Cinema

Room

- roomName: string
- roomType: RoomType
- seates: List<Seat>

+Room(string roomName, RoomType roomType, int rowsCount, int seatesPerRow)
```
    {
        this.roomType = roomType;
        this.roomName = roomName;
        seates = new List<Seat>();

        for (int i = 0; i < rowsCount; i++)
        {
            char row = (char)(65 + i);
            for (int j = 1; j <= seatesPerRow; j++)
            {
                seates.Add(new Seat(row, j));
            }
        }
    }
```

1   *

- title: String
- date: String
- r: Room
- price: int

+ Show(strin
```
    {
        this.ti
        this.d
        this.r
        this.p
    }
```

-seates   *

Seat

- row: char
- seat: int
- state: SeatState

+Seat(char row, int seat)
```
    {
        this.row = row;
        this.seat = seat;
        state = SeatState.free;
    }
```

#show
#seat
#price

1   + Tick

Child : Ticket

+Child(Show show, Seat seat) : base(show, seat)
```
{
  switch (show.getRoom().getRoomType())
  {
    case RoomType.small:
      this.price *= (100 - 40) / 100.0;
      break;
    case RoomType.medium:
      this.price *= (100 - 40) / 100.0;
      break;
    default:
      break;
```

Student : Ticket

+Student(Show show, Seat seat) : base(show, seat)
```
{
    switch (show.getRoom().getRoomType())
    {
      case RoomType.small:
        this.price *= (100 - 30) / 100.0;
        break;
      case RoomType.medium:
        this.price *= (100 - 20) / 100.0;
        break;
      default:
```

Adult : Ticket

+ Adult(Show show, Seat seat) : base(sh
```
{
    switch (show.getRoom().getRoomTyp
    {
      case RoomType.small:
        this.price *= (100 - 10) / 100;
        break;
      default:
        break;
    }
}
```
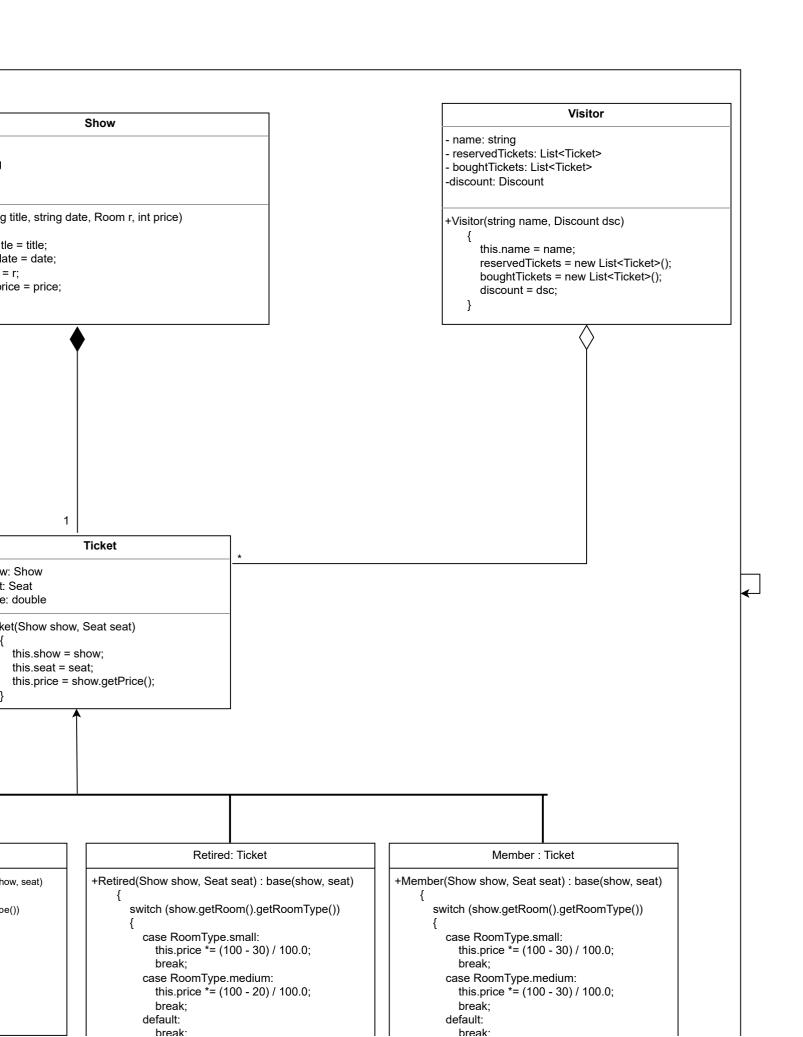
Show

g title, string date, Room r, int price)

tle = title;
ate = date;
= r;
rice = price;

Visitor

- name: string
- reservedTickets: List<Ticket>
- boughtTickets: List<Ticket>
-discount: Discount

+Visitor(string name, Discount dsc)
   {
     this.name = name;
     reservedTickets = new List<Ticket>();
     boughtTickets = new List<Ticket>();
     discount = dsc;
   }

1

Ticket

w: Show
t: Seat
e: double

ket(Show show, Seat seat)
{
   this.show = show;
   this.seat = seat;
   this.price = show.getPrice();
}

*

Retired: Ticket

+Retired(Show show, Seat seat) : base(show, seat)
   {
     switch (show.getRoom().getRoomType())
     {
      case RoomType.small:
       this.price *= (100 - 30) / 100.0;
       break;
      case RoomType.medium:
       this.price *= (100 - 20) / 100.0;
       break;
      default:
       break;

Member : Ticket

+Member(Show show, Seat seat) : base(show, seat)
   {
     switch (show.getRoom().getRoomType())
     {
      case RoomType.small:
       this.price *= (100 - 30) / 100.0;
       break;
      case RoomType.medium:
       this.price *= (100 - 30) / 100.0;
       break;
      default:
       break;

how, seat)

be())

```
                break;
            }
        }
```

```
                default:
                    break;
            }
        }
```

```
                default:
                    break;
            }
        }
```

```
                break;
            }
        }
```

```
                break;
            }
        }
```