



- Realization between (movie factory) and (horror movie factory, comedy movie factory)
- Realization between (observer) and (user)
- Realization between (Ticket) and (basic ticket, ticket decorator)
- Generalization of (horror movie, comedy movie) from (movie)
- Generalization of (VIP decorator, Snack decorator) from (ticket decorator)
- Composition between (ticket) and (ticket decorator)
- Composition between (screen) and (seat)
- (User) associates with (ticket)
- (User) associates with (payment)
- (booking) associates with (notification)

Roles of classes:

1. Theatre

- Represents a physical cinema location. Contains multiple screens.
- *Example:* "Cineplex Downtown" with 10 screens.

2. Screen

- An individual auditorium within a theatre (e.g., "Screen 5: IMAX").
- Manages seat layouts and showtimes.

3. Seat

- A single bookable seat (e.g., "A12"). Tracks availability and type (VIP/Standard).

4. Booking

- A user's reservation. Links tickets, payment, and showtime.
- *Attributes:* `booking_id`, `payment_status`.

5. Payment

- Processes transactions (credit card, amount, status).
- *Methods:* `processPayment()`, `refund()`.

6. Notification

- Sends confirmations/reminders via email/SMS.
- *Methods:* `sendConfirmation()`, `sendReminder()`.

7. MovieFactory (Interface)

- Defines `createMovie()` for genre-specific movies.
- *Implemented by:* `ComedyMovieFactory`, `HorrorMovieFactory`.

8. ComedyMovieFactory / HorrorMovieFactory

- Creates movies with genre-specific features (e.g., trailers, pricing).
- *Outputs:* `ComedyMovie`, `HorrorMovie` objects.

9. Movie (Abstract)

- Base class for all movies. Defines `title`, `duration`, `displayDetails()`.

10. ComedyMovie / HorrorMovie

- Concrete movie types.
- *Example:* `ComedyMovie` adds `humorStyle`; `HorrorMovie` adds `scareLevel`.

11. **Ticket** (Interface)

- Defines `getCost()` and `getDescription()` for all tickets.

12. **BasicTicket**

- Standard ticket with base price.
- *Example:* "\$12 for Seat A12".

13. **TicketDecorator** (Abstract)

- Wraps a `Ticket` to add features. Base for all decorators.

14. **VIPTicketDecorator**

- Adds VIP perks (e.g., +\$5 for lounge access).
- *Method:* `getCost()` returns `ticket.getCost() + 5`.

15. **SnackDecorator**

- Adds snacks/drinks (e.g., +\$3 for popcorn).

16. **Observer** (Interface)

- Defines `update()` for notification recipients.

17. **User**

- Implements `Observer`. Receives booking alerts.
- *Method:* `update("Your booking is confirmed!")`.