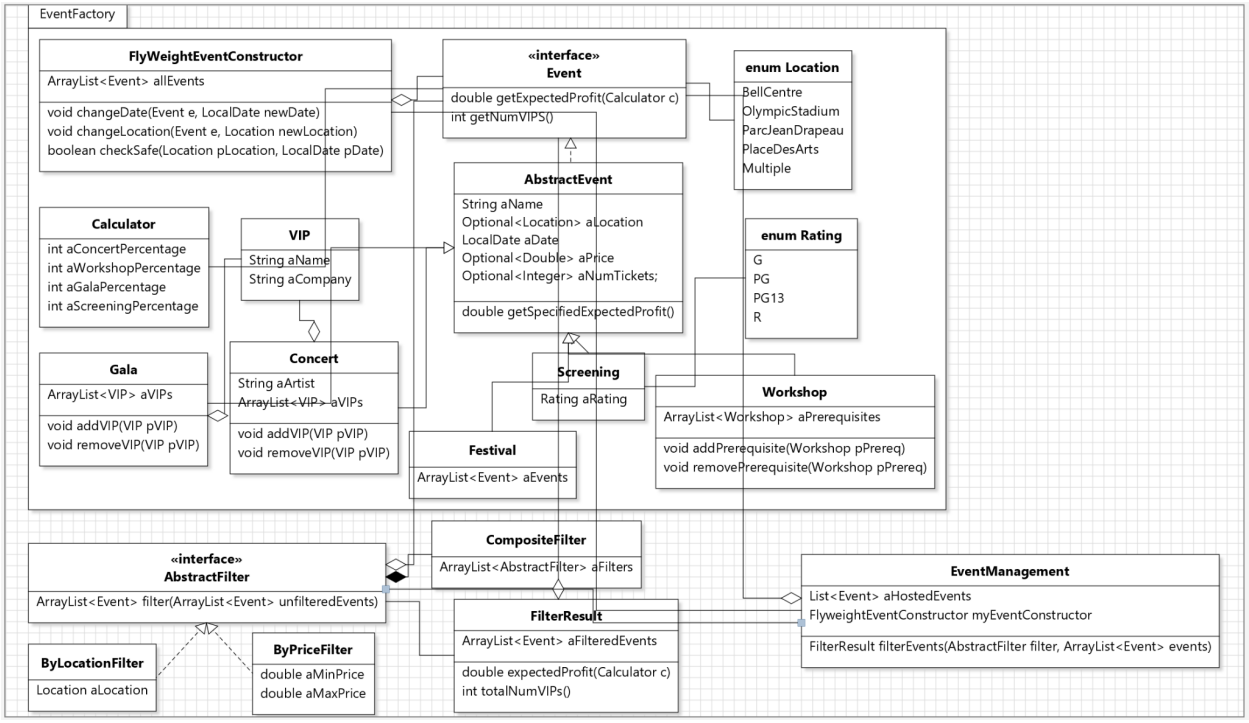
Tess Gompper 260947251
COMP303 Assignment 5

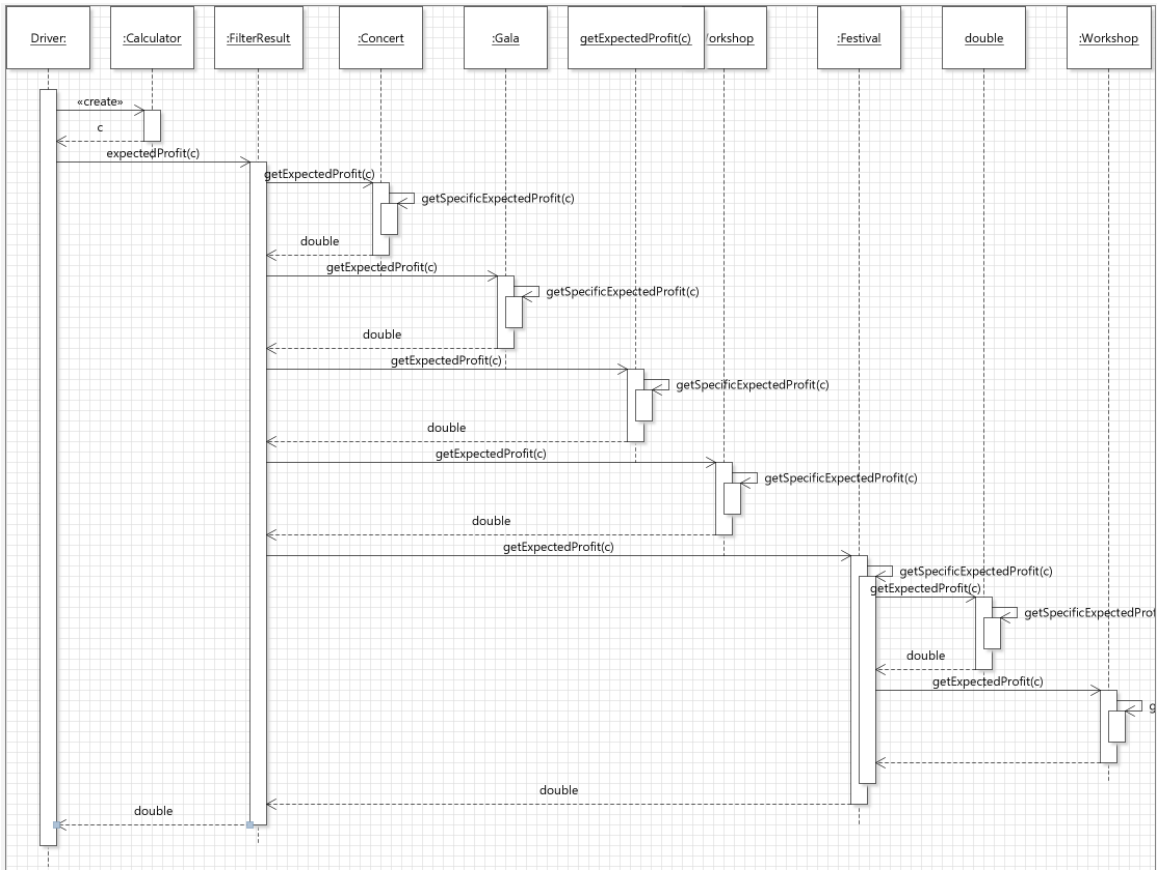
For this assignment I made a couple of design decisions. In order to restrict the creation of events on the same date at the same location I used the Flyweight pattern with my FlyweightEventConstructor and by changing the event constructors to protected. I used the Strategy pattern to allow for easy construction of new filters and I also used the Composite pattern to allow for combining filters. For my events I created an AbstractEvent class and used inheritance to avoid repeating code. I used the template pattern for calculating individual event profits to avoid repeating code. To allow for Coming Soon events that would not break execution of getters and setters, I used Optional types.

In my testing I used Stub events to test filters to avoid having to create many large events. I used reflection to test my private checkSafe() method which checks if a new event is allowed to be created. My branch and statement coverage is 100% for all classes except the test and stub classes.

See class and sequence diagrams below



Class diagram representing entire application



Sequence Diagram that demonstrates the calculation of revenue for the FilterResult containing at least one of each type of Event