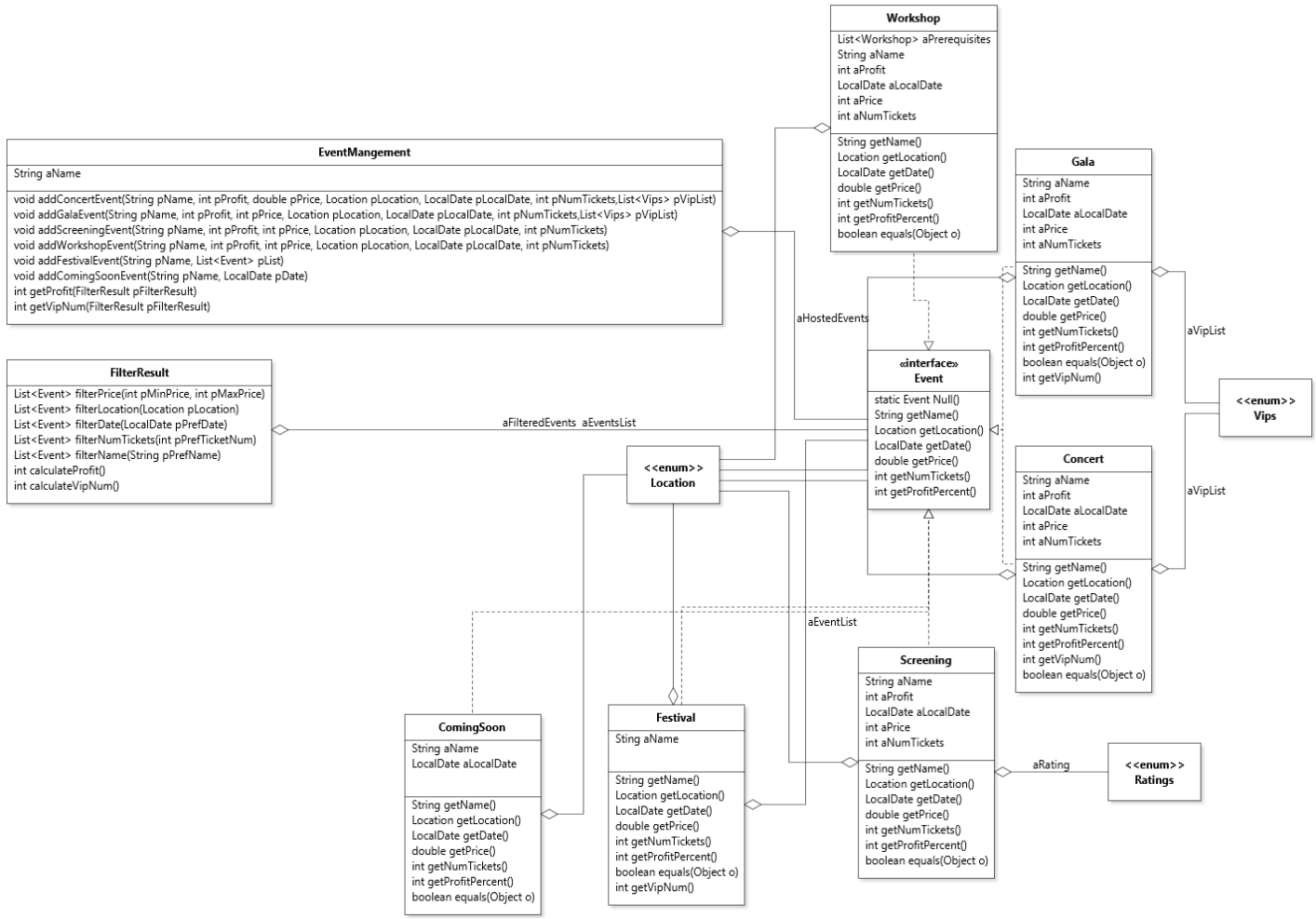
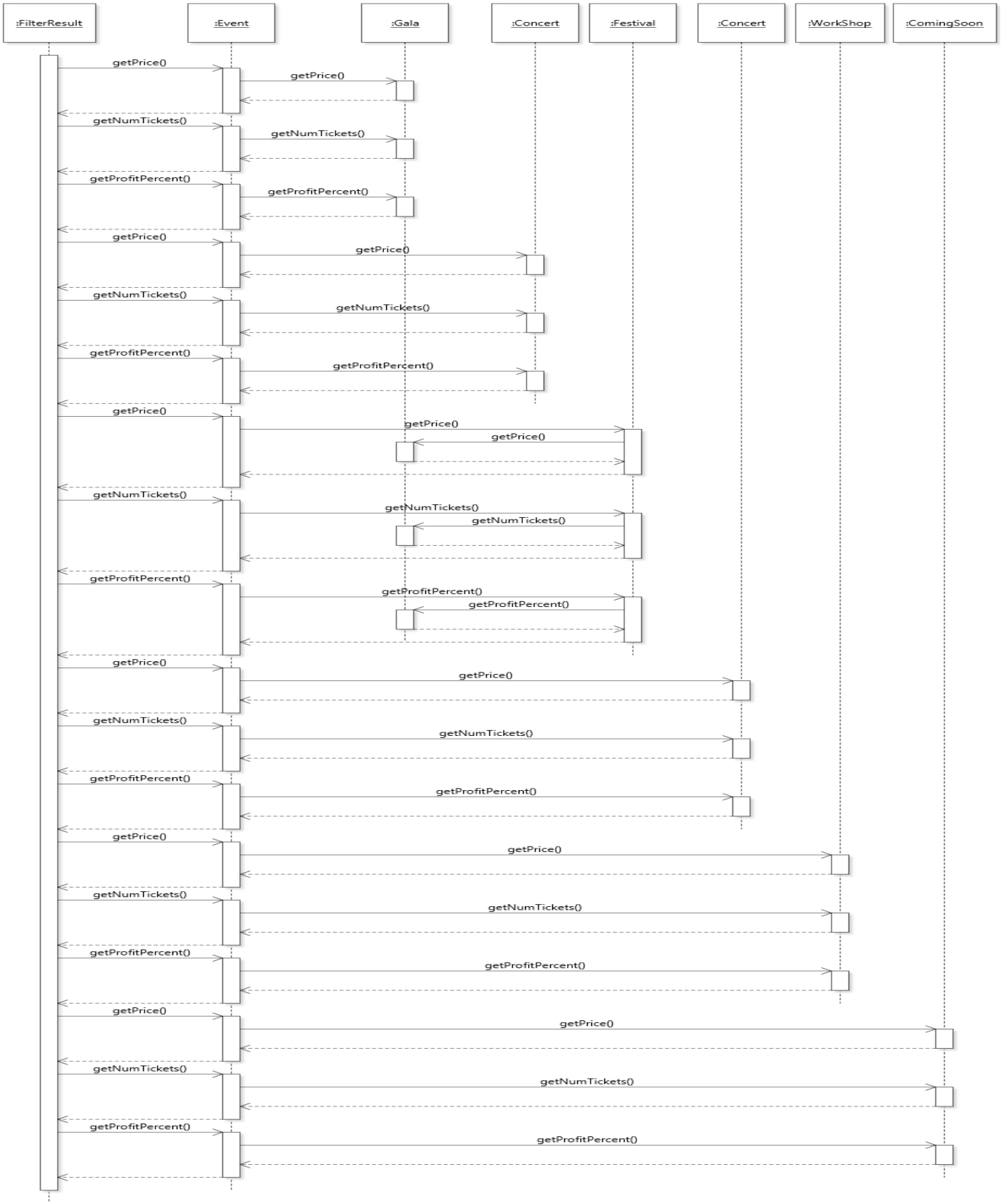
1) Created 4 requested classes. They implement Event and override methods from Event. Accessors for requested fields are included in the classes. Since Bob won't call any of the Event classes, I assumed the fields wouldn't be altered after initialization. So, location, name, date and ticket price are immutable. Ratings and Vips are implemented as enums.
2) I used the Composite design pattern to create a festival. It implements Event and overrides methods from event. By using for loops and if statements it follows the requirements specified in the PDF.
3) I created filter methods for each type of criteria Bob might want to filter the Events with. FilterResult takes an event list and when any of the filter methods is called it will return a FilterResult object which contains the aFilteredList. While filtering I made sure that duplicate events wouldn't be added to filtered list. I added other filtering methods so that Bob can add any other criteria he wants to his filtering operation,
4) CalculateProfit() method in FilterResult will retrieve the price, ticket number and profit percentage from events of the filtered list and multiply these values to calculate the expected value of FilterResult. I added getProfitPercentage() to Event interface. This way when bob is creating events through Eventmanagement he will be able to enter percentages for the event he is creating, and the percentage will be passed as argument and retrieved by getProfitPercentage() when it's called.CalculateVipNum() method will call getVipNum() if the event is an instance of Gala,Concert or Festival which will call getVipNum() again for the events it contains and if they are also instances of Gala and Concert, it will retrieve Vip nums. getVipNum() in Gala and Concert returns the number of Elements in the list with List.size().
5) I used Null object design pattern to avoid ComingSoon.getLocation() from breaking the execution of the program. I added static Event Null inside Event interface which returns null and 0 according to the type. When ComingSoon.getLocation() is called it won't throw NullPointerException. For other unknown fields I returned 0 since we don't know the values.
6) I added if statements to make sure no event with same date and location is created. It will print "Event with same date and location already exists" when Bob tries to create such events.
7) Every add Event method takes parameters it requires to create a new Event of the requested type and after the creation if it's not duplicate it will be added to aHostedEvents. I used Assert statements for input validation. As I demonstrated in Driver. Java Bob can create events and calculate expected profit of Filtered Result without calling any Event Class directly. I also added addFestivalEven, addComingSoonEvent and two calculate methods for profit and vip calculation to make it easy for Bob to use EventManagement class.
8) By tests cases are in the test folder the coverage is on the next page

Element	Class, %	Method, %	Line, %	Branch, %
com				
images				
java				
javax				
jdk				
META-INF				
netscape				
org				
sun				
toolbarButtonGraphics				
ComingSoon	100% (1/1)	33% (3/9)	60% (9/15)	0% (0/5)
Concert	100% (1/1)	73% (11/15)	84% (22/26)	25% (2/8)
Driver	0% (0/1)	0% (0/1)	0% (0/18)	100% (0/0)
Event	0% (0/2)	0% (0/7)	0% (0/7)	100% (0/0)
EventManagement	100% (1/1)	100% (11/11)	90% (38/42)	13% (4/30)
Festival	100% (1/1)	88% (8/9)	87% (35/40)	45% (5/11)
FilterResult	100% (1/1)	100% (9/9)	98% (59/60)	56% (14/25)
Gala	100% (1/1)	83% (10/12)	90% (20/22)	40% (2/5)
Location	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
Ratings	0% (0/1)	0% (0/1)	0% (0/2)	100% (0/0)
Screening	100% (1/1)	66% (8/12)	81% (18/22)	37% (3/8)
TestEvenManagement	100% (1/1)	100% (12/12)	100% (54/54)	100% (0/0)
TestFestival	100% (1/1)	100% (6/6)	100% (117/117)	100% (0/0)
TestFilterResult	100% (1/1)	100% (7/7)	100% (150/150)	100% (0/0)
Vips	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
Workshop	100% (1/1)	50% (6/12)	73% (17/23)	0% (0/8)

9)Here are my Class and Sequence diagrams:




:FilterResult
:Event
:Gala
:Concert
:Festival
:Concert
:WorkShop
:ComingSoon
getPrice()
getNumTickets()
getProfitPercent()