

ExamManager Tests

Class Scope Testing

Modal Class Test:

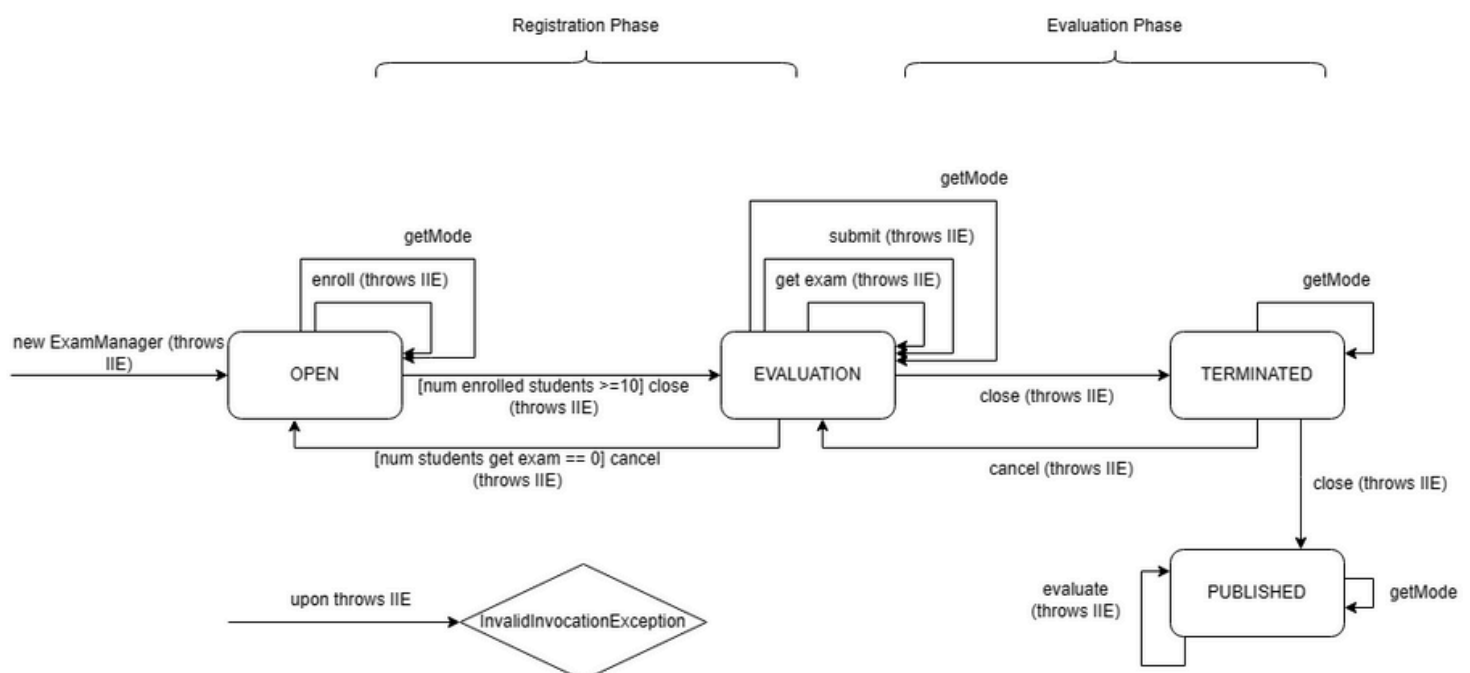
This pattern involves state-based testing, which aligns well with the behavior of **ExamManager** as it transitions through different modes (e.g., open, evaluation, published, terminated). State-based testing focuses on testing the behavior of a system or class based on its current state and the transitions it undergoes between states.

- The ExamManager class has fixed constraints on the sequence of messages/actions, such as enrolling students, closing exams, and publishing results.
- It transitions between different states (OPEN, EVALUATION, PUBLISHED, TERMINATED) based on specific conditions and message sequences, similar to the example of an Account object not accepting certain messages based on its balance or state.

Therefore, the ExamManager class aligns well with the Modal Class Test pattern as it involves testing the behavior of a class with fixed constraints on message sequence and interactions between message sequences and state, which are crucial aspects covered by the Modal Class Test pattern.

Modal Class Test Strategy usage:

1 - Generating the state model for CUT (ExamManager):

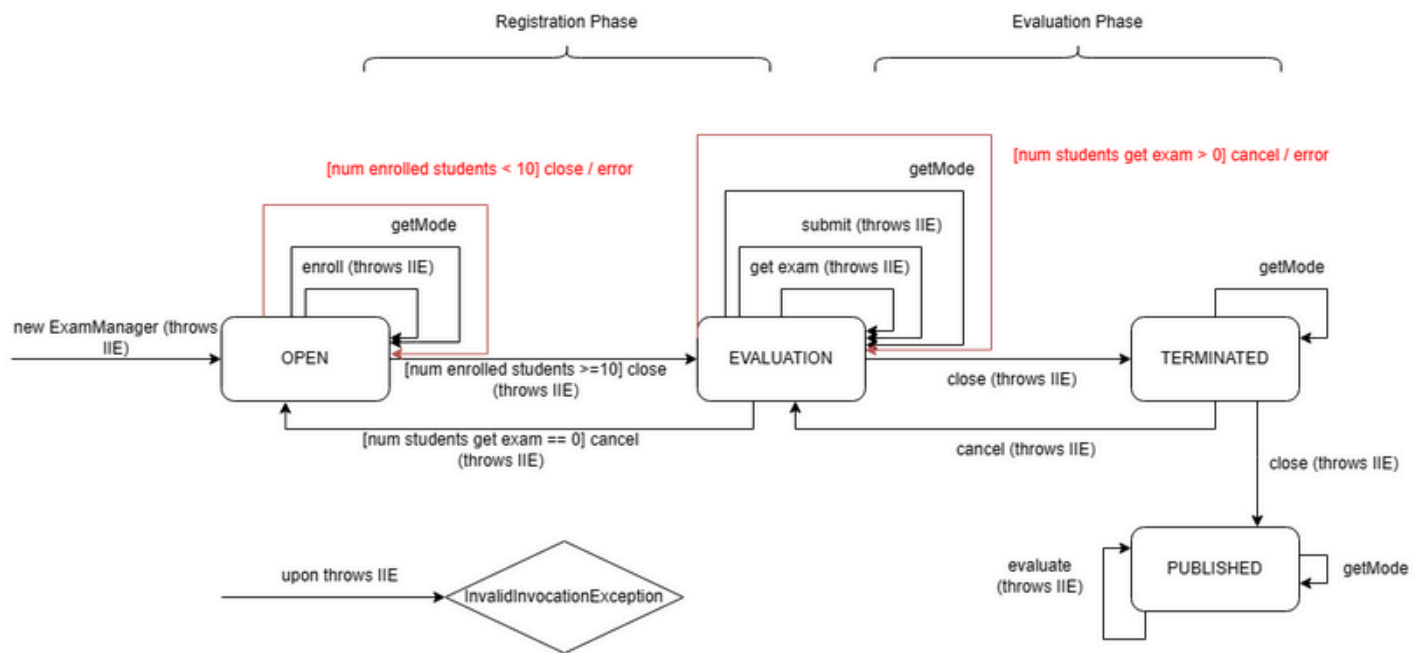


2 - Full expansion of conditional transition variants:

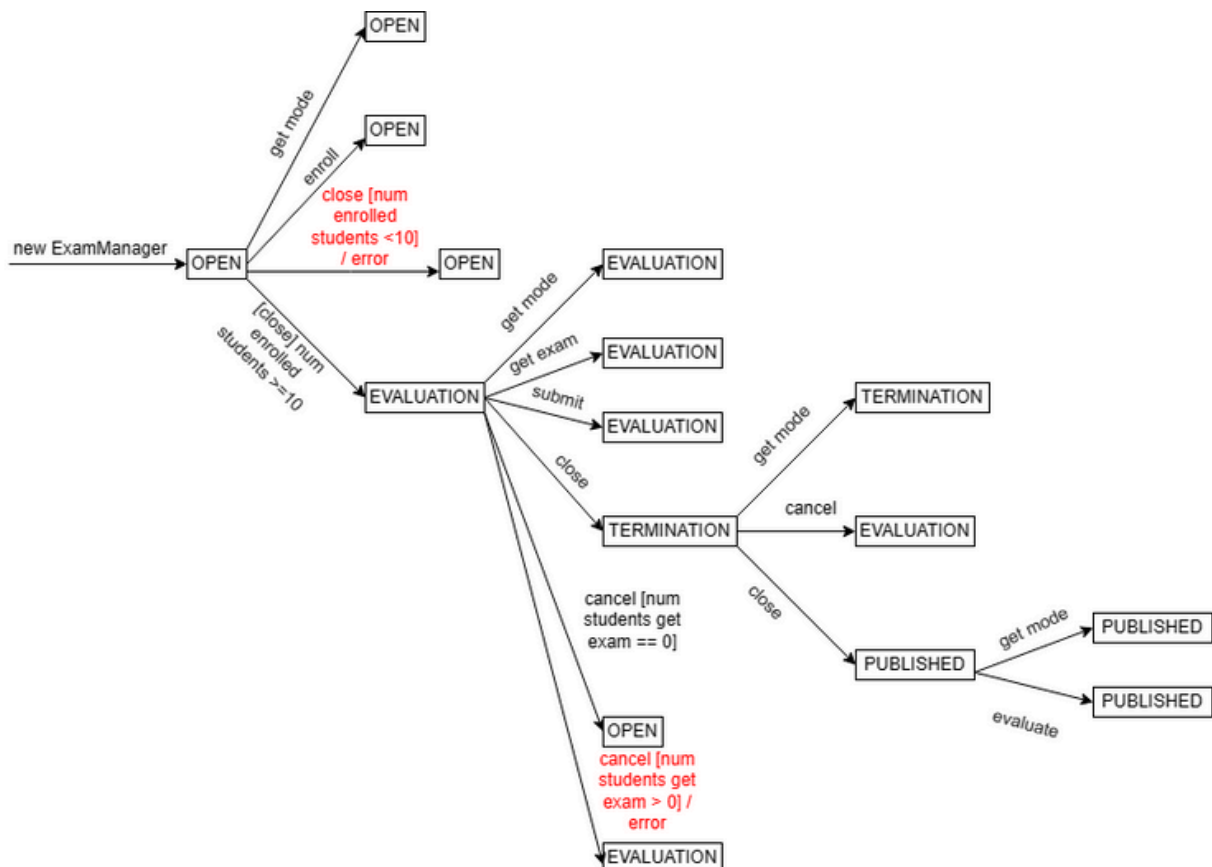
State	Message	Condition	Next State
OPEN	close ✗	POST: num enrolled students >=10	EVALUATION
EVALUATION	cancel ✗	PRE: num students get exam == 0	OPEN

close() in OPEN generates an additional transition when 'num enrolled students' < 10.
cancel() in EVALUATION generates an additional transition when 'num students get exam' > 0.

Updated state diagram with the additional transitions:



3 - Generate transition tree:



4 - Generate Conformance Test Suite (Tabulate events and actions along each path to form message sequences):

Run	Test Run/Event Path	Test Run/Event Path	Test Run/Event Path	Test Run/Event Path	Test Run/Event Path	Expected Terminal State	Exception
	<i>Level 1</i>	<i>Level 2</i>	<i>Level 3</i>	<i>Level 4</i>	<i>Level 5</i>		
1	new					OPEN	-
2	new	get mode				OPEN	-
3	new	enroll				OPEN	-
4	new	close [num enrolled students <10] / error				OPEN	IIE
5	new	close [num enrolled students >=10]				EVALUATION	-
6	new	close [num enrolled students >=10]	cancel [num students get exam > 0] / error			EVALUATION	IIE
7	new	close [num enrolled students >=10]	get mode			EVALUATION	-
8	new	close [num enrolled students >=10]	get exam			EVALUATION	-
9	new	close [num enrolled students >=10]	submit			EVALUATION	-
10	new	close [num enrolled students >=10]	num students get exam == 0			OPEN	-
11	new	close [num enrolled students >=10]	close			TERMINATION	-
12	new	close [num enrolled students >=10]	close	get mode		TERMINATION	-
13	new	close [num enrolled students >=10]	close	cancel		EVALUATION	-
14	new	close [num enrolled students >=10]	close	close		PUBLISHED	-
15	new	close [num enrolled students >=10]	close	close	get mode	PUBLISHED	-
16	new	close [num enrolled students >=10]	close	close	evaluate	PUBLISHED	-

5 - Develop test data for each path using Invariant Boundaries pattern for events, messages, and actions:

OPEN	Condition	ON	OFF
num enrolled students	<10	10 ✓	9 (repeated)
	>=10	10 (repeated)	9 ✓

EVALUATION	Condition	ON	OFF
num students get exam	>0	0 (repeated)	1 ✓
	==0	0 ✓	1 (repeated), -1 (impossible)

7 - Develop a sneak path test suite:

Events	States	States	States	States
	<i>OPEN</i>	<i>EVALUATION</i>	<i>TERMINATION</i>	<i>PUBLISHED</i>
enroll	✓	PSP	PSP	PSP
close	✓	✓	✓	PSP
cancel	PSP	✓	✓	PSP
get exam	PSP	✓	PSP	PSP
submit	PSP	✓	PSP	PSP
evaluate	PSP	PSP	PSP	✓

7b - Develop a sneak path test suite:

One test case per PSP:

Run	Test Run/Event Path	Test Run/Event Path	Test Run/Event Path	Test Run/Event Path	Test Run/Event Path	Expected Terminal State	Exception
	Level 1	Level 2	Level 3	Level 4	Level 5		
17	new	cancel				OPEN	IIE
18	new	get exam				OPEN	IIE
19	new	submit				OPEN	IIE
20	new	evaluate				OPEN	IIE
21	new	close	enroll			EVALUATION	IIE
22	new	close	evaluate			EVALUATION	IIE
23	new	close	close	enroll		TERMINATION	IIE
24	new	close	close	get exam		TERMINATION	IIE
25	new	close	close	submit		TERMINATION	IIE
26	new	close	close	evaluate		TERMINATION	IIE
27	new	close	close	close	enroll	PUBLISHED	IIE
28	new	close	close	close	close	PUBLISHED	IIE
29	new	close	close	close	cancel	PUBLISHED	IIE
30	new	close	close	close	get exam	PUBLISHED	IIE
31	new	close	close	close	submit	PUBLISHED	IIE