

Ejercicio para practicar mapas

Implemente la clase **Juego** la cual esté compuesta por un nombre de tipo cadena, que representa el nombre del juego. Un número máximo de jugadores de tipo entero, que representa el número máximo de jugadores que han recibido puntuación en el juego. Un mapa de la librería `map<key,value>` de la STL que almacene el dni y la puntuación obtenida de cada jugador. El dni debe ser único, por lo que no puede haber dos jugadores con el mismo dni. El dni será de tipo cadena, mientras que la puntuación de tipo entero.

Codificar la siguiente funcionalidad:

- **Constructor parametrizado**, cuyos únicos parámetros obligatorios son el nombre del juego y el número máximo de jugadores.
- Método **addJugador()**. Este método recibe una cadena con el dni del jugador e inserta el jugador con puntuación inicial cero. El método será booleano, ya que si el jugador se encuentra insertado no se insertará y retornará false, y en caso de ser insertado, retornará true.
Nota: si el juego ha alcanzado el número máximo de jugadores no se podrá añadir más jugadores.
- Método **incrementarPuntuacion()**. Este método recibe la puntuación que se desea incrementar y el dni del jugador al cual se le desea incrementar dicha puntuación. El método será booleano, ya que si el jugador no se encontrase no se puede incrementar la puntuación y el método retornará false. En caso de ser incrementada se retornará true.
- Método **getPuntuacion()**, que recibe el dni de un jugador y retorna la puntuación de dicho jugador. Si el jugador pasado como argumento no se encontrase en el mapa, el método retornará -1.



- Método **size()**. Este método retorna el número de jugadores almacenados.
- Método **getGanador()**. Dicho método retorna el dni del jugador con mayor puntuación. Si no hay jugadores registrados, el método retornará “No hay jugadores registrados”. Si existiesen más de un jugador con la puntuación máxima, el método retornará el primer jugador que haya alcanzado dicha puntuación.
- Método **getNombre()**. Este método será el observador del nombre del juego.
- Método **getMaxJugadores()**. Este método será el observador del máximo número de jugadores establecido.
- Método **toString()**. Dicho método retornará una cadena la cuál contendrá el nombre del juego, seguido de los jugadores y las puntuaciones de los jugadores registrados en el juego. Este sería un ejemplo de la cadena que se desea retornar:

“Juego: La brisca, Jugadores: [2342D, 23] [5746T, 132] [5746H, 84]”

Debe codificar la clase en los ficheros juego.cc y juego.h. Para compilar debe realizar desde la terminal “make test” y para ejecutar “./test”.

605 45 70 99 (Ángel)
Únicos especialistas en Grado en Ingeniería Informática



www.academiamain.es