

# STATEMENT

“El clásico” between Barcelona and Real Madrid football teams will take place the 20 of December. A lot of people will go to the match and; for this purpose, we will have to help them with the tickets. In this exercise, a datatype class called `FootballTicket` has to be built. The aim of this activity is to create two `FootballTicket` objects with their corresponding data (month and day where they were bought, type of ticket (VIP, normal or grada jove), member ID and the price). Then, some methods must be created (consultors, modifiers, a method that calculates the final price of the ticket based on some discounts and a method that raffle a prize among the members who bought a ticket). To do it, the next instructions must be followed:

- Private attributes for the type of ticket (`char`), for the member ID (`int`), for the final price of the ticket (`double`) and for the date and time when the ticket is bought: month, day (`int`).
- Public class (`static`) constant attributes for the base price (`double`) which is 30 euros, for the date of the match (the month and the day) (`int`), and for the type of ticket: vip (`'V'`), grada jove (`'J'`) or normal (`'N'`).
- A constructor without parameters, that must initialise properly all attributes, using a standard date of purchase (The first of December), the final price without charges or discounts and normal as default type of ticket.
- A constructor that receives as parameters the date where the ticket was bought (2 `int` for month and day), the member ID (`int`) (in case he/she is not a member, ID must be 0), the type of ticket (`char`) and the final price (computed by the method explained below). The constructor must initialise properly all attributes. The ID written must be a number between 0 and 9999.
- `Get` and `set` methods for month, day, id, type of ticket and final price attributes with the following restrictions:
  - The id must be a number between 0 and 9999.
  - The final price must be positive.
  - `set` method referred to type of ticket must avoid modifications to values not defined in the constants.
- An `equals` method that compares if two `FootballTicket` objects are equal, taking into account the date (day and month), the member ID, the type of ticket and the final price.
- A `toString` method that returns the data of the ticket in two different ways:
  - If he/she is a member: Ticket for the Barcelona/Madrid football match of 20/12 at 21:30. Member with id: ID. Type of seat: TYPE, and price: PRICE euros.
  - If he/she is not a member: Ticket for the Barcelona/Madrid football match of 20/12 at 21:30. Not member. Type of seat: TYPE, and price: PRICE euros.

- A method that computes the final price of the ticket. If he/she is a member, a discount of a 50% to the base price will be applied. Then, depending on the type of ticket, some modifications must be done. If the type is VIP, a charge of 50% must be added to the price. If it is grada jove, the ticket will cost 5 euros less, and if it is normal, no modifications must be done. Finally, if he/she bought the ticket more than a week before the match, a discount of the 15% will be applied to the price.
- A method that gets a random number that must be equal to the ID of one of the members, in order to raffle a prize among all of them.