

Introducción a la Ciencias de la Computación

2021-1

Proyecto Final

Profesora: Amparo López Gaona
Ayudante: Diana Isabel Ramírez García
Ayudante: Francisco Manuel Monreal Gamboa

Fecha de entrega:

1. Problema

En la facultad de Ciencias se realizará un torneo de juegos de mesa con una duración de tres días.

Antes de iniciar el torneo se realizará un registro de participantes , los datos a pedir son: nombre, fecha de nacimiento(tal que cumpla con la norma : dd/mm/aaaa)₁, sexo y carrera a la que pertenecen. Será necesario que el sistema genere una clave de participante para identificarlos.

HINT₁:Puedes solicitar los datos por separado y posteriormente concatenarlos en un String.

Los administradores del torneo para agilizar y llevar un mejor manejo, requieren tener un menú con las siguientes opciones:

- Concursantes registrados
- Partidas registradas: Muestra aquellas partidas ya concluidas, es decir a los jugadores y el estado de la partida (empate o el nombre del ganador).
- Lista del torneo: Muestra las parejas del torneo del día correspondiente. Las parejas deberán ser concisas es decir, si aún no cambia de día al visualizar las parejas estas deberán ser las mismas.
- Campeón: Mostrará al jugador con mejor puntuación hasta el momento. En caso de haber más de un jugador con la misma mejor puntuación, de igual manera se mostrarán.

Por último se tendrá el torneo, como se mencionó anteriormente , este contará con una duración de 3 días. Cada día concluirá cuando todos las partidas programadas tengan algún resultado. En cada día habrá un juego distinto, de la siguiente manera:

HINT: Puedes simular el inicio y fin de cada día guardando la información registrada en un archivo.

- Día 1: El juego seleccionado será Mini Blackjack, el ganador será aquel cuya suma de su mano sea la más cercana al 21 sin pasarse, de lo contrario pierde automáticamente.
EL valor de las cartas :

- As puede valer 1 o 11.
- J, Q, K tienen el valor de 10.

Cada mano del jugador empieza con una carta, posteriormente puedes pedir otra carta o quedarte. Si el jugador decide quedarse ya no puede seguir pidiendo cartas. El juego puede terminar de 2 maneras: Cuando alguno de ellos rebasa los 21 o hasta que ambos jugadores decidan quedarse y se elija un ganador, que será aquel cuyo número sea más cercano a 21.

- Día 2: El juego seleccionado será dados tal que consiste en conseguir un 7 o un 11 en los dados, y se pierde si sale un 2, 3 o 12. De salir un 4, 5, 6, 8, 9 o 10, entonces el jugador vuelve a tirar. Si en la nueva tirada sale el mismo número de la partida anterior gana, por el contrario pierde. En caso de que ambos jugadores tiren un 7 o un 11, se repite la partida hasta que alguno pierda.
- Día 3: El juego seleccionado será un conecta cuatro con un tablero de 4x4. El jugador únicamente puede elegir la columna en donde desea depositar su ficha.

Reglas del torneo:

1. Se asignarán parejas diferentes para los partidos de cada día. En caso de que un jugador quede sin pareja jugará contra la computadora.
2. Para iniciar una partida es necesario ingresar desde la consola las claves de los jugadores. Si las claves no coinciden se deberá mandar un error. Si la consola recibe únicamente una clave se sobreentiende que juega contra la computadora siempre y cuando así lo indique la lista de parejas, de lo contrario manda error. Si no recibe ninguna clave, se manda error.

```
>java JuegoEjemplo jugador8
"fulanito"(jugador8) jugará contra la computadora...

>java JuegoEjemplo jugador1 jugador25
"perenganito"(jugador1) jugará contra "zutanito"(jugador25)...

>java JuegoEjemplo
Se requiere por lo menos un jugador.
```

3. El ganador de cada partida obtendrá 10 puntos, mientras que el perdedor ninguno. En caso de haber un empate se repetirá la partida, si en el segundo intento de nuevo hay un empate, entonces cada jugador se llevará 5 puntos.
4. El ganador del torneo será aquel que tenga la mayor cantidad de puntos al final de los 3 días. Si hay un empate se felicitará a todos los jugadores con esa puntuación.

HINT: Tu proyecto podrá contar con todas las clases que creas necesarias, de igual manera puedes tener más de un ejecutable si así lo requieres.

2. Requerimientos

El programa que desarrolles para solucionar el problema anterior necesita utilizar todos los temas vistos a lo largo del curso, que son:

- Creación y uso de datos primitivos
- Creación y uso de objetos
- Creación y uso de clases
- Objetos como atributos
- Arreglos y Arreglos bidimensionales
- Herencia
- Excepciones
- Clase Abstracta y/o Interfaces
- Serialización de objetos

3. Restricciones

1. Cada clase que programes debe ser robusta, es decir, debe estar preparada para manejar cualquier error que pueda presentarse.
2. Cada clase debe incluir por lo menos un constructor, métodos modificadores e inspectores para cada uno de los atributos que tendrán sus objetos. No dupliques métodos.
3. Debes emplear los conceptos vistos en clase, en particular debes crear al menos una jerarquía de clases.
4. No se debe permitir que el programa termine por una excepción no controlada.
5. Es indispensable que cuando se presente una excepción, el usuario sea notificado de los detalles de la misma. Por lo que cada excepción deba de ser manejada de forma individual y diferente.
6. Cada clase debe estar bien documentada haciendo uso de las etiquetas @author, @version, @param, @return y @throws.
7. Genera la documentación en HTML de cada una de las clases que programes y deberás guardar cada una de ellas en una carpeta llamada Documentación.

IMPORTANTE:

Queda completamente prohibido utilizar alguna biblioteca que no se haya visto durante el curso como: ArrayList, Arrays, LinkedList, etc.

4. Forma de entrega

IMPORTANTE:

Deberás incluir un archivo de texto README.txt en donde incluirás las indicaciones para la ejecución de tu programa, ya que estas serán las que nosotros tomaremos en cuenta. Deberás asegurarte que tus indicaciones sean claras y correctas, ya que si no están bien definidas, o no se entienden, **no se tomara en cuenta tu proyecto.**

Deberás incluir un archivo de texto Comentarios.txt con una breve explicación del programa, además podrás incluir cualquier comentario que te parezca necesario.

1. El proyecto se deberá de entregar de forma individual.
2. El proyecto (sus archivos y directorios) deberá estar contenido en un directorio llamado <apellidoPaterno_nombre_proyecto>
Por ejemplo: **monreal_francisco_proyecto**
3. **NO** incluir los archivos .class dentro de la carpeta
4. Comprimir el directorio <apellidoPaterno_nombre_proyecto> en un archivo <apellidoPaterno_nombre_proyecto>.zip
Por ejemplo: **monreal_francisco_proyecto.zip**
5. **Todos los archivos de código fuente deben estar documentados.**
6. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier proyecto plagiado total o parcialmente será penalizado con cero para los involucrados.
7. El proyecto será enviado al classroom de la clase en la sección que le corresponda.