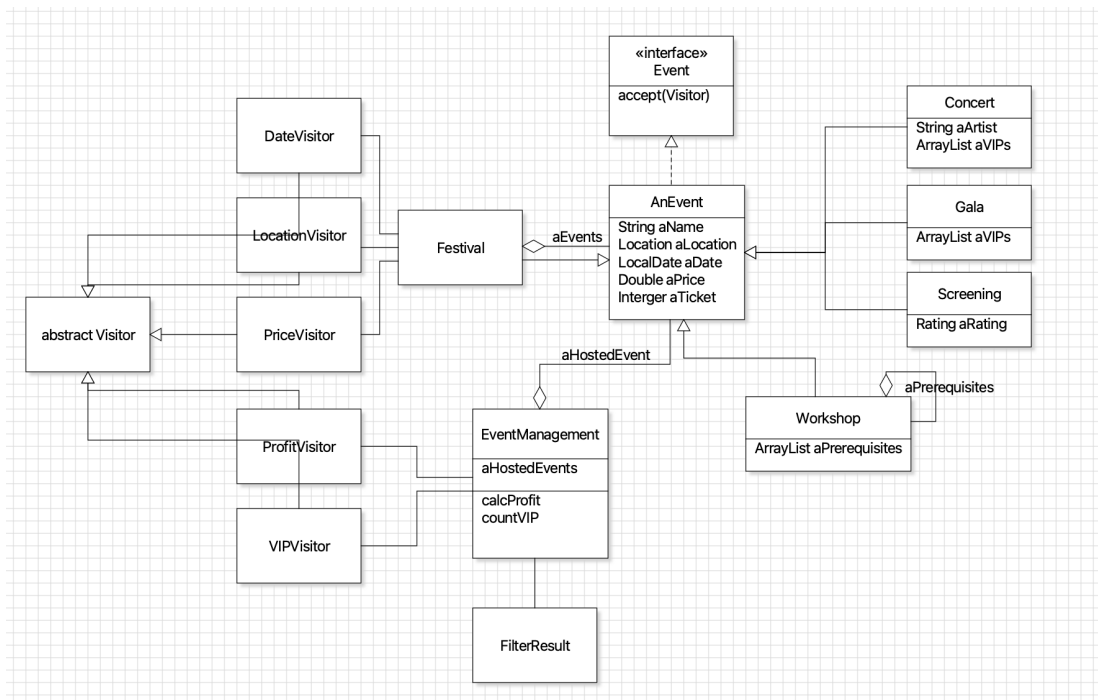
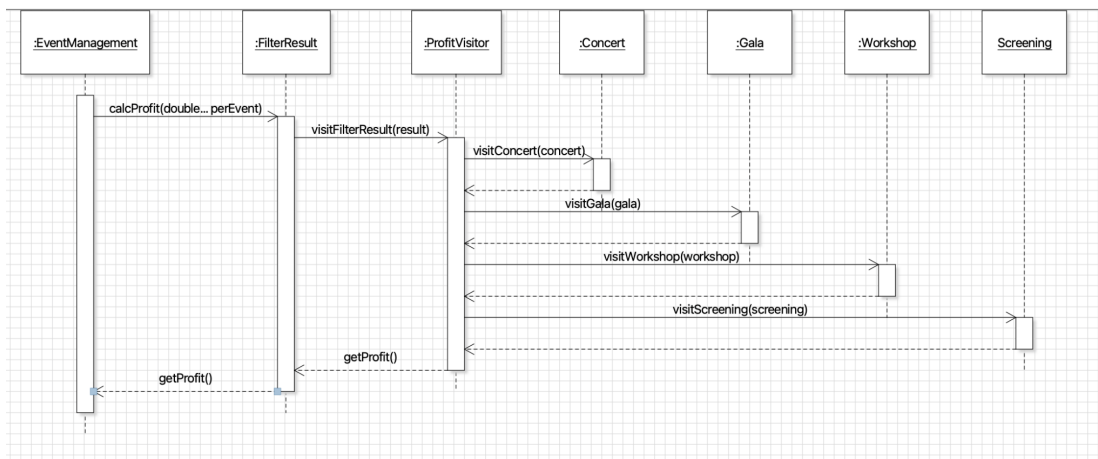




















1. Concert, Gala, Workshop, Screening and Festival are subclasses of AnEvent which implements the Event interface.
2. Visitor design pattern is used for setting fields for a Festival based on the corresponding fields of the events it contains.
3. Visitor design pattern is used for calculating profit and counting VIPs.
4. Lambda expressions with Predicate are used for filtering functionalities.
5. Coming Soon events are achieved by overloading the event constructors, with the missing field set to default values which can be determined later.



 AnEvent	100% (1/1)	76% (10/13)	78% (25/32)	33% (1/3)
 Concert	100% (1/1)	80% (4/5)	87% (7/8)	100% (0/0)
 DateVisitor	100% (1/1)	100% (5/5)	85% (17/20)	37% (3/8)
 Event	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
 EventManagement	100% (1/1)	100% (18/18)	100% (57/57)	69% (9/13)
 Festival	100% (1/1)	77% (7/9)	91% (21/23)	100% (0/0)
 FilterResult	100% (1/1)	100% (4/4)	100% (10/10)	100% (0/0)
 Gala	100% (1/1)	100% (4/4)	100% (6/6)	100% (0/0)
 Location	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
 LocationVisitor	100% (1/1)	83% (5/6)	73% (19/26)	44% (4/9)
 PriceVisitor	100% (1/1)	83% (5/6)	84% (11/13)	100% (4/4)
 ProfitVisitor	100% (1/1)	83% (5/6)	90% (10/11)	100% (0/0)
 Rating	100% (1/1)	100% (3/3)	100% (7/7)	100% (0/0)
 Screening	100% (1/1)	75% (3/4)	87% (7/8)	50% (1/2)
 TicketVisitor	100% (1/1)	83% (5/6)	86% (13/15)	100% (5/5)
 VIP	100% (1/1)	100% (0/0)	100% (1/1)	100% (0/0)
 VIPVisitor	100% (1/1)	100% (3/3)	100% (4/4)	100% (0/0)
 Visitor	100% (1/1)	100% (2/2)	100% (7/7)	100% (2/2)

Line/Branch Coverage