EVENTBRITE PROJECT DESCRIPTION

We must first consider the ways to build the Festival, that can be considered like an Event, composed of other events. We modify the Event interface to have an accept function for a Visitor design pattern.

The Visitor design pattern allows us to determine aggregate information on a List of Events, such as the number of VIPs, the expected profits, or determine the updates on events done by the User. This design pattern allows for us to interact with the concrete classes of Events depending on their specificities.

We use a Singleton to contain the information pertaining to the computation of the profits and revenue share, in the whole package. This allows us to keep the profit computation coherent throughout the application.

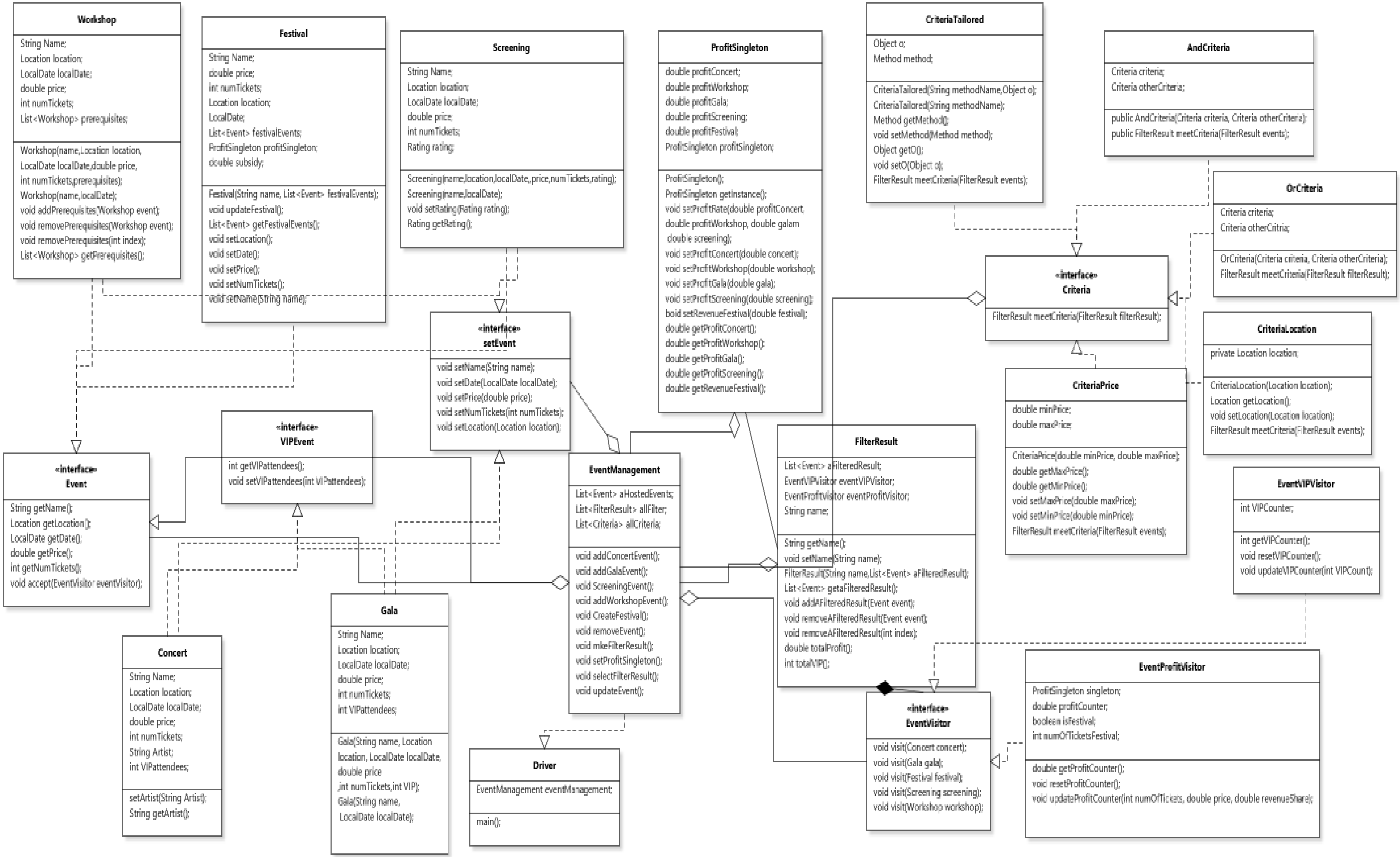
We use a Filter design pattern with the FilterResult class, in order to build criteria classes returning new FilterResult objects. The Filter design pattern allows returning tailored FilterResult objects.

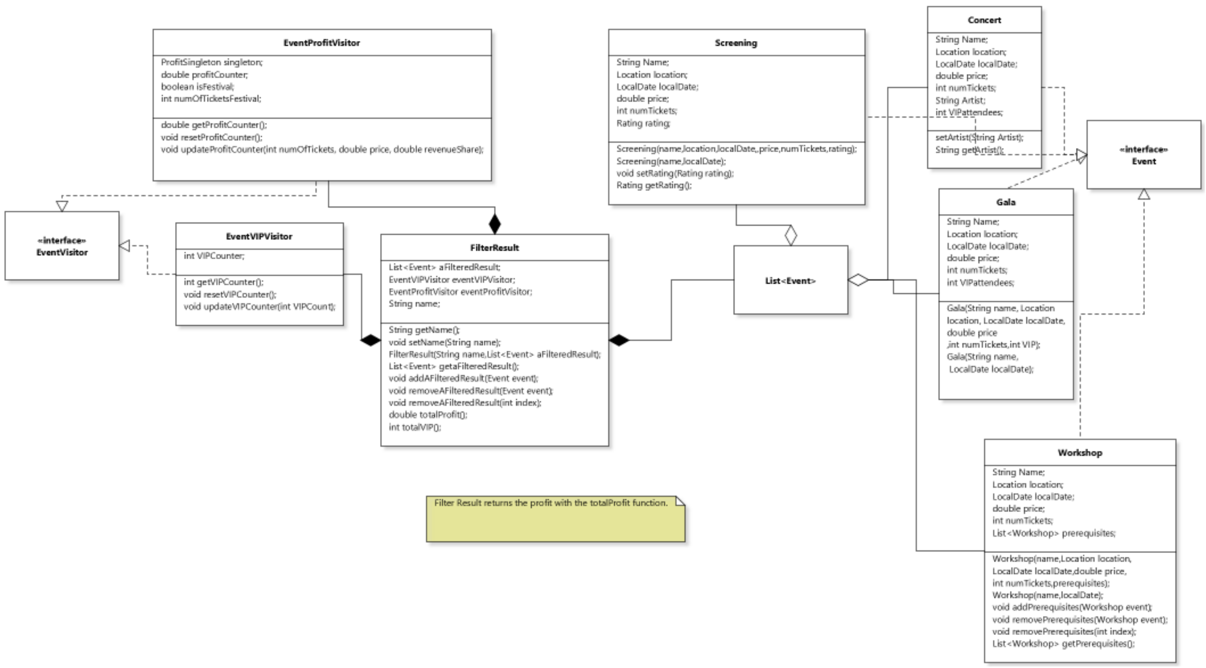
We use loops in order to contain the different functions used by Bob, directly, as there are restrictions on the functions that he may use. Those loops are limited by break and return keywords, to facilitate the usage of the application.

We built Tests for the Filter, Singleton, Visitor design patterns and other requirements. The Driver java class allows us to demonstrate the usage of the EventManagement class. We use Reflections to invoke the getter and setter method in the EventManagement class, as well as in the Tailored Criteria class, so that we can parse through all Event objects.

References:

1. https://www.tutorialspoint.com/design_pattern/visitor_pattern.htm
2. https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm
3. https://www.tutorialspoint.com/design_pattern/filter_pattern.htm
4. https://stackoverflow.com/questions/14413581/how-to-print-all-enum-values-in-java
5. https://www.tutorialspoint.com/design_pattern/strategy_pattern.htm
6. https://stackoverflow.com/questions/6185808/get-all-methods-of-an-interface-or-abstract-class-using-reflection


Workshop
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
List<Workshop> prerequisites;
Workshop(name,Location location,
LocalDate localDate,double price,
int numTickets,prerequisites);
Workshop(name,localDate);
void addPrerequisites(Workshop event);
void removePrerequisites(Workshop event);
void removePrerequisites(int index);
List<Workshop> getPrerequisites();
Festival
String Name;
double price;
int numTickets;
Location location;
LocalDate;
List<Event> festivalEvents;
ProfitSingleton profitSingleton;
double subsidy;
Festival(String name, List<Event> festivalEvents);
void updateFestival();
List<Event> getFestivalEvents();
void setLocation();
void setDate();
void setPrice();
void setNumTickets();
void setName(String name);
Screening
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
Rating rating;
Screening(name,location,localDate,,price,numTickets,rating);
Screening(name,localDate);
void setRating(Rating rating);
Rating getRating();
ProfitSingleton
double profitConcert;
double profitWorkshop;
double profitGala;
double profitScreening;
double profitFestival;
ProfitSingleton profitSingleton;
ProfitSingleton();
ProfitSingleton getInstance();
void setProfitRate(double profitConcert,
double profitWorkshop, double galam
double screening);
void setProfitConcert(double concert);
void setProfitWorkshop(double workshop);
void setProfitGala(double gala);
void setProfitScreening(double screening);
boid setRevenueFestival(double festival);
double getProfitConcert();
double getProfitWorkshop();
double getProfitGala();
double getProfitScreening();
double getRevenueFestival();
CriteriaTailored
Object o;
Method method;
CriteriaTailored(String methodName,Object o);
CriteriaTailored(String methodName);
Method getMethod();
void setMethod(Method method);
Object getO();
void setO(Object o);
FilterResult meetCriteria(FilterResult events);
AndCriteria
Criteria criteria;
Criteria otherCriteria;
public AndCriteria(Criteria criteria, Criteria otherCriteria);
public FilterResult meetCriteria(FilterResult events);
OrCriteria
Criteria criteria;
Criteria otherCritria;
OrCriteria(Criteria criteria, Criteria otherCriteria);
FilterResult meetCriteria(FilterResult filterResult);
«interface»
Criteria
FilterResult meetCriteria(FilterResult filterResult);
«interface»
setEvent
void setName(String name);
void setDate(LocalDate localDate);
void setPrice(double price);
void setNumTickets(int numTickets);
void setLocation(Location location);
CriteriaLocation
private Location location;
CriteriaLocation(Location location);
Location getLocation();
void setLocation(Location location);
FilterResult meetCriteria(FilterResult events);
CriteriaPrice
double minPrice;
double maxPrice;
CriteriaPrice(double minPrice, double maxPrice);
double getMaxPrice();
double getMinPrice();
void setMaxPrice(double maxPrice);
void setMinPrice(double minPrice);
FilterResult meetCriteria(FilterResult events);
«interface»
VIPEvent
int getVIPattendees();
void setVIPattendees(int VIPattendees);
FilterResult
List<Event> aFilteredResult;
EventVIPVisitor eventVIPVisitor;
EventProfitVisitor eventProfitVisitor;
String name;
String getName();
void setName(String name);
FilterResult(String name,List<Event> aFilteredResult);
List<Event> getaFilteredResult();
void addAFilteredResult(Event event);
void removeAFilteredResult(Event event);
void removeAFilteredResult(int index);
double totalProfit();
int totalVIP();
«interface»
Event
String getName();
Location getLocation();
LocalDate getDate();
double getPrice();
int getNumTickets();
void accept(EventVisitor eventVisitor);
EventManagement
List<Event> aHostedEvents;
List<FilterResult> allFilter;
List<Criteria> allCriteria;
void addConcertEvent();
void addGalaEvent();
void ScreeningEvent();
void addWorkshopEvent();
void CreateFestival();
void removeEvent();
void mkeFilterResult();
void setProfitSingleton();
void selectFilterResult();
void updateEvent();
EventVIPVisitor
int VIPCounter;
int getVIPCounter();
void resetVIPCounter();
void updateVIPCounter(int VIPCount);
Concert
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
String Artist;
int VIPattendees;
setArtist(String Artist);
String getArtist();
Gala
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
int VIPattendees;
Gala(String name, Location
location, LocalDate localDate,
double price
,int numTickets,int VIP);
Gala(String name,
LocalDate localDate);
EventProfitVisitor
ProfitSingleton singleton;
double profitCounter;
boolean isFestival;
int numOfTicketsFestival;
double getProfitCounter();
void resetProfitCounter();
void updateProfitCounter(int numOfTickets, double price, double revenueShare);
«interface»
EventVisitor
void visit(Concert concert);
void visit(Gala gala);
void visit(Festival festival);
void visit(Screening screening);
void visit(Workshop workshop);
Driver
EventManagement eventManagement;
main();


EventProfitVisitor
ProfitSingleton singleton;
double profitCounter;
boolean isFestival;
int numOfTicketsFestival;
double getProfitCounter();
void resetProfitCounter();
void updateProfitCounter(int numOfTickets, double price, double revenueShare);
«interface»
EventVisitor
EventVIPVisitor
int VIPCounter;
int getVIPCounter();
void resetVIPCounter();
void updateVIPCounter(int VIPCount);
FilterResult
List<Event> aFilteredResult;
EventVIPVisitor eventVIPVisitor;
EventProfitVisitor eventProfitVisitor;
String name;
String getName();
void setName(String name);
FilterResult(String name,List<Event> aFilteredResult);
List<Event> getaFilteredResult();
void addAFilteredResult(Event event);
void removeAFilteredResult(Event event);
void removeAFilteredResult(int index);
double totalProfit();
int totalVIP();
Screening
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
Rating rating;
Screening(name,location,localDate,,price,numTickets,rating);
Screening(name,localDate);
void setRating(Rating rating);
Rating getRating();
List<Event>
Concert
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
String Artist;
int VIPattendees;
setArtist(String Artist);
String getArtist();
«interface»
Event
Gala
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
int VIPattendees;
Gala(String name, Location
location, LocalDate localDate,
double price
,int numTickets,int VIP);
Gala(String name,
LocalDate localDate);
Workshop
String Name;
Location location;
LocalDate localDate;
double price;
int numTickets;
List<Workshop> prerequisites;
Workshop(name,Location location,
LocalDate localDate,double price,
int numTickets,prerequisites);
Workshop(name,localDate);
void addPrerequisites(Workshop event);
void removePrerequisites(Workshop event);
void removePrerequisites(int index);
List<Workshop> getPrerequisites();

Filter Result returns the profit with the totalProfit function.