

1o-Parcial.pdf



imsanti



Programación Orientada a Objetos



1º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

**“Mi imperio romano
es el Bizum que me
debe mi amigo el rata.”**



Cuenta NoCuenta

Perfecta para hacer todos
tus Bizums... ¡y pedirlos!

Cuéntame más



(CERTIFICA TU NIVEL DE INGLÉS **B1-B2 Y C1**)

YES



CAMBRIDGE
English



Past



Somos Centro Examinador, certifica tu nivel
de Inglés **B1- B2 - C1** en tan solo **48 horas**.
Reconocido por las Universidades.

QUE NO TE PILLE EL TORO...



Matricúlate AHORA

Proyecto prGolf (2 puntos)

Se va a crear una aplicación para mantener información sobre un torneo de golf. En dicho torneo participarán jugadores que tendrán que recorrer los 18 hoyos del campo anotando los golpes que necesitan para completarlos. La puntuación de un jugador será el número de golpes dados por el jugador. En el torneo también pueden participar jugadores aficionados que tendrán un hándicap que alterará su puntuación. Para todo ello se van a crear las clases Jugador, Aficionado y Torneo, todas dentro del paquete golf.

- Mientras no se indique lo contrario, las variables de instancia serán privadas y los métodos públicos.
- Pueden añadirse los métodos privados que se consideren.

Clase Jugador (0.40 ptos.)

- (0.10 ptos.) Crea la clase Jugador que mantiene información sobre el nombre de un jugador (`String nombre`) y el número de golpes que ha necesitado dar para completar los 18 hoyos (`int numGolpes`). La clase tendrá un constructor cuyo argumento es el nombre del jugador.
- (0.05 ptos.) Crea el método `String getNombre()` que devuelve el nombre del jugador.
- (0.05 ptos.) Crea el método `int puntuacion()` que devuelve la puntuación del jugador. La puntuación de un jugador coincide con el número de golpes.
- (0.05 ptos.) Crea el método `void agregaGolpes(int glp)` que agrega los golpes `glp` al número de golpes de este jugador.
- (0.15 ptos.) Da una representación de los jugadores de manera que un jugador llamado "Smith" que haya dado 78 golpes se vea como

```
nombre= Smith, golpes= 78
```

Clase Torneo (1.10 pto.)

- (0.10 ptos.) La clase Torneo mantiene información de todos los jugadores inscritos en el torneo en una lista de jugadores (`List<Jugador> inscritos`). Cada torneo tiene un "par del campo" que es el número de golpes promedio que se necesitan dar para completar los 18 hoyos (`int parCampo`). En el constructor se le pasa la lista de jugadores y el par del campo.
- (0.15 ptos.) Define el método `private int busca(String nombre)` en el que dado el nombre de un jugador, devuelve la posición en la que se encuentra el jugador con ese nombre dentro de la lista de jugadores inscritos. Si el nombre no aparece en la lista, devuelve -1.
- (0.05 ptos.) Define el método `void agregaGolpes(String nombre, int glp)` en el que dado el nombre de un jugador, lo localiza en la lista de inscritos y le añade los golpes dados como segundo argumento del método. Si el nombre no aparece en los inscritos debe lanzar una `RuntimeException` indicando el motivo.
- (0.05 ptos.) Define el método `int tarjetaDe(String nombre)` que devuelve la tarjeta de un jugador con ese nombre. Se define la tarjeta de un jugador como la diferencia entre la puntuación obtenida por el jugador y el par del campo. Gana el torneo el que tiene menor tarjeta (este valor puede ser negativo). Si el nombre no aparece en los inscritos debe lanzar una `RuntimeException` indicando el motivo.

WUOLAH

- (0.45 ptos.) Define el método `private int mejorTarjeta(List<String> aux)` que dada una lista de nombres de jugadores, devuelve la posición en esa lista del nombre del jugador que tiene la mejor tarjeta. Si hay varios con la misma tarjeta devuelve el primero. Solo se consideran los jugadores cuyo nombre aparezcan en `aux`.
- (0.50 ptos.) Define el método `public List<Jugador> clasificacion()` que devuelva la clasificación del torneo, es decir una lista de los jugadores inscritos ordenados de menor a mayor tarjeta. El primero de esta lista será el ganador. Para resolver este método se debe proceder de la siguiente manera:
 - Se crea una lista vacía de jugadores. Esta lista será la que al final del proceso contendrá la clasificación.
 - Se crea una lista auxiliar de cadenas de caracteres que contendrá a todos los nombres de los jugadores inscritos.
 - Mientras la lista auxiliar no esté vacía
 - Se busca la posición en la lista auxiliar del nombre del jugador con mejor tarjeta.
 - Se localiza a este jugador entre los inscritos y se añade a la lista que mantiene la clasificación (por el final).
 - Se elimina el nombre de este jugador de la lista auxiliar.
 - Se devuelve la lista con la clasificación.

Se proporciona un programa de prueba cuyo resultado debe ser:

```
Torneo con solo profesionales
nombre= Juan Garcia, golpes= 71 tarjeta = -2
nombre= VanDerPoel, golpes= 72 tarjeta = -1
nombre= HerMan, golpes= 73 tarjeta = 0
nombre= James Lee, golpes= 77 tarjeta = 4
```

Clase Aficionado (0.40 ptos.)

- (0.10 ptos.) Crea la clase `Aficionado` que se comporta como un jugador pero que tiene un `hándicap` (`int handicap`) que es un valor que se utilizará para redefinir su puntuación.
- (0.10 ptos.) En el constructor se le pasa además del nombre del jugador, su `hándicap`. Este `hándicap` debe ser un número entre 1 y 54. Se debe lanzar una `RuntimeException` si el `hándicap` no es válido.
- (0.10 ptos.) La puntuación para un aficionado se calcula restando a la puntuación como jugador su `hándicap`.
- (0.10 ptos.) La representación de un aficionado será como la de un jugador, pero añadiendo el `hándicap`. Así, un aficionado llamado "Smith" que ha dado 98 golpes y tiene un `hándicap` de 15 se verá como:

```
nombre= Smith, golpes= 98, handicap= 15
```

Puede utilizarse el mismo programa de prueba cambiando `ejemplo1` por `ejemplo2` (quitar los comentarios al método `ejemplo2`) y el resultado debe ser:

```
Torneo con aficionados y profesionales
nombre= Pedro Lopez, golpes= 97, handicap= 35 tarjeta = -11
nombre= Heissemberg, golpes= 73 tarjeta = 0
nombre= Harry Ran, golpes= 77 tarjeta = 4
nombre= Eric Met, golpes= 99, handicap= 20 tarjeta = 6
```

