

# Summary

**Sink States:**0( $0 \times 10^0$ )

Table 1: Pulse Analysis Summary

Classes	Methods	States	Unsatisfiable Clauses	Unreachable States	Possible concurrent Methods	Total. no. of pairs	No. of concurrent pairs	Percentage of concurrent Methods
JGFMonteCarloBenchSizeA	2	1	0	0	1	3	1	33
JGFMonteCarloBench	6	1	0	0	5	21	15	71
JGFInstrumentor	4	1	0	0	0	10	0	0
CallAppDemo	3	1	0	0	2	6	3	50
AppDemo	15	1	0	0	14	120	105	88
Universal	11	1	0	0	10	66	49	74
JGFTimer	2	1	0	0	0	3	0	0
RatePath	11	1	0	0	10	66	55	83
PathId	11	1	0	0	10	66	10	15
PriceStock	3	1	0	0	2	6	3	50
MonteCarloPath	18	1	0	0	17	171	153	89
ToInitAllTasks	20	1	0	0	19	210	135	64
ReturnPath	21	1	0	0	20	231	20	9
DemoException	1	1	0	0	0	1	0	0
ToResult	14	1	0	0	13	105	70	67
ToTask	5	1	0	0	4	15	7	47
Utilities	4	1	0	0	3	10	6	60
Total Classes=17	151	17	0	0	130	1110	632	57

## Contents

<b>1</b>	<b>JGFMonteCarloBenchSizeA</b>	<b>3</b>
<b>2</b>	<b>JGFMonteCarloBench</b>	<b>4</b>
<b>3</b>	<b>JGFInstrumentor</b>	<b>5</b>
<b>4</b>	<b>CallAppDemo</b>	<b>6</b>
<b>5</b>	<b>AppDemo</b>	<b>7</b>
<b>6</b>	<b>Universal</b>	<b>8</b>
<b>7</b>	<b>JGFTimer</b>	<b>9</b>
<b>8</b>	<b>RatePath</b>	<b>10</b>
<b>9</b>	<b>PathId</b>	<b>11</b>
<b>10</b>	<b>PriceStock</b>	<b>12</b>
<b>11</b>	<b>MonteCarloPath</b>	<b>13</b>
<b>12</b>	<b>ToInitAllTasks</b>	<b>15</b>
<b>13</b>	<b>ReturnPath</b>	<b>17</b>
<b>14</b>	<b>DemoException</b>	<b>19</b>
<b>15</b>	<b>ToResult</b>	<b>20</b>
<b>16</b>	<b>ToTask</b>	<b>21</b>
<b>17</b>	<b>Utilities</b>	<b>22</b>
<b>18</b>	<b>Abbreviation</b>	<b>23</b>
<b>19</b>	<b>Annotated Version of Sequential Java Program generated by Sip4j</b>	<b>24</b>

# 1 JGFMonteCarloBenchSizeA

Table 2: Methods Requires Clause Satisfiability

Method	Satisfiability
JGFMonteCarloBenchSizeA	✓
main	✓

Table 3: State Transition Matrix

	alive
alive	↑

Table 4: Methods Concurrency Matrix

	JGFMonteCarloBenchSizeA	main
JGFMonteCarloBenchSizeA	⌘	⌘
main	⌘	

## 2 JGFMonteCarloBench

Table 5: Methods Requires Clause Satisfiability

Method	Satisfiability
JGFMonteCarloBench	✓
JGFrun	✓
JGFinitialise	✓
JGFvalidate	✓
JGFtidyup	✓
JGFsetsize	✓

Table 6: State Transition Matrix

	alive
alive	↑

Table 7: Methods Concurrency Matrix

	JGFMonteCarloBench	JGFrun	JGFinitialise	JGFvalidate	JGFtidyup	JGFsetsize
JGFMonteCarloBench	✗	✗	✗	✗	✗	✗
JGFrun	✗					
JGFinitialise	✗					
JGFvalidate	✗					
JGFtidyup	✗					
JGFsetsize	✗					

### 3 JGFInstrumentor

Table 8: Methods Requires Clause Satisfiability

Method	Satisfiability
JGFInstrumentor	✓
addTimer	✓
addOpsToTimer	✓
printTimer	✓

Table 9: State Transition Matrix

	alive
alive	↑

Table 10: Methods Concurrency Matrix

	JGFInstrumentor	addTimer	addOpsToTimer	printTimer
JGFInstrumentor	⌘	⌘	⌘	⌘
addTimer	⌘	⌘	⌘	⌘
addOpsToTimer	⌘	⌘	⌘	⌘
printTimer	⌘	⌘	⌘	⌘

## 4 CallAppDemo

Table 11: Methods Requires Clause Satisfiability

Method	Satisfiability
CallAppDemo	✓
initialise	✓
runiters	✓

Table 12: State Transition Matrix

	alive
alive	↑

Table 13: Methods Concurrency Matrix

	CallAppDemo	initialise	runiters
CallAppDemo	⌈	⌈	⌈
initialise	⌈	⌈	⌈
runiters	⌈	⌈	⌈

## 5 AppDemo

Table 14: Methods Requires Clause Satisfiability

Method	Satisfiability
AppDemo	✓
runSerial	✓
initTasks	✓
processResults	✓
setdataDirname	✓
getdataFilename	✓
setdataFilename	✓
getnTimeStepsMC	✓
setnTimeStepsMC	✓
getnRunsMC	✓
setnRunsMC	✓
gettasks	✓
settasks	✓
getresults	✓
setresults	✓

Table 15: State Transition Matrix

	alive
alive	↑

Table 16: Methods Concurrency Matrix

	AppDemo	runSerial	initTasks	processResults	setdataDirname	getdataFilename	setdataFilename	getnTimeStepsMC	setnTimeStepsMC	getnRunsMC	setnRunsMC	gettasks	settasks	getresults	setresults
AppDemo	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
runSerial	⌘														
initTasks	⌘														
processResults	⌘														
setdataDirname	⌘														
getdataFilename	⌘														
setdataFilename	⌘														
getnTimeStepsMC	⌘														
setnTimeStepsMC	⌘														
getnRunsMC	⌘														
setnRunsMC	⌘														
gettasks	⌘														
settasks	⌘														
getresults	⌘														
setresults	⌘														

## 6 Universal

Table 17: Methods Requires Clause Satisfiability

Method	Satisfiability
Universal	✓
setprompt	✓
setDEBUG	✓
dbgPrintln	✓
getDEBUG	✓
getUNIVERSALDEBUG	✓
setUNIVERSALDEBUG	✓
getprompt	✓
dbgPrint	✓
errPrintln	✓
errPrint	✓

Table 18: State Transition Matrix

	alive
alive	↑

Table 19: Methods Concurrency Matrix

	Universal	setprompt	setDEBUG	dbgPrintln	getDEBUG	getUNIVERSALDEBUG	setUNIVERSALDEBUG	getprompt	dbgPrint	errPrintln	errPrint
Universal	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setprompt	⌘	⌘	⌘				⌘				
setDEBUG	⌘	⌘	⌘				⌘				
dbgPrintln	⌘										
getDEBUG	⌘										
getUNIVERSALDEBUG	⌘										
setUNIVERSALDEBUG	⌘	⌘	⌘				⌘				
getprompt	⌘										
dbgPrint	⌘										
errPrintln	⌘										
errPrint	⌘										



## 7 JGFTimer

Table 20: Methods Requires Clause Satisfiability

Method	Satisfiability
JGFTimer	✓
addops	✓

Table 21: State Transition Matrix

	alive
alive	↑

Table 22: Methods Concurrency Matrix

	JGFTimer	addops
JGFTimer	⧻	⧻
addops	⧻	⧻

## 8 RatePath

Table 23: Methods Requires Clause Satisfiability

Method	Satisfiability
RatePath	✓
readRatesFile	✓
incpathValue	✓
getpathValue	✓
setpathValue	✓
getpathDate	✓
setpathDate	✓
getEndPathValue	✓
getPathValue	✓
getReturnCompounded	✓
getReturnNonCompounded	✓

Table 24: State Transition Matrix

	alive
alive	↑

Table 25: Methods Concurrency Matrix

	RatePath	readRatesFile	incpathValue	getpathValue	setpathValue	getpathDate	setpathDate	getEndPathValue	getPathValue	getReturnCompounded	getReturnNonCompounded
RatePath	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
readRatesFile	⌘										
incpathValue	⌘										
getpathValue	⌘										
setpathValue	⌘										
getpathDate	⌘										
setpathDate	⌘										
getEndPathValue	⌘										
getPathValue	⌘										
getReturnCompounded	⌘										
getReturnNonCompounded	⌘										

## 9 PathId

Table 26: Methods Requires Clause Satisfiability

Method	Satisfiability
PathId	✓
setname	✓
setendDate	✓
setdTime	✓
getname	✓
setstartDate	✓
getstartDate	✓
getendDate	✓
getdTime	✓
copyInstanceVariables	✓
dbgDumpFields	✓

Table 27: State Transition Matrix

	alive
alive	↑

Table 28: Methods Concurrency Matrix

	PathId	setname	setendDate	setdTime	getname	setstartDate	getstartDate	getendDate	getdTime	copyInstanceVariables	dbgDumpFields
PathId	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setname	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setendDate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setdTime	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getname	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setstartDate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getstartDate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getendDate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getdTime	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
copyInstanceVariables	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
dbgDumpFields	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

## 10 PriceStock

Table 29: Methods Requires Clause Satisfiability

Method	Satisfiability
PriceStock	✓
setInitAllTasks	✓
run	✓

Table 30: State Transition Matrix

	alive
alive	↑

Table 31: Methods Concurrency Matrix

	PriceStock	setInitAllTasks	run
PriceStock	⌈	⌈	⌈
setInitAllTasks	⌈	⌈	⌈
run	⌈	⌈	⌈

## 11 MonteCarloPath

Table 32: Methods Requires Clause Satisfiability

Method	Satisfiability
MonteCarloPath	✓
copyInstanceVariables	✓
setpathValue	✓
setfluctuations	✓
getnTimeSteps	✓
computePathValue	✓
getpathValue	✓
getreturnDefinition	✓
setreturnDefinition	✓
getexpectedReturnRate	✓
setexpectedReturnRate	✓
getvolatility	✓
setvolatility	✓
setnTimeSteps	✓
getpathStartValue	✓
setpathStartValue	✓
getRatePath	✓
computeFluctuationsGaussian	✓

Table 33: State Transition Matrix

	alive
alive	↑

Table 34: Methods Concurrency Matrix

	MonteCarloPath	copyInstanceVariables	setpathValue	setfluctuations	getnTimeSteps	computePathValue	getpathValue	getreturnDefinition	setreturnDefinition	getexpectedReturnRate	setexpectedReturnRate	getvolatility	setvolatility	setnTimeSteps	getpathStartValue	setpathStartValue	getRatePath	computeFluctuationsGaussian
MonteCarloPath	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
copyInstanceVariables	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setpathValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setfluctuations	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getnTimeSteps	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
computePathValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getpathValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getreturnDefinition	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

setreturnDefinition	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
getexpectedReturnRate	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
setexpectedReturnRate	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
getvolatility	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
setvolatility	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
setnTimeSteps	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
getpathStartValue	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
setpathStartValue	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
getRatePath	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡
computeFluctuationsGaussian	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡	⚡

## 12 ToInitAllTasks

Table 35: Methods Requires Clause Satisfiability

Method	Satisfiability
ToInitAllTasks	✓
getname	✓
getnTimeSteps	✓
setheader	✓
setname	✓
getstartDate	✓
setstartDate	✓
getendDate	✓
setendDate	✓
getdTime	✓
setDTime	✓
getreturnDefinition	✓
setReturnDefinition	✓
getexpectedReturnRate	✓
setExpectedReturnRate	✓
getvolatility	✓
setVolatility	✓
setnTimeSteps	✓
getpathStartValue	✓
setpathStartValue	✓

Table 36: State Transition Matrix

	alive
alive	↑

Table 37: Methods Concurrency Matrix

	ToInitAllTasks	getname	getnTimeSteps	setheader	setname	getstartDate	setstartDate	getendDate	setendDate	getdTime	setDTime	getreturnDefinition	setReturnDefinition	getexpectedReturnRate	setExpectedReturnRate	getvolatility	setVolatility	setnTimeSteps	getpathStartValue	setpathStartValue
ToInitAllTasks	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getname	⌘																			
getnTimeSteps	⌘																			
setheader	⌘			⌘	⌘		⌘		⌘		⌘		⌘		⌘		⌘	⌘		⌘
setname	⌘			⌘	⌘		⌘		⌘		⌘		⌘		⌘		⌘	⌘		⌘
getstartDate	⌘																			
setstartDate	⌘			⌘	⌘		⌘		⌘		⌘		⌘		⌘		⌘	⌘		⌘
getendDate	⌘																			

setendDate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getdTime	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setDTime	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getreturnDefinition	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setReturnDefinition	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getexpectedReturnRate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setExpectedReturnRate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getvolatility	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setVolatility	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setnTimeSteps	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getpathStartValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setpathStartValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘



## 13 ReturnPath

Table 38: Methods Requires Clause Satisfiability

Method	Satisfiability
ReturnPath	✓
estimatePath	✓
computeMean	✓
computeVariance	✓
computeExpectedReturnRate	✓
getexpectedReturnRate	✓
getvolatility	✓
getvolatility2	✓
setpathValue	✓
getnPathValue	✓
setnPathValue	✓
getreturnDefinition	✓
setreturnDefinition	✓
setexpectedReturnRate	✓
setvolatility	✓
setvolatility2	✓
getmean	✓
setmean	✓
getvariance	✓
setvariance	✓
dbgDumpFields	✓

Table 39: State Transition Matrix

	alive
alive	↑

Table 40: Methods Concurrency Matrix

	ReturnPath	estimatePath	computeMean	computeVariance	computeExpectedReturnRate	getexpectedReturnRate	getvolatility	getvolatility2	setpathValue	getnPathValue	setnPathValue	getreturnDefinition	setreturnDefinition	setexpectedReturnRate	setvolatility	setvolatility2	getmean	setmean	getvariance	setvariance
ReturnPath	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
estimatePath	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
computeMean	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
computeVariance	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
computeExpectedReturnRate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

getexpectedReturnRate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getvolatility	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getvolatility2	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setpathValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getnPathValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setnPathValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getreturnDefinition	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setreturnDefinition	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setexpectedReturnRate	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setvolatility	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setvolatility2	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getmean	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setmean	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getvariance	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setvariance	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
dbgDumpFields	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

## 14 DemoException

Table 41: Methods Requires Clause Satisfiability

Method	Satisfiability
DemoException	✓

Table 42: State Transition Matrix

	alive
alive	↑

## 15 ToResult

Table 43: Methods Requires Clause Satisfiability

Method	Satisfiability
ToResult	✓
getexpectedReturnRate	✓
getvolatility	✓
toString	✓
getheader	✓
setheader	✓
setexpectedReturnRate	✓
setvolatility	✓
getVolatility2	✓
setvolatility2	✓
getfinalStockPrice	✓
setfinalStockPrice	✓
getpathValue	✓
setpathValue	✓

Table 44: State Transition Matrix

	alive
alive	↑

Table 45: Methods Concurrency Matrix

	ToResult	getexpectedReturnRate	getvolatility	toString	getheader	setheader	setexpectedReturnRate	setvolatility	getVolatility2	setvolatility2	getfinalStockPrice	setfinalStockPrice	getpathValue	setpathValue
ToResult	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getexpectedReturnRate	⌘													
getvolatility	⌘													
toString	⌘													
getheader	⌘													
setheader	⌘					⌘	⌘	⌘		⌘		⌘		⌘
setexpectedReturnRate	⌘					⌘	⌘	⌘		⌘		⌘		⌘
setvolatility	⌘					⌘	⌘	⌘		⌘		⌘		⌘
getVolatility2	⌘													
setvolatility2	⌘					⌘	⌘	⌘		⌘		⌘		⌘
getfinalStockPrice	⌘													
setfinalStockPrice	⌘					⌘	⌘	⌘		⌘		⌘		⌘
getpathValue	⌘													
setpathValue	⌘					⌘	⌘	⌘		⌘		⌘		⌘

## 16 ToTask

Table 46: Methods Requires Clause Satisfiability

Method	Satisfiability
ToTask	✓
getheader	✓
setheader	✓
getrandomSeed	✓
setrandomSeed	✓

Table 47: State Transition Matrix

	alive
alive	↑

Table 48: Methods Concurrency Matrix

	ToTask	getheader	setheader	getrandomSeed	setrandomSeed
ToTask	⌘	⌘	⌘	⌘	⌘
getheader	⌘		⌘		
setheader	⌘		⌘		⌘
getrandomSeed	⌘				
setrandomSeed	⌘		⌘		⌘

## 17 Utilities

Table 49: Methods Requires Clause Satisfiability

Method	Satisfiability
Utilities	✓
which	✓
splitString	✓
joinString	✓

Table 50: State Transition Matrix

	alive
alive	↑

Table 51: Methods Concurrency Matrix

	Utilities	which	splitString	joinString
Utilities	⌘	⌘	⌘	⌘
which	⌘			
splitString	⌘			
joinString	⌘			

## 18 Abbreviation

Table 52: Used Abbreviation

Symbol	Meaning
✓	requires clause of the method is satisfiable
✗	requires clause of the method is unsatisfiable
↑	The row-state can be transitioned to the column-state
✕	The row-state cannot be transitioned to the column-state
	The row-method can be possibly executed parallel with the column-method
⋈	The row-method cannot be executed parallel with the column-method

## 19 Annotated Version of Sequential Java Program generated by Sip4j

```
1 package outputs;
2 import edu.cmu.cs.plural.annot.*;
3
4 @ClassStates({@State(name = "alive")})
5 class JGFMonteCarloBenchSizeA {
6   @Perm(ensures="unique(this) in alive")
7   JGFMonteCarloBenchSizeA() { }
8
9   @Perm(requires="none(this) in alive",
10  ensures="unique(this) in alive")
11   void main(String argv[]) {
12   }
13
14 }ENDOFCLASS
15
16 @ClassStates({@State(name = "alive")})
17
18 class JGFMonteCarloBench {
19   @Perm(ensures="unique(this) in alive")
20   JGFMonteCarloBench() { }
21
22   @Perm(requires="none(this) in alive",
23  ensures="unique(this) in alive")
24   public void JGFrun(int size) {
25   }
26   @Perm(requires="none(this) in alive",
27  ensures="unique(this) in alive")
28   public void JGFinitialise() {
29   }
30   @Perm(requires="pure(this) in alive",
31  ensures="pure(this) in alive")
32   public void JGFvalidate() {
33   }
34   @Perm(requires="unique(this) in alive",
35  ensures="unique(this) in alive")
36   public void JGFtidyup() {
37   }
38   @Perm(requires="full(this) in alive",
39  ensures="full(this) in alive")
40   public void JGFsetsize(int size) {
41   }
42
43 }ENDOFCLASS
44
45 @ClassStates({@State(name = "alive")})
46
47 class JGFInstrumentor {
48   @Perm(ensures="unique(this) in alive")
49   JGFInstrumentor() { }
50
51   @Perm(requires="full(this) in alive",
52  ensures="full(this) in alive")
53   void addTimer(String name, String opname, int size) {
54   }
55   @Perm(requires="full(this) in alive",
56  ensures="full(this) in alive")
57   void addOpsToTimer(String name, double count) {
58   }
59   @Perm(requires="full(this) in alive",
60  ensures="full(this) in alive")
61   void printTimer(String name) {
62   }
63
64 }ENDOFCLASS
65
66 @ClassStates({@State(name = "alive")})
67
68 class CallAppDemo {
69   @Perm(ensures="unique(this) in alive")
70   CallAppDemo() { }
71
72   @Perm(requires="none(this) in alive",
73  ensures="unique(this) in alive")
74   public void initialise() {
75   }
```



```

76 @Perm(requires="none(this) in alive",
77 ensures="unique(this) in alive")
78 public void runiters() {
79 }
80
81 }ENDOFCLASS
82
83 @ClassStates({@State(name = "alive")})
84
85 class AppDemo {
86 @Perm(ensures="unique(this) in alive")
87 AppDemo() { }
88
89 @Perm(requires="none(this) in alive",
90 ensures="unique(this) in alive")
91 public void runSerial() {
92 }
93 @Perm(requires="none(this) in alive",
94 ensures="unique(this) in alive")
95 private void initTasks(int nRunsMC) {
96 }
97 @Perm(requires="full(this) in alive",
98 ensures="full(this) in alive")
99 private void processResults() {
100 }
101 @Perm(requires="full(this) in alive",
102 ensures="full(this) in alive")
103 public void setdataDirname(String dataDirname) {
104 }
105 @Perm(requires="pure(this) in alive",
106 ensures="pure(this) in alive")
107 public String getdataFilename() {
108 return null;
109 }
110 @Perm(requires="full(this) in alive",
111 ensures="full(this) in alive")
112 public void setdataFilename(String dataFilename) {
113 }
114 @Perm(requires="pure(this) in alive",
115 ensures="pure(this) in alive")
116 public int getnTimeStepsMC() {
117 return 0;
118 }
119 @Perm(requires="full(this) in alive",
120 ensures="full(this) in alive")
121 public void setnTimeStepsMC(int nTimeStepsMC) {
122 }
123 @Perm(requires="pure(this) in alive",
124 ensures="pure(this) in alive")
125 public int getnRunsMC() {
126 return 0;
127 }
128 @Perm(requires="full(this) in alive",
129 ensures="full(this) in alive")
130 public void setnRunsMC(int nRunsMC) {
131 }
132 @Perm(requires="pure(this) in alive",
133 ensures="pure(this) in alive")
134 public Vector gettasks() {
135 return null;
136 }
137 @Perm(requires="full(this) in alive",
138 ensures="full(this) in alive")
139 public void settasks(Vector tasks) {
140 }
141 @Perm(requires="pure(this) in alive",
142 ensures="pure(this) in alive")
143 public Vector getresults() {
144 return null;
145 }
146 @Perm(requires="full(this) in alive",
147 ensures="full(this) in alive")
148 public void setresults(Vector results) {
149 }
150
151 }ENDOFCLASS
152
153 @ClassStates({@State(name = "alive")})
154
155 class Universal {
156 @Perm(ensures="unique(this) in alive")

```

```

157 Universal() { }

159 @Perm(requires="full(this) in alive",
160 ensures="full(this) in alive")
161 public void setprompt(String prompt) {
162 }
163 @Perm(requires="full(this) in alive",
164 ensures="full(this) in alive")
165 public void setDEBUG(boolean DEBUG) {
166 }
167 @Perm(requires="pure(this) in alive",
168 ensures="pure(this) in alive")
169 public void dbgPrintln(String s) {
170 }
171 @Perm(requires="pure(this) in alive",
172 ensures="pure(this) in alive")
173 public boolean getDEBUG() {
174     return 0;
175 }
176 @Perm(requires="pure(this) in alive",
177 ensures="pure(this) in alive")
178 public boolean getUNIVERSALDEBUG() {
179     return 0;
180 }
181 @Perm(requires="full(this) in alive",
182 ensures="full(this) in alive")
183 public void setUNIVERSALDEBUG(boolean UNIVERSAL_DEBUG) {
184 }
185 @Perm(requires="pure(this) in alive",
186 ensures="pure(this) in alive")
187 public String getprompt() {
188     return null;
189 }
190 @Perm(requires="pure(this) in alive",
191 ensures="pure(this) in alive")
192 public void dbgPrint(String s) {
193 }
194 @Perm(requires="pure(this) in alive",
195 ensures="pure(this) in alive")
196 public void errPrintln(String s) {
197 }
198 @Perm(requires="pure(this) in alive",
199 ensures="pure(this) in alive")
200 public void errPrint(String s) {
201 }

203 }ENDOFCLASS

205 @ClassStates({@State(name = "alive")})

207 class JGFTimer {
208     @Perm(ensures="unique(this) in alive")
209     JGFTimer() { }

211     @Perm(requires="full(this) in alive",
212     ensures="full(this) in alive")
213     public void addops(double count) {
214     }

216 }ENDOFCLASS

218 @ClassStates({@State(name = "alive")})

220 class RatePath {
221     @Perm(ensures="unique(this) in alive")
222     RatePath() { }

224     @Perm(requires="full(this) in alive",
225     ensures="full(this) in alive")
226     private void readRatesFile(String dirName, String filename) {
227     }
228     @Perm(requires="full(this) in alive",
229     ensures="full(this) in alive")
230     public void incpathValue(double scale) {
231     }
232     @Perm(requires="pure(this) in alive",
233     ensures="pure(this) in alive")
234     public double[] getpathValue() {
235         return null;
236     }
237     @Perm(requires="full(this) in alive",

```

```

238 ensures="full(this) in alive")
239 public void setpathValue(double[] pathValue) {
240 }
241 @Perm(requires="pure(this) in alive",
242 ensures="pure(this) in alive")
243 public int[] getpathDate() {
244     return null;
245 }
246 @Perm(requires="full(this) in alive",
247 ensures="full(this) in alive")
248 public void setpathDate(int[] pathDate) {
249 }
250 @Perm(requires="pure(this) in alive",
251 ensures="pure(this) in alive")
252 public double getEndPathValue() {
253     return 0;
254 }
255 @Perm(requires="pure(this) in alive",
256 ensures="pure(this) in alive")
257 public double getPathValue(int index) {
258     return 0;
259 }
260 @Perm(requires="none(this) in alive",
261 ensures="unique(this) in alive")
262 public ReturnPath getReturnCompounded() {
263     return null;
264 }
265 @Perm(requires="full(this) in alive",
266 ensures="full(this) in alive")
267 public ReturnPath getReturnNonCompounded() {
268     return null;
269 }
270 }ENDOFCLASS
271
272 @ClassStates({@State(name = "alive")})
273
274 class PathId {
275 @Perm(ensures="unique(this) in alive")
276 PathId() { }
277
278 @Perm(requires="full(this) in alive",
279 ensures="full(this) in alive")
280 public void setname(String name) {
281 }
282 @Perm(requires="full(this) in alive",
283 ensures="full(this) in alive")
284 public void setendDate(int endDate) {
285 }
286 @Perm(requires="full(this) in alive",
287 ensures="full(this) in alive")
288 public void setdTime(double dTime) {
289 }
290 @Perm(requires="full(this) in alive",
291 ensures="full(this) in alive")
292 public String getname() {
293     return null;
294 }
295 @Perm(requires="full(this) in alive",
296 ensures="full(this) in alive")
297 public void setstartDate(int startDate) {
298 }
299 @Perm(requires="full(this) in alive",
300 ensures="full(this) in alive")
301 public int getstartDate() {
302     return 0;
303 }
304 @Perm(requires="full(this) in alive",
305 ensures="full(this) in alive")
306 public int getendDate() {
307     return 0;
308 }
309 @Perm(requires="full(this) in alive",
310 ensures="full(this) in alive")
311 public double getdTime() {
312     return 0;
313 }
314 @Perm(requires="full(this) in alive",
315 ensures="full(this) in alive")
316 public void copyInstanceVariables(PathId obj) {
317 }
318 }

```

```

319 @Perm(requires="pure(this) in alive",
320 ensures="pure(this) in alive")
321 public void dbgDumpFields() {
322 }
323
324 }ENDOFCLASS
325
326 @ClassStates({@State(name = "alive")})
327
328 class PriceStock {
329 @Perm(ensures="unique(this) in alive")
330 PriceStock() { }
331
332 @Perm(requires="full(this) in alive",
333 ensures="full(this) in alive")
334 public void setInitAllTasks(Object obj) {
335 }
336 @Perm(requires="none(this) in alive",
337 ensures="unique(this) in alive")
338 public void run() {
339 }
340
341 }ENDOFCLASS
342
343 @ClassStates({@State(name = "alive")})
344
345 class MonteCarloPath {
346 @Perm(ensures="unique(this) in alive")
347 MonteCarloPath() { }
348
349 @Perm(requires="full(this) in alive",
350 ensures="full(this) in alive")
351 private void copyInstanceVariables(ReturnPath obj) {
352 }
353 @Perm(requires="full(this) in alive",
354 ensures="full(this) in alive")
355 public void setpathValue(double[] pathValue) {
356 }
357 @Perm(requires="full(this) in alive",
358 ensures="full(this) in alive")
359 public void setfluctuations(double[] fluctuations) {
360 }
361 @Perm(requires="full(this) in alive",
362 ensures="full(this) in alive")
363 public int getnTimeSteps() {
364 return 0;
365 }
366 @Perm(requires="full(this) in alive",
367 ensures="full(this) in alive")
368 public void computePathValue(double startValue) {
369 }
370 @Perm(requires="full(this) in alive",
371 ensures="full(this) in alive")
372 public double[] getpathValue() {
373 return null;
374 }
375 @Perm(requires="full(this) in alive",
376 ensures="full(this) in alive")
377 public int getreturnDefinition() {
378 return 0;
379 }
380 @Perm(requires="full(this) in alive",
381 ensures="full(this) in alive")
382 public void setreturnDefinition(int returnDefinition) {
383 }
384 @Perm(requires="full(this) in alive",
385 ensures="full(this) in alive")
386 public double getexpectedReturnRate() {
387 return 0;
388 }
389 @Perm(requires="full(this) in alive",
390 ensures="full(this) in alive")
391 public void setexpectedReturnRate(double expectedReturnRate) {
392 }
393 @Perm(requires="full(this) in alive",
394 ensures="full(this) in alive")
395 public double getvolatility() {
396 return 0;
397 }
398 @Perm(requires="full(this) in alive",
399 ensures="full(this) in alive")

```

```

400 public void setvolatility(double volatility) {
401 }
402 @Perm(requires="full(this) in alive",
403 ensures="full(this) in alive")
404 public void setnTimeSteps(int nTimeSteps) {
405 }
406 @Perm(requires="full(this) in alive",
407 ensures="full(this) in alive")
408 public double getpathStartValue() {
409     return 0;
410 }
411 @Perm(requires="full(this) in alive",
412 ensures="full(this) in alive")
413 public void setpathStartValue(double pathStartValue) {
414 }
415 @Perm(requires="none(this) in alive",
416 ensures="unique(this) in alive")
417 public RatePath getRatePath() {
418     return null;
419 }
420 @Perm(requires="full(this) in alive",
421 ensures="full(this) in alive")
422 public void computeFluctuationsGaussian(long randomSeed) {
423 }
424
425 }ENDOFCLASS
426
427 @ClassStates({@State(name = "alive")})
428
429 class ToInitAllTasks {
430 @Perm(ensures="unique(this) in alive")
431 ToInitAllTasks() { }
432
433 @Perm(requires="pure(this) in alive",
434 ensures="pure(this) in alive")
435 public String getname() {
436     return null;
437 }
438 @Perm(requires="pure(this) in alive",
439 ensures="pure(this) in alive")
440 public int getnTimeSteps() {
441     return 0;
442 }
443 @Perm(requires="full(this) in alive",
444 ensures="full(this) in alive")
445 public void setheader(String header) {
446 }
447 @Perm(requires="full(this) in alive",
448 ensures="full(this) in alive")
449 public void setname(String name) {
450 }
451 @Perm(requires="pure(this) in alive",
452 ensures="pure(this) in alive")
453 public int getstartDate() {
454     return 0;
455 }
456 @Perm(requires="full(this) in alive",
457 ensures="full(this) in alive")
458 public void setstartDate(int startDate) {
459 }
460 @Perm(requires="pure(this) in alive",
461 ensures="pure(this) in alive")
462 public int getendDate() {
463     return 0;
464 }
465 @Perm(requires="full(this) in alive",
466 ensures="full(this) in alive")
467 public void setendDate(int endDate) {
468 }
469 @Perm(requires="pure(this) in alive",
470 ensures="pure(this) in alive")
471 public double getdTime() {
472     return 0;
473 }
474 @Perm(requires="full(this) in alive",
475 ensures="full(this) in alive")
476 public void setDTime(double dTime) {
477 }
478 @Perm(requires="pure(this) in alive",
479 ensures="pure(this) in alive")
480 public int getreturnDefinition() {

```

```

481     return 0;
482 }
483 @Perm(requires="full(this) in alive",
484 ensures="full(this) in alive")
485 public void setReturnDefinition(int returnDefinition) {
486 }
487 @Perm(requires="pure(this) in alive",
488 ensures="pure(this) in alive")
489 public double getexpectedReturnRate() {
490     return 0;
491 }
492 @Perm(requires="full(this) in alive",
493 ensures="full(this) in alive")
494 public void setExpectedReturnRate(double expectedReturnRate) {
495 }
496 @Perm(requires="pure(this) in alive",
497 ensures="pure(this) in alive")
498 public double getvolatility() {
499     return 0;
500 }
501 @Perm(requires="full(this) in alive",
502 ensures="full(this) in alive")
503 public void setVolatility(double volatility) {
504 }
505 @Perm(requires="full(this) in alive",
506 ensures="full(this) in alive")
507 public void setnTimeSteps(int nTimeSteps) {
508 }
509 @Perm(requires="pure(this) in alive",
510 ensures="pure(this) in alive")
511 public double getpathStartValue() {
512     return 0;
513 }
514 @Perm(requires="full(this) in alive",
515 ensures="full(this) in alive")
516 public void setpathStartValue(double pathStartValue) {
517 }
518
519 }ENDOFCLASS
520
521 @ClassStates({@State(name = "alive")})
522
523 class ReturnPath {
524 @Perm(ensures="unique(this) in alive")
525 ReturnPath() { }
526
527 @Perm(requires="full(this) in alive",
528 ensures="full(this) in alive")
529 public void estimatePath() {
530 }
531 @Perm(requires="full(this) in alive",
532 ensures="full(this) in alive")
533 public void computeMean() {
534 }
535 @Perm(requires="full(this) in alive",
536 ensures="full(this) in alive")
537 public void computeVariance() {
538 }
539 @Perm(requires="full(this) in alive",
540 ensures="full(this) in alive")
541 public void computeExpectedReturnRate() {
542 }
543 @Perm(requires="full(this) in alive",
544 ensures="full(this) in alive")
545 public double getexpectedReturnRate() {
546     return 0;
547 }
548 @Perm(requires="full(this) in alive",
549 ensures="full(this) in alive")
550 public double getvolatility() {
551     return 0;
552 }
553 @Perm(requires="full(this) in alive",
554 ensures="full(this) in alive")
555 public double getvolatility2() {
556     return 0;
557 }
558 @Perm(requires="full(this) in alive",
559 ensures="full(this) in alive")
560 public void setpathValue(double[] pathValue) {
561 }

```

```

562 @Perm(requires="full(this) in alive",
563 ensures="full(this) in alive")
564 public int getnPathValue() {
565     return 0;
566 }
567 @Perm(requires="full(this) in alive",
568 ensures="full(this) in alive")
569 public void setnPathValue(int nPathValue) {
570 }
571 @Perm(requires="full(this) in alive",
572 ensures="full(this) in alive")
573 public int getreturnDefinition() {
574     return 0;
575 }
576 @Perm(requires="full(this) in alive",
577 ensures="full(this) in alive")
578 public void setreturnDefinition(int returnDefinition) {
579 }
580 @Perm(requires="full(this) in alive",
581 ensures="full(this) in alive")
582 public void setexpectedReturnRate(double expectedReturnRate) {
583 }
584 @Perm(requires="full(this) in alive",
585 ensures="full(this) in alive")
586 public void setvolatility(double volatility) {
587 }
588 @Perm(requires="full(this) in alive",
589 ensures="full(this) in alive")
590 public void setvolatility2(double volatility2) {
591 }
592 @Perm(requires="full(this) in alive",
593 ensures="full(this) in alive")
594 public double getmean() {
595     return 0;
596 }
597 @Perm(requires="full(this) in alive",
598 ensures="full(this) in alive")
599 public void setmean(double mean) {
600 }
601 @Perm(requires="full(this) in alive",
602 ensures="full(this) in alive")
603 public double getvariance() {
604     return 0;
605 }
606 @Perm(requires="full(this) in alive",
607 ensures="full(this) in alive")
608 public void setvariance(double variance) {
609 }
610 @Perm(requires="pure(this) in alive",
611 ensures="pure(this) in alive")
612 public void dbgDumpFields() {
613 }
614
615 }ENDOFCLASS
616
617 @ClassStates({@State(name = "alive")})
618
619 class DemoException {
620 @Perm(ensures="unique(this) in alive")
621 DemoException() { }
622 }
623
624 }ENDOFCLASS
625
626 @ClassStates({@State(name = "alive")})
627
628 class ToResult {
629 @Perm(ensures="unique(this) in alive")
630 ToResult() { }
631 }
632
633 @Perm(requires="pure(this) in alive",
634 ensures="pure(this) in alive")
635 public double getexpectedReturnRate() {
636     return 0;
637 }
638 @Perm(requires="pure(this) in alive",
639 ensures="pure(this) in alive")
640 public double getvolatility() {
641     return 0;
642 }
643 @Perm(requires="pure(this) in alive",

```

```

643 ensures="pure(this) in alive")
644 public String toString() {
645     return null;
646 }
647 @Perm(requires="pure(this) in alive",
648 ensures="pure(this) in alive")
649 public String getheader() {
650     return null;
651 }
652 @Perm(requires="full(this) in alive",
653 ensures="full(this) in alive")
654 public void setheader(String header) {
655 }
656 @Perm(requires="full(this) in alive",
657 ensures="full(this) in alive")
658 public void setexpectedReturnRate(double expectedReturnRate) {
659 }
660 @Perm(requires="full(this) in alive",
661 ensures="full(this) in alive")
662 public void setvolatility(double volatility) {
663 }
664 @Perm(requires="pure(this) in alive",
665 ensures="pure(this) in alive")
666 public double getVolatility2() {
667     return 0;
668 }
669 @Perm(requires="full(this) in alive",
670 ensures="full(this) in alive")
671 public void setvolatility2(double volatility2) {
672 }
673 @Perm(requires="pure(this) in alive",
674 ensures="pure(this) in alive")
675 public double getfinalStockPrice() {
676     return 0;
677 }
678 @Perm(requires="full(this) in alive",
679 ensures="full(this) in alive")
680 public void setfinalStockPrice(double finalStockPrice) {
681 }
682 @Perm(requires="pure(this) in alive",
683 ensures="pure(this) in alive")
684 public double[] getpathValue() {
685     return null;
686 }
687 @Perm(requires="full(this) in alive",
688 ensures="full(this) in alive")
689 public void setpathValue(double[] pathValue) {
690 }
691
692 }ENDOFCLASS
693
694 @ClassStates({@State(name = "alive")})
695
696 class ToTask {
697     @Perm(ensures="unique(this) in alive")
698     ToTask() { }
699
700     @Perm(requires="pure(this) in alive",
701     ensures="pure(this) in alive")
702     public String getheader() {
703         return null;
704     }
705     @Perm(requires="full(this) in alive",
706     ensures="full(this) in alive")
707     public void setheader(String header) {
708     }
709     @Perm(requires="pure(this) in alive",
710     ensures="pure(this) in alive")
711     public long getrandomSeed() {
712         return 0;
713     }
714     @Perm(requires="full(this) in alive",
715     ensures="full(this) in alive")
716     public void setrandomSeed(long randomSeed) {
717     }
718
719 }ENDOFCLASS
720
721 @ClassStates({@State(name = "alive")})
722
723 class Utilities {

```



```
724 @Perm(ensures="unique(this) in alive")
725 Utilities() { }

727 @Perm(requires="pure(this) in alive",
728 ensures="pure(this) in alive")
729 String which(String executable, String pathEnv) {
730     return null;
731 }
732 @Perm(requires="pure(this) in alive",
733 ensures="pure(this) in alive")
734 String[] splitString(String splitChar, String arg) {
735     return null;
736 }

738 String joinString(String joinChar, String stringArray[]) {
739     return null;
740 }

742 }ENDOFCLASS
```