:             new=2535,visited=5671,backtracked=8183,end=29
search:             maxDepth=29,constraints=0
choice generators:  thread=2534 (signal=0,lock=1608,sharedRef=3,threadApi=5,reschedule=918), data=0
heap:               new=408,released=19609,maxLive=391,gcCycles=8206
instructions:       55535
max memory:         155MB
loaded code:        classes=65,methods=1484
```

```
========================================================= search finished: 5/5/21 8:56 AM
```