

Problem Statement:

There is a venue which hosts 3 movies and 2 concerts: so there's a class called **Venue**, as well as a **Movie** class and a **Concert** class. For the concerts and movies, you need to book tickets.

There's a ticket class and two child classes of this ticket class called the **concertTicket** and **movieTicket** classes. They inherit some features of the ticket class.

Movie tickets can be gold, silver or bronze and their price varies accordingly. Gold gives you a ticket for the first three rows of a theatre hall, Silver for the next three rows, and Bronze tickets for the remaining rows.

Concert tickets can be VIP or basic and their price varies accordingly. There is a set capacity or maximum for the VIP and basic tickets.

In the movie class, we have a seat matrix based on which we find the next available seat of each type (first 3 rows are gold, next 3 silver, rest bronze). In concert only the total capacity matters for VIP and basic.

In the venue class, based on the screen number (each movie is assigned a particular screen number) or concert number (each artist is assigned a concert number) and ticket type, we check if we can book a ticket for that particular type (gold, silver, bronze for movie and VIP, basic for concert) and for that particular event and if we can, we calculate its price after possible discount and then return the ticket.

There is a discount if the total movie tickets (across all 3 movies) are a multiple of 10 or if the total concert tickets (across both concerts) are a multiple of 10. We need to thus keep a static variables which counts the total tickets booked for each class.

You may have to convert int row number into char row number (for eg, row 0 corresponds to a, row 1 to b and so on..)

Caveats

- Read the javadoc carefully and ensure that the function names are the same as the ones written there.
- Check the javadocs for the return values and exact prototypes of the functions.

- Once you have written the code in Eclipse, you can check if it passes all the test cases by importing the junit5 library and running the jar file(as demonstrated).
- Once you have made sure the code works, it must be uploaded to codePost for checking.
- Code on codePost can be uploaded multiple times. The final upload will be considered.
- Make sure you upload the code before the deadline.