|  |
| --- |
| Ticket |
| -event: String  -quantity: int |
| +Ticket(event:String,quantity:int)  +getEvent(String)  +getQuantity(int)  *+PrintTicketDetails():void* |

|  |
| --- |
| TicketCounter |
| -ticketInventory:Map<String>  -purchesedTicket:List<Ticket> |
| +TicketCounter();  +CheckAvailability(event:String,quantity:int);  +PurchesedTicket(event:String,quantity:int);  +getPurchesedTickets():List<Ticket>; |

|  |
| --- |
| ConcertTicket |
|  |
|  |

|  |
| --- |
| SportsTicket |
|  |
|  |

|  |
| --- |
| MovieTicket |
|  |
|  |

|  |
| --- |
| <<TicketOperator>> |
|  |
| PurchaseTicket(event:String,quantity:int):int;void |

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

interface TicketOperations {

void purchaseTicket(String event, int quantity);

}

abstract class Ticket {

private String event;

private int quantity;

public Ticket(String event, int quantity) {

this.event = event;

this.quantity = quantity;

}

public String getEvent() {

return event;

}

public int getQuantity() {

return quantity;

}

public abstract void printTicketDetails();

}

class ConcertTicket extends Ticket {

public ConcertTicket(String event, int quantity) {

super(event, quantity);

}

@Override

public void printTicketDetails() {

System.out.println("Concert Ticket Details");

System.out.println("Event: " + getEvent());

System.out.println("Quantity: " + getQuantity());

System.out.println("Enjoy the concert!");

}

}

class SportsTicket extends Ticket {

public SportsTicket(String event, int quantity) {

super(event, quantity);

}

@Override

public void printTicketDetails() {

System.out.println("Sports Ticket Details");

System.out.println("Event: " + getEvent());

System.out.println("Quantity: " + getQuantity());

System.out.println("Enjoy the sports match!");

}

}

class MovieTicket extends Ticket {

public MovieTicket(String event, int quantity) {

super(event, quantity);

}

@Override

public void printTicketDetails() {

System.out.println("Movie Ticket Details");

System.out.println("Event: " + getEvent());

System.out.println("Quantity: " + getQuantity());

System.out.println("Enjoy the movie!");

}

}

class TicketCounter implements TicketOperations {

private Map<String, Integer> ticketInventory;

private List<Ticket> purchasedTickets;

public TicketCounter() {

ticketInventory = new HashMap<>();

ticketInventory.put("Concert", 100);

ticketInventory.put("Sports Match", 200);

ticketInventory.put("Movie", 150);

purchasedTickets = new ArrayList<>();

}

public boolean checkAvailability(String event, int quantity) {

if (ticketInventory.containsKey(event)) {

int availableTickets = ticketInventory.get(event);

return availableTickets >= quantity;

}

return false;

}

@Override

public void purchaseTicket(String event, int quantity) {

if (checkAvailability(event, quantity)) {

int availableTickets = ticketInventory.get(event);

availableTickets -= quantity;

ticketInventory.put(event, availableTickets);

Ticket purchasedTicket;

if (event.equalsIgnoreCase("Concert")) {

purchasedTicket = new ConcertTicket(event, quantity);

} else if (event.equalsIgnoreCase("Sports Match")) {

purchasedTicket = new SportsTicket(event, quantity);

} else {

purchasedTicket = new MovieTicket(event, quantity);

}

purchasedTickets.add(purchasedTicket);

System.out.println("Tickets purchased successfully!");

purchasedTicket.printTicketDetails();

} else {

System.out.println("Tickets not available!");

}

}

public List<Ticket> getPurchasedTickets() {

return purchasedTickets;

}

}

public class TicketCounterSystem {

public static void main(String[] args) {

TicketCounter ticketCounter = new TicketCounter();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter event name: ");

String event = scanner.nextLine();

System.out.print("Enter quantity: ");

int quantity = scanner.nextInt();

ticketCounter.purchaseTicket(event, quantity);

}

}