HW3 Design

Event (Abstract Base Class)

Method	Description	Calls Other Methods
Event(int id, std::string event_name, double price, int available_tickets)	Initialize shared attributes present in all Events	None
virtual ~Event()	Virtual destructor to be used by all derived classes unless a specific destructor is needed	None
int getId() const	Returns Event Id	None
double getPrice() const	Returns Event price	None
virtual EventCategory getEventCategory() const = 0	Virtual function to return the specific EventCategory of derived classes	None
std::string getEventName() const	Returns the name of the Event	None
bool hasTickets(int qty) const	Determines if the Event has the amount of tickets requested still available	None
void setTicketStatus(TicketStatus status)	Modifies the ticket status of the Event	None
TicketStatus getTicketStatus() const	Returns the ticket status of the Event	None

Members

Members	Desc
int id_	Unique id
std::string event_name_	Name of the event
double price_	Price of the Event
int available_tickets_	Number of tickets available for purchase
TicketStatus ticket_status_	Status of ticket availablility

Concert (Abstract Derived Class)

Method	Description	Calls Other Methods
Concert(int id, std::string event_name, double price, int available_tickets)	Initializes shared attributes present in all Concert Events	Event(int id, std::string event_name, double price, int available_tickets)

Method	Description	Calls Other Methods
virtual MusicGenre getGenre() const = 0	Virtual function to be defined in derived classes to return the music genre of the ConcertEvent	None
EventCategory getEventCategory() const override	Overridden virtual function returns the EventCategory assigned to the Event	None

Members

Members	Desc	Inherited Members	Desc
EventCategory category_	Type of event	int id_	Unique id
		std::string event_name_	Name of the event
		double price_	Price of the Event
		int available_tickets_	Number of tickets available for purchase
		TicketStatus ticket_status_	Status of ticket availablility

Sport (Abstract Derived Class)

Method	Description	Calls Other Methods
Sport(int id, std::string event_name, double price, int available_tickets)	Initializes shared attributes present in all Sport Events	Event(int id, std::string event_name, double price, int available_tickets)
virtual SportType getSportType() const = 0	Virtual function implemented by derived classes to return the type of sport at the SportEvent	None
EventCategory getEventCategory() const override	Overridden virtual function returns the EventCategory assigned to the Event	None

Members

Members	Desc	Inherited Members	Desc
EventCategory category_	Type of event	int id_	Unique id
		std::string event_name_	Name of the event

Members	Desc	Inherited Members	Desc
		double price_	Price of the Event
		int available_tickets_	Number of tickets available for purchase
		TicketStatus ticket_status_	Status of ticket availablility

Theater (Abstract Derived Class)

Method	Description	Calls Other Methods
Theater(int id, std::string event_name, double price, int available_tickets)	Initializes shared attributes present in all Theater Events	Event(int id, std::string event_name, double price, int available_tickets)
EventCategory getEventCategory() const override	Overridden virtual function returns the EventCategory assigned to the Event	None
virtual TheaterGenre getTheaterGenre() const = 0	Virtual function implemented by derived classes to return the genre of TheaterEvent	None

Members

Members	Desc	Inherited Members	Desc
EventCategory category_	Type of event	int id_	Unique id
		std::string event_name_	Name of the event
		double price_	Price of the Event
		int available_tickets_	Number of tickets available for purchase
		TicketStatus ticket_status_	Status of ticket availablility

Convention (Abstract Derived Class)

Method	Description	Calls Other Methods
Convention(int id, std::string event_name, double price, int available_tickets)	Initializes shared attributes present in all Convention Events	Event(int id, std::string event_name, double price, int available_tickets)

Method	Description	Calls Other Methods
EventCategory getEventCategory() const override	Overridden virtual function returns the EventCategory assigned to the Event	None
virtual std::string getIndustryType() const = 0	Virtual function implemented by derived classes to return the IndustryType of the ConventionEvent	None

Members

Members	Desc	Inherited Members	Desc
EventCategory category_	Type of event	int id_	Unique id
		std::string event_name_	Name of the event
		double price_	Price of the Event
		int available_tickets_	Number of tickets available for purchase
		TicketStatus ticket_status_	Status of ticket availablility

Comedy (Abstract Derived Class)

Method	Description	Calls Other Methods
Comedy(int id, std::string event_name, double price, int available_tickets)	Initializes shared attributes present in all Comedy Events	Event(int id, std::string event_name, double price, int available_tickets)
EventCategory getEventCategory() const override	Overridden virtual function returns the EventCategory assigned to the Event	None
virtual std::string getPerformer() const = 0	Virtual function to be implemented in derived classes to return the performer of the ComedyEvent	

Members

Members	Desc	Inherited Members	Desc
EventCategory category_	Type of event	int id_	Unique id
		std::string event_name_	Name of the event

Members	Desc	Inherited Members	Desc
		double price_	Price of the Event
		int available_tickets_	Number of tickets available for purchase
		TicketStatus ticket_status_	Status of ticket availablility

ConcertEvent (Concrete Derived Class)

Method	Description	Calls Other Methods
ConcertEvent(int id, std::string event_name, double price, int available_tickets, std::vector<std::string> artists, MusicGenre genre, DateTime event_date)	Initializes ConcertEvent specific attributes and calls Concert constructor for shared attribute handling	Concert(id, event_name, price, available_tickets)
MusicGenre getGenre() const override	Overridden virtual function returns the genre of the ConcertEvent	None
std::vector<std::string> getArtists() const	Returns a list of the Artists at the ConcertEvent	None
void setArtists(std::vector<std::string> artists)	Modifies the artists at the ConcertEvent	None
DateTime getDate() const	Returns the DateTime info of the ConcertEvent	None
void setDate(DateTime date)	Modifies the DateTime info of the ConcertEvent	None

Members

Members	Desc	Inherited Members	Desc
std::vector<std::string> artists_	List of artists	int id_	Unique id
DateTime event_date_	Stores date and time info of event	std::string event_name_	Name of the event
MusicGenre genre_	Genre of music at the event	double price_	Price of the Event
		int available_tickets_	Number of tickets available for purchase

Members	Desc	Inherited Members	Desc
		TicketStatus ticket_status_	Status of ticket availablility
		EventCategory category_	Type of event

SportEvent (Concrete Derived Class)

Method	Description	Calls Other Methods
SportEvent(int id, std::string event_name, double price, int available_tickets, DateTime event_date, std::string away_team, std::string home_team, SportType sport_type)	Initializes SportEvent specific attributes and calls Sport constructor for shared attribute handling	Sport(id, event_name, price, available_tickets)
SportType getSportType() const override	Overridden function returns the SportType of the Sport Event	None
std::string getAwayTeam() const	Returns the away team	None
void setAwayTeam(std::string away_team)	Modifies the away team	None
std::string getHomeTeam() const	Returns the home team	None
void setHomeTeam(std::string home_team)	Modifies the home team	None
DateTime getDate() const	Returns the DateTime info of the SportEvent	None
void setDate(DateTime date)	Modifies the DateTime info of the SportEvent	None

Members

Members	Desc	Inherited Members	Desc
std::string away_team_	Away team at event	int id_	Unique id
std::string home_team_	Home team at event	std::string event_name_	Name of the event
SportType sport_type_	Type of sport being played	double price_	Price of the Event
DateTime event_date_	Stores date and time info of event	int available_tickets_	Number of tickets available for purchase

Members	Desc	Inherited Members	Desc
		TicketStatus ticket_status_	Status of ticket availablility
		EventCategory category_	Type of event

TheaterEvent (Concrete Derived Class)

Method	Description	Calls Other Methods
TheaterEvent(int id, std::string event_name, double price, int available_tickets, std::string original_title, std::string director, std::vector<std::string> performers, TheaterGenre genre, bool age_restricted, DateTime date)	Initializes TheaterEvent specific attributes and calls Theater constructor for shared attribute handling	Theater(id, event_name, price, available_tickets)
TheaterGenre getTheaterGenre() const override	Overridden function returns the genre of the TheaterEvent	None
std::vector<std::string> getPerformers() const	Returns a list of the performers	None
void setPerformers(std::vector<std::string> performers)	Modifies performers	None
std::string getOriginalTitle() const	Returns the title of the work being performed	None
void setOriginalTitle(std::string title)	Modifies the name of the work being performed	None
bool isAgeRestricted() const	Returns if the TheaterEvent is age restricted	None
void setAgeRestricted(bool is_restricted)	Modifies the age restriction	None
DateTime getDate() const	Returns the DateTime info of the TheaterEvent	None
void setDate(DateTime date)	Modifies the DateTime info of the TheaterEvent	None

Members

Members	Desc	Inherited Members	Desc

Members	Desc	Inherited Members	Desc
std::string original_title_	Title of the play	int id_	Unique id
std::string director_	Director of the production	std::string event_name_	Name of the event
std::vector/std::string performers_	List of performers	double price_	Price of the Event
TheaterGenre genre_	Type of theater performance	int available_tickets_	Number of tickets available for purchase
bool age_restricted_	Age restricted status of performance	TicketStatus ticket_status_	Status of ticket availablility
DateTime event_date_	Stores date and time info of event	EventCategory category_	Type of event

ConventionEvent (Concrete Derived Class)

Method	Description	Calls Other Methods
ConventionEvent(int id, std::string event_name, double price, int available_tickets, std::string industry_type, int num_exhibitors, std::vector<std::string> exhibitors,std::vector<std::string> sponsors, int num_days, DateTime date)	Initializes ConventionEvent specific attributes and calls Convention constructor for shared attribute handling	Convention(id, event_name, price, available_tickets)
std::string getIndustryType() const override	Overridden function returns IndustryType of the ConventionEvent	None
int getNumExhibitors() const	Returns the number of Exhibitors	None
void setNumExhibitors(int num_exhibitors)	Modifies the number of Exhibitors	None
std::vector<std::string> getExhibitors() const	Returns a list of the Exhibitors	None
void setExhibitors(std::vector<std::string> exhibitors)	Modifies the Exhibitors at the ConventionEvent	None
std::vector<std::string> getSponsors() const	Returns a list of the Sponsors	None

Method	Description	Calls Other Methods
void setSponsors(std::vector<std::string> sponsors)	Modifies the Sponsors at the ConventionEvent	None
int getNumDays() const	Returns the number of days the ConventionEvent lasts	None
void setNumDays(int num_days)	Modifies the duration of the ConventionEvent	None
DateTime getDate() const	Returns the DateTime info of the ConventionEvent	None
void setDate(DateTime date)	Modifies the DateTime info of the ConventionEvent	None

Members

Members	Desc	Inherited Members	Desc
std::string industry_type_	Stores type of industry	int id_	Unique id
int num_exhibitors_	Stores the number of exhibitors	std::string event_name_	Name of the event
std::vector<std::string> exhibitors_	Stores names of exhibitors	double price_	Price of the Event
std::vector<std::string> sponsors_	Stores names of sponsors	int available_tickets_	Number of tickets available for purchase
int num_days_	Stores number of days the event lasts	TicketStatus ticket_status_	Status of ticket availablility
DateTime event_date_	Stores date and time info of event	EventCategory category_	Type of event

ComedyEvent (Concrete Derived Class)

Method	Description	Calls Other Methods

Method	Description	Calls Other Methods
ComedyEvent(int id, std::string event_name, double price, int available_tickets, std::string performer, bool age_restricted, std::vector<std::string> topics, DateTime date)	Initializes ComedyEvent specific attributes and calls Comedy constructor for shared attribute handling	Comedy(id, event_name, price, available_tickets)
std::string getPerformer() const override	Overridden function returns the performer of the ComedyEvent	None
bool getAgeRestricted() const	Returns age restricted status of the ComedyEvent	None
void setAgeRestricted(bool age_restricted)	Modifies the ComedyEvent	None
std::vector<std::string> getTopics() const	Returns a list of the topics covered at the ComedyEvent	None
void setTopics(std::vector<std::string> topics)	Modifies the topics	None
DateTime getDate() const	Returns the DateTime info of the ComedyEvent	None
void setDate(DateTime date)	Modifies the DateTime info of the ComedyEvent	None

Members

Members	Desc	Inherited Members	Desc
std::string performer_	Stores the name of the performer	int id_	Unique id
bool age_restricted_	Stores if the event is age restricted	std::string event_name_	Name of the event
std::vector<std::string> topics_	Stores the topics discussed	double price_	Price of the Event
DateTime event_date_	Stores date and time info of event	int available_tickets_	Number of tickets available for purchase
		TicketStatus ticket_status_	Status of ticket availablility
		EventCategory category_	Type of event

EventFactory - Singleton EventFactory Design

Method	Description	Calls Other Methods
EventFactory()	Prevents creation of of public default constructor	None
Event* createConcertEvent(int id, std::string event_name, double price, int available_tickets, std::vector<std::string> artists, MusicGenre genre, DateTime event_date)	instantiates a ConcertEvent stored in an Event pointer	Concert(id, event_name, price, available_tickets)
Event* createSportEvent(int id, std::string event_name, double price, int available_tickets, DateTime event_date, std::string away_team, std::string home_team, SportType sport_type)	instantiates a SportEvent stored in an Event pointer	Sport(id, event_name, price, available_tickets)
Event* createTheaterEvent(int id, std::string event_name, double price, int available_tickets, std::string original_title, std::string director, std::vector<std::string> performers, TheaterGenre genre, bool age_restricted, DateTime date)	instantiates a TheaterEvent stored in an Event pointer	Theater(id, event_name, price, available_tickets)
Event* createConventionEvent(int id, std::string event_name, double price, int available_tickets, std::string industry_type, int num_exhibitors, std::vector<std::string> exhibitors,std::vector<std::string> sponsors, int num_days, DateTime date)	instantiates a ConventionEvent stored in an Event pointer	Convention(id, event_name, price, available_tickets)
Event* createComedyEvent(int id, std::string event_name, double price, int available_tickets, std::string performer, bool age_restricted, std::vector<std::string> topics, DateTime date)	instantiates a ComedyEvent stored in an Event pointer	Comedy(id, event_name, price, available_tickets)
static EventFactory* getInstance()	Returns the EventFactory instance or creates one	EventFactory()
Event* createEvent(EventCategory category)	Responsible for calling the correct helper function to create a Event object	Event* createConcertEvent(), Event* createSportEvent(), Event* createTheaterEvent(), Event* createConventionEvent(), Event* createComedyEvent()

Members

Members	Desc
static EventFactory* instance_	Singleton instance of EventFactory

User (Abstract Base Class)

Method	Description	Calls Other Methods
User(int id, std::string name, double balance)	Initialize all values shared by Users	None
virtual std::string getRole() const = 0	Virtual fucntion to be defined by derived classes to return the role of the user	
int getId() const	Return the id of a user	None
std::string getName() const	Return user name	None
double getBalance() const	Return user balance	None
void setBalance(double new_balance)	Modifies the balance of a user	None
std::vector<Event*> getHistory() const	Returns the user history	None
void setHistory(std::vector<Event*> history)	Modifies the user history	None

Members

Members	Desc
id_	Store unique id
std::string name_	Store user name
double balance_	Store user balance
std::vector<Event*> history_	Store user history

Attendee (Concrete Derived Class)

Method	Description	Calls Other Methods
Attendee(int id, std::string name, double balance)	Uses User constuctor to initialize shared attributes	User(id, name, balance)
std::string getRole() const override	Returns the role of the user	None

Members

Inherited Members	Desc
id_	Store unique id
std::string name_	Store user name
double balance_	Store user balance
std::vector<Event*> history_	Store user history

Organizer (Concrete Derived Class)

Method	Description	Calls Other Methods
Organizer(int id, std::string name, double balance)	Uses User constuctor to initialize shared attributes	User(id, name, balance)
std::string getRole() const override	Returns the role of the user	None

Members

Inherited Members	Desc
id_	Store unique id
std::string name_	Store user name
double balance_	Store user balance
std::vector<Event*> history_	Store user history

EventManager - Singleton Design

Method	Description	Calls Other Methods
static EventManager* get_instance()	Returns or creates a new EventManager	EventManager()
EventManager()	Prevents the creation of a public default constructor	None
User* getUser(int id) const	Returns the user* associated with the id	None
void addUser(User* user)	Creates a new user and stores it in users_	Organizer(int id, std::string name, double balance), Attendee(int id, std::string name, double balance)

Method	Description	Calls Other Methods
void createEvent(User* organizer)	Creates a new event while prompting user for info	Event* createEvent(EventCategory category), std::vector<Event*> getHistory() const, void setHistory(std::vector<Event*> history)
Event* getEvent(int id) const	Returns an Event pointer matching the id	None
void moveToAvailable(int id)	Removes kvp from unavailable_events_ and adds it to available_events_	None
void moveToUnavailable(int id)	Removes kvp from available_events_ and adds to unavailable_events_	None
void printAvailableEvents() const	diplays all available events	None
void purchaseEvent(User* user, int event_id, int qty)	Purchases 1+ tickets to an event and manages side effects	std::vector<Event*> getHistory() const, void setHistory(std::vector<Event*> history), double getBalance() const, void setBalance(double new_balance), double getPrice() const, bool hasTickets(int qty) const, void setTicketStatus(TicketStatus status), TicketStatus getTicketStatus() const
void sellTicket(User* user, int event_id)	Sells one of the users tickets and handles side effects	std::vector<Event*> getHistory() const, void setHistory(std::vector<Event*> history), double getBalance() const, void setBalance(double new_balance), double getPrice() const, bool hasTickets(int qty) const, void setTicketStatus(TicketStatus status), TicketStatus getTicketStatus() const
void printBalance(User* user) const	Displays User balance	None
void printUserHistory(int id) const	Displays User history	None

Members

Members	Desc
std::unordered_map<int, User*> users_	Stores id, user pointer value pairs
std::unordered_map<int, Event*> available_events_	Stores id, pointer value pairs of available events
std::unordered_map<int, Event*> unavailable_events_	Stores id, pointer value pairs of unavailable events

Main() Psuedocode

```
int main (){
    eventID = 1
    userID = 1
    create instance of EventManager and EventFactory
    load data from user csv and store in event manager users
    load data from event csv and store in event manager events
    prompt user to identify as attendee or organizer
    prompt user to use an existing user or to make a new user
    fetch user id or create new user
    while not quit{
        if attendee {
            list options {
                list available events
                purchase event
                sell ticket
                see user history
                check balance
                switch users
                quit
            }
            switch input {
                list available events:
                    list events with available tickets
                purchase event:
                    get event price
                    add fee for event
                    verify balance
                    decrement balance
                    append attendee history
                    decrement event available tickets
                Sell ticket:
                    increment tickets available for event
                    remove event from user history
                    increment user balance
                See user history:
                    list user history
                check balance:
                    display balance
                switch users:
                    modify user state in main
                quit:
                    return 0
            }
        }
        if organizer {
            list options {
                create new event
                list history
                check balance
                switch users
                quit
            }
            switch input{
                create event:
                    call event factory to create new event
                        prompt for event details
                    add event to event manager and organizer history
                list history:
                    display history
                check balance:
                    display balance
                switch user:
                    modify user state in main
                quit:
                    return 0
            }
        }
    }
}
```