CINEMA

TICKETS

DR.MAHA ALTHOBAITI

Students Names:

WED SALEH ALJOUED > 44453690

SHADA HAMED ALNAFEEI > 44453057

SHIMA BANDER ALAZWARI > 44453109

ARYAM MASHAN ALOTAIBI > 44458436

SECTION:

2019

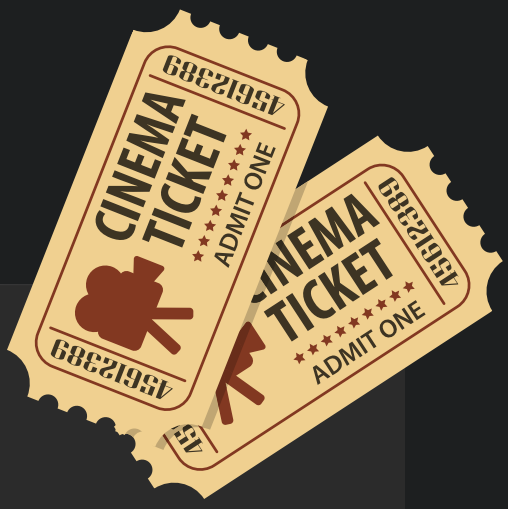
PROJECT DESCRIPTION:

This program is a Cinema Ticket Booking System that allows users to view a list of available movies, select a movie, and book either a Regular or VIP ticket. It features:

- Cinema Class: Manages a collection of movies, enabling adding and displaying available movies.
- Movie Class: Represents a movie with a title and price.
- Ticket Class (Abstract): Defines the structure for tickets with details like movie name, ticket type, and price.
- VIPTicket and RegularTicket Classes: Implement the Ticket class to handle pricing for VIP and regular tickets.
- Customer Class: Allows customers to book tickets by choosing a movie and ticket type (Regular or VIP).
- TestCinema Class: The main program where the cinema system is demonstrated, including adding movies, booking a ticket, and displaying ticket details.

This program provides a simple user interaction flow for booking tickets with flexible ticket type options.

CLASS: CINEMA




```java
package cinematickts;
//Importing ArrayList
import java.util.ArrayList;
import java.util.List;
public class Cinema {
    private List<Movie> movies; // The Cinema class has a list of Movie objects

    public Cinema() {
        movies = new ArrayList<>();
    }

    // Method to add a movie
    public void addMovie(Movie movie) {
        movies.add(movie);
    }

    // Method to display available movies
    public void displayMovies() {
        System.out.println("Available movies:");
        for (int i = 0; i < movies.size(); i++) {
            System.out.println((i + 1) + ". " + movies.get(i).getTitle());
        }
    }

    // Method to get the movie name by index
    public String getMovie(int index) {
        if (index >= 0 && index < movies.size()) {
            return movies.get(index).getTitle();
        } else {
            throw new IndexOutOfBoundsException("Invalid movie choice.");
        }
    }

    // Setter for movies list
    public void setMovies(List<Movie> movies) {
        this.movies = movies;
    }
}
```

The Cinema class represents a cinema that stores and manages a list of movies available for booking. It provides methods to add movies to the cinema, display the list of available movies, and retrieve the movie titles based on the index. Here's a breakdown of its components:

1. Attributes:
   - movies: A list of Movie objects, which stores the movies available in the cinema.
2. Constructor:
   - The constructor initializes the movies list as an empty ArrayList.
3. Methods:
   - addMovie(Movie movie): Adds a new Movie object to the cinema's list of movies.
   - displayMovies(): Prints a list of all available movies, displaying their index and title.
   - getMovie(int index): Returns the title of the movie at the specified index in the movies list. If the index is invalid, it throws an IndexOutOfBoundsException.
   - setMovies(List<Movie> movies): Sets the list of movies in the cinema to a new list of Movie objects.

Overall, the Cinema class serves as a container and manager for the movies in the ticket booking system.

CLASS: MOVIE

```java
package cinematickts;
public class Movie {
    private String title;
    private double price;

    // Constructor
    public Movie(String title, double price) {
        this.title = title;
        this.price = price;
    }

    // Getters
    public String getTitle() {
        return title;
    }

    public double getPrice() {
        return price;
    }

    // Setters
    public void setTitle(String title) {
        this.title = title;
    }

    public void setPrice(double price) {
        if (price >= 0) { // Ensure price is non-negative
            this.price = price;
        } else {
            System.out.println("Invalid price. Price cannot be negative.");
        }
    }
}
```

The Movie class represents a movie in the cinema system. It contains information about the movie's title and ticket price. The class provides methods to set and get the title and price of the movie.
Here's a breakdown of its components:

1. *Attributes:*
   - *title: A String representing the name of the movie.*
   - *price: A double representing the price of the movie ticket.*
2. *Constructor:*
   - *The constructor initializes a movie object with a given title and price.*
3. *Methods:*
   - *getTitle(): Returns the title of the movie.*
   - *getPrice(): Returns the price of the movie's ticket.*
   - *setTitle(String title): Sets a new title for the movie.*
   - *setPrice(double price): Sets a new price for the movie's ticket, ensuring that the price is non-negative. If a negative price is provided, it prints an error message.*

The Movie class is a basic representation of a movie, storing key details like the title and price, which are used when booking tickets.

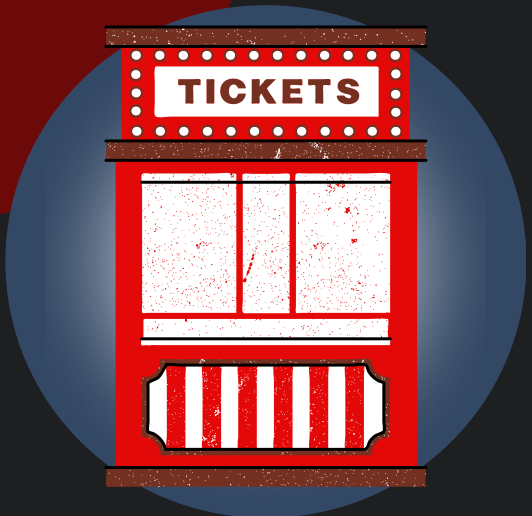
CLASS: TICKET

```java
package cinematickts;
public abstract class Ticket {
    protected String movieName;
    protected double price;
    protected String ticketType;

    // Constructor
    public Ticket(String movieName, String ticketType) {
        this.movieName = movieName;
        this.ticketType = ticketType;
    }

    // Abstract method for calculating price
    public abstract void calculatePrice();

    // Method to display ticket information
    public void displayTicketInfo() {
        System.out.println("Ticket Information:");
        System.out.println("Movie: " + movieName);
        System.out.println("Ticket Type: " + ticketType);
        System.out.println("Price: $" + price);
    }

    // Setters
    public void setMovieName(String movieName) {
        this.movieName = movieName;
    }

    public void setTicketType(String ticketType) {
        this.ticketType = ticketType;
    }

    public void setPrice(double price) {
        if (price >= 0) { // Ensure price is non-negative
            this.price = price;
        } else {
            System.out.println("Invalid price. Price cannot be negative.");
        }
    }
}
```

The Ticket class is an abstract class that represents a ticket for a movie in the cinema system. It defines the basic structure and behaviors common to all ticket types (e.g., Regular and VIP tickets), but leaves specific ticket price calculation to subclasses.
Here's a breakdown of its components:

1. *Attributes:*
   - *movieName: A String representing the name of the movie for which the ticket is issued.*
   - *price: A double representing the price of the ticket.*
   - *ticketType: A String indicating the type of the ticket (e.g., "Regular" or "VIP").*
2. *Constructor:*
   - *The constructor initializes the ticket with a movie name and ticket type (e.g., "Regular" or "VIP").*
3. *Methods:*
   - *calculatePrice(): An abstract method that must be implemented by subclasses (e.g., VIPTicket or RegularTicket) to define how the ticket price is calculated.*
   - *displayTicketInfo(): Prints the ticket's information, including the movie name, ticket type, and price.*
   - *Setters for the attributes (setMovieName(), setTicketType(), setPrice()) allow modification of the ticket details. The setPrice() method ensures that the price is non-negative and prints an error message if an invalid price is provided.*

The Ticket class serves as a blueprint for all types of tickets, providing a structure to store ticket details and a method to display them. The actual price calculation logic is handled by the subclasses, like VIPTicket and RegularTicket.



CLASS: VIPTICKET & REGULARTICKET

```java
package cinematickts;
public class VIPTicket extends Ticket {
    // Constructor
    public VIPTicket(String movieName) {
        super(movieName, "VIP");
        calculatePrice();
    }

    // Overriding the calculatePrice method for VIP ticket
    @Override
    public void calculatePrice() {
        this.price = 15.0; // VIP ticket price
    }
}
```

```java
package cinematickts;
public class RegularTicket extends Ticket {
    // Constructor
    public RegularTicket(String movieName) {
        super(movieName, "Regular");
        calculatePrice();
    }

    // Overriding the calculatePrice method for Regular ticket
    @Override
    public void calculatePrice() {
        this.price =10.0; // Regular ticket price
    }
}
```

The VIPTicket and RegularTicket classes are subclasses of the abstract Ticket class. They represent specific types of tickets with different pricing logic. Each class overrides the calculatePrice() method to define the ticket price for that specific type.

VIPTicket Class

- *Purpose: Represents a VIP ticket for a movie with a fixed price.*
- *Constructor:*
  - *The constructor takes the movie name as an argument and initializes the ticket with the type "VIP". It then calls calculatePrice() to set the price.*
- *Methods:*
  - *calculatePrice(): Overrides the abstract method from Ticket. It sets the price of the VIP ticket to a fixed value of $15.00.*

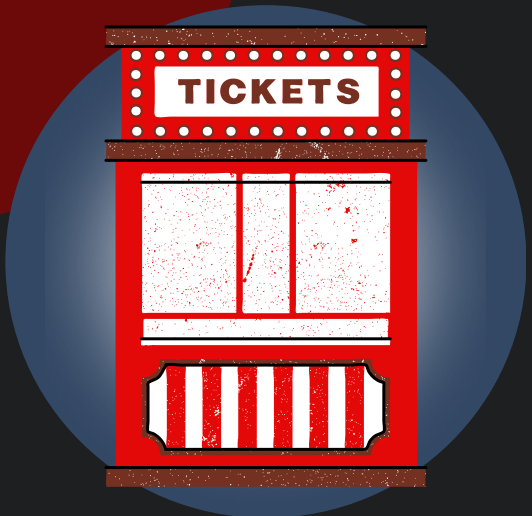
RegularTicket Class

- *Purpose: Represents a regular ticket for a movie with a fixed price.*
- *Constructor:*
  - *The constructor takes the movie name as an argument and initializes the ticket with the type "Regular". It then calls calculatePrice() to set the price.*
- *Methods:*
  - *calculatePrice(): Overrides the abstract method from Ticket. It sets the price of the regular ticket to a fixed value of $10.00.*

Summary of Differences:

- *Both classes extend the Ticket class and inherit its attributes and methods.*
- *VIPTicket sets a fixed price of $15.00, while RegularTicket sets a fixed price of $10.00.*
- *The calculatePrice() method in each class defines the specific price logic for the respective ticket type.*

These two classes represent the specific ticket options available for a customer, allowing them to choose between a regular ticket and a VIP ticket with different prices.



CLASS: CUSTOMER

```java
package cinematickts;
import java.util.Scanner;

public class Customer {
    private String name;

    // Constructor
    public Customer(String name) {
        this.name = name;
    }

    // Method to book a ticket
    public Ticket bookTicket(Cinema cinema) {
        Scanner scanner = new Scanner(System.in);

        // Display available movies
        cinema.displayMovies();

        System.out.print("Enter the number of the movie you want to book: ");
        int movieChoice = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        // Get the selected movie name
        String movieName = cinema.getMovie(movieChoice - 1);

        // Ask if the customer wants a VIP ticket
        System.out.print("Do you want a VIP ticket? (yes/no): ");
        String vipChoice = scanner.nextLine().toLowerCase();

        // Create a Ticket object based on the VIP choice
        Ticket ticket;
        if (vipChoice.equals("yes")) {
            ticket = new VIPTicket(movieName);
        } else {
            ticket = new RegularTicket(movieName);
        }

        return ticket;
    }
}
```

The Customer class represents a customer who interacts with the cinema system to book tickets. It allows the customer to select a movie and choose between a Regular or VIP ticket. Here's a breakdown of its components:

1. Attributes:

- *name: A String representing the name of the customer.*

2. Constructor:

- *The constructor takes the customer's name as an argument and initializes the name attribute.*

3. Methods:

- *bookTicket(Cinema cinema): This method allows the customer to book a ticket. It performs the following actions:*
  - *Displays the list of available movies using cinema.displayMovies().*
  - *Prompts the customer to select a movie by entering the movie number.*
  - *Based on the customer's choice, it retrieves the selected movie name from the Cinema object.*
  - *Asks the customer whether they want a VIP ticket by prompting them with a yes/no question.*
  - *Depending on the customer's choice, it creates either a VIPTicket or a RegularTicket for the selected movie.*
  - *Returns the created Ticket object (either VIP or Regular).*

4. Interaction:

- *The Customer class interacts directly with the Cinema class to retrieve available movies and book a ticket. It also uses the Ticket class to generate and return a ticket, ensuring the proper ticket type (VIP or Regular) is created based on the customer's preference.*

Summary:

The Customer class represents an individual who books a ticket in the cinema system. It allows the customer to select a movie and choose between a regular or VIP ticket, facilitating the booking process in the system.

CLASS: TESTCINEMA

```java
package TestCinema;
import cinematickts .*;
import java.util.Scanner;

public class Testcinema {
    public static void main(String[] args) {
        // Create a new Cinema object
        Cinema cinema = new Cinema();

        // Add some sample movies to the cinema
        cinema.addMovie(new Movie("Avatar", 10.0));
        cinema.addMovie(new Movie("Inception", 14.0));
        cinema.addMovie(new Movie("Spider-Man: Homecoming", 12.0));
        cinema.addMovie(new Movie("Avengers: Infinity War ", 10.0));
        cinema.addMovie(new Movie("The Dark Knight", 13.0));

        System.out.println("Welcome to Cinema Ticket Booking System!");

        // Ask for customer name
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your name: ");
        String customerName = scanner.nextLine();

        // Create a new Customer object
        Customer customer = new Customer(customerName);

        // Book a ticket for the customer
        Ticket ticket = customer.bookTicket(cinema);

        // Display the ticket information
        System.out.println("\nThank you for booking! Here are your ticket details:");
        ticket.displayTicketInfo();

        scanner.close();
    }
}
```

The TestCinema class is the main entry point for the Cinema Ticket Booking System. It demonstrates how the various components of the system (such as movies, customers, and tickets) work together. The class contains the main() method, where the program execution begins.

Here's a breakdown of its components:

1. Attributes:

- *The TestCinema class does not have any attributes as it primarily serves to run the program.*

2. Main Method:

- *Create a Cinema Object: The program starts by creating a Cinema object, representing the cinema where movies are available for booking.*
- *Add Movies: Several Movie objects are created and added to the cinema's movie list using cinema.addMovie() method. This populates the cinema with a selection of movies like "Avatar", "Inception", "Spider-Man: Homecoming", etc.*
- *Display Welcome Message: The program displays a welcome message to the user, greeting them and explaining that they are interacting with the cinema ticket booking system.*
- *Prompt for Customer Name: It asks the user for their name, which is then used to create a Customer object.*
- *Book a Ticket: Using the Customer object's bookTicket() method, the customer books a ticket from the cinema. This method will display the available movies and allow the user to select one, as well as choose whether they want a VIP or Regular ticket.*
- *Display Ticket Information: After the ticket is booked, the program displays the details of the ticket (movie name, ticket type, and price) using the Ticket object's displayTicketInfo() method.*

3. Functionality:

- *User Interaction: The TestCinema class serves as the user interface for interacting with the system, guiding the user through the process of selecting a movie and booking a ticket.*
- *Cinema and Customer Interaction: The class demonstrates how a customer interacts with the Cinema object to choose a movie and how the Customer class communicates with the Ticket class to generate a ticket.*

Summary:

The TestCinema class acts as the driver of the program, setting up the cinema system, adding movies, and allowing a customer to book a ticket. It provides the overall flow of the cinema ticket booking process, from choosing a movie to displaying the ticket details.

UML

Movie
<cinematickets>

- title : String
- price :double

+ Movie(title: String, price: double)
+ get Title ( ) : String
+ get price ( ) : double
+ setPrice(price: double): void
+ setTitle(title: String): void

Cinema
<cinematickets>

- movies : List <Movie>

+ CinemaO
+ addMovie (movie: Movie): void
+ displayMovies( ) : void
+ setMovies(movies: List<Movie>): void
+ getMovie (index: int) String

TestCinema
<TestCinema>

+ main (args : Sting [ ]) : void

Ticket
<cinematickets>

- movieName : String
- price : double
- ticketType : String

+ Ticket (movie name : String , ticketType : String)
+ setMovieName(movieName: String): void
+ setPrice(price: double): void
+ setTicketType(ticketType: String): void
+ CalculatePrice ( ) : void (abstract)
+ displayTicketInfo ( ) : void

VIPTicket
<cinematickets>

+ VIPTicket (movieName : String)
+ calculatePrice ( ) : void

RegularTicket
<cinematickets>

+ RegularTicket (movie name : String)
+ calculatePrice ( ) : void

Customer
<cinematickets>

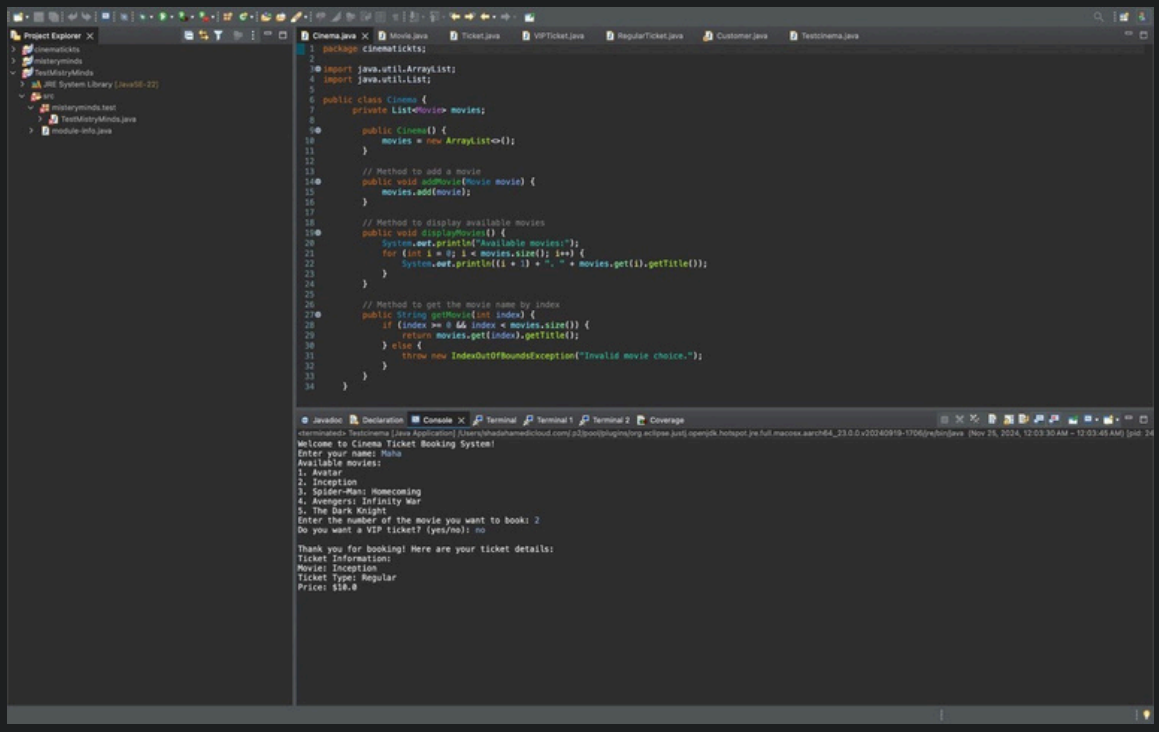
- name : String

+ Customer (name : String)
+ bookTicket (cinema : Cinema) : Ticket

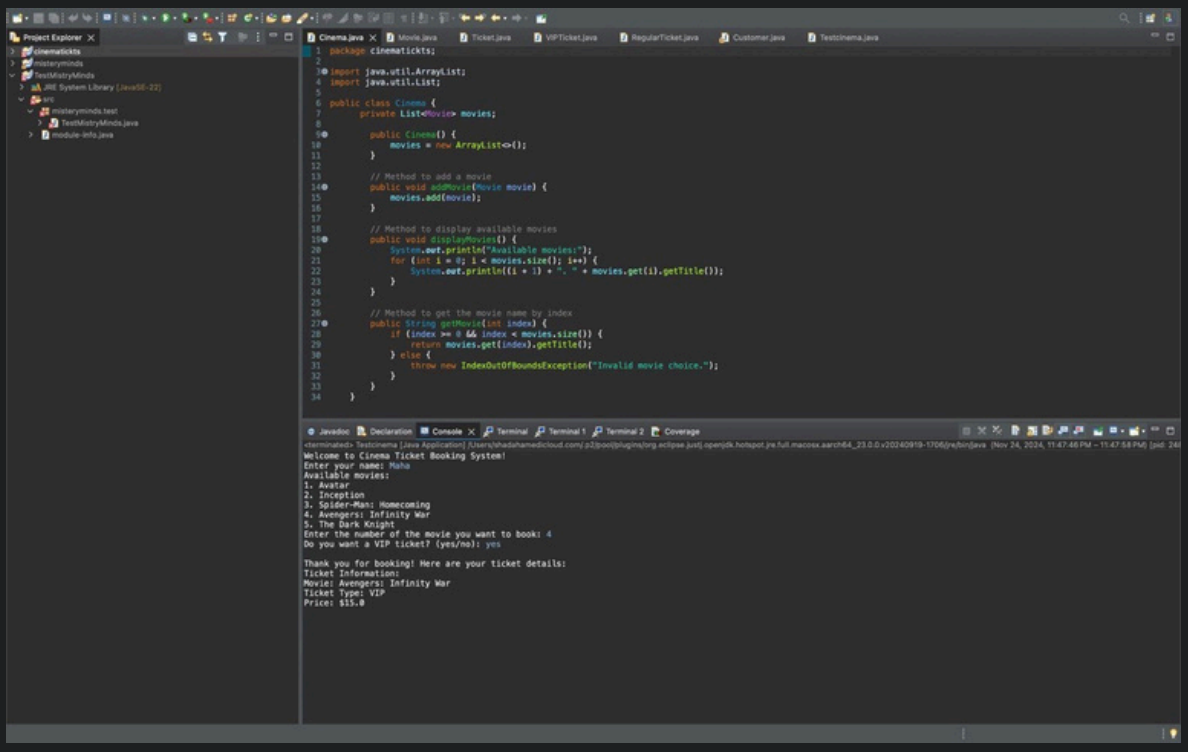
Has-A

Is-A

OUTPUTS:



```
Welcome to Cinema Ticket Booking System!
Enter your name: Maha
Available movies:
1. Avatar
2. Inception
3. Spider-Man: Homecoming
4. Avengers: Infinity War
5. The Dark Knight
Enter the number of the movie you want to book: 2
Do you want a VIP ticket? (yes/no): no

Thank you for booking! Here are your ticket details:
Ticket Information:
Movie: Inception
Ticket Type: Regular
Price: $10.0
```



```
Welcome to Cinema Ticket Booking System!
Enter your name: Maha
Available movies:
1. Avatar
2. Inception
3. Spider-Man: Homecoming
4. Avengers: Infinity War
5. The Dark Knight
Enter the number of the movie you want to book: 4
Do you want a VIP ticket? (yes/no): yes

Thank you for booking! Here are your ticket details:
Ticket Information:
Movie: Avengers: Infinity War
Ticket Type: VIP
Price: $15.0
```

THANK YOU