Programación II - Trabajo Práctico Integrador
2do Cuatrimestre 2022
SEGUNDA PARTE

Fecha de presentación: Sábado 22 de octubre (subido al Campus)

Fecha de entrega: Sábado 5 de noviembre (por el Campus)

En esta segunda parte deben entregar la implementación, análisis de complejidad en donde se pida, y el IREP para la representación de datos seleccionada.
Para poder empezar con la segunda parte deben tener aprobado el diseño presentado en la primera parte.

Requerimientos técnicos:

- Grupos :**El mismo grupo de la primera parte del TP.** Si hay alguna modificación debe ser aprobada por sus docentes de la comisión.

- Se deben utilizar donde sea conveniente las herramientas de Tecnologías Java que se vieron en la materia. Al menos una vez deben usarse:

- *Stringbuilder,* cuyo uso debe basarse en la necesidad de modificar el string.
- *Iteradores y Foreach* para recorrer las colecciones de Java

Se deberá utilizar en el desarrollo del trabajo al menos 3 de estos conceptos: **herencia, polimorfismo, sobreescritura, sobrecarga e interfaces**. Como también, en los casos que corresponda, se deberá implementar **clases abstractas**. En el informe se debe explicar qué conceptos se utilizaron, dónde y de qué modo.

Por otro lado, desde la materia se proveerá

a) Un código cliente y la explicación de sus métodos para que se utilicen como base para la implementación,
b) La clase Fábrica con los métodos para facilitar la implementación. En esta clase les damos ya implementados los métodos que permiten generar el valor base de las figuritas, los premios, los países y sus ranking, y los datos de los mundiales. También se brindan los 32 países clasificados.
c) Una clase de testeo (junit). Será condición necesaria para aprobar esta parte del trabajo que tanto el código cliente como el test se ejecuten sin errores.

- Además de pasar el test de *junit* suministrado junto con el TP, en la corrección se testean los ejercicios con otro junit adicional, por lo que se recomienda armar un conjunto propio de testeo acorde a su implementación, antes de entregar el TP. Puede entregarlo también.

- **Escribir el *IREP* de la representación elegida para la implementación**.

La entrega se realiza subiendo al Campus el proyecto de Java con su implementación (Seleccionar solamente los archivos .java) Se debe entregar la documentación con los puntos pedidos previamente por escrito en un archivo Word junto con el proyecto.

Consideraciones importantes para la implementación:

La implementación de los TADs deben responder a su diseño presentado en la Primera Parte ***teniendo en cuenta las correcciones que se indicaron/indicarán a cada grupo*.**

- Deberá correr satisfactoriamente con el código cliente entregado
- Deberá pasar satisfactoriamente el test junit proporcionado.
- Deberá aprovechar correctamente las estructuras de datos elegidas.
- El código debe tener implementado el método ***toString*** del Sistema de Album.
- Se debe responder si un Participante cualquiera, dado su dni, completó su álbum en O(1).

- Explicar por escrito, la complejidad del método
  `int buscarFiguritaRepetida(int dni);`

Test (JUnit):

Se habilitará el archivo de test en el Moodle, junto a este enunciado, de donde deberán descargarlo.

Código Cliente:

```
public class CodigoCliente {

    public static void main(String[] args) {

        IAlbumDelMundial sistema = new AlbumDelMundial();

        sistema.registrarParticipante(222222, "Christian", "Tradicional");
        sistema.registrarParticipante(333333, "Mariana", "Extendido");
        sistema.registrarParticipante(111111, "José", "Web");
        sistema.registrarParticipante(555555, "Miguel", "Web");
        sistema.registrarParticipante(666666, "Jazzmine", "Extendido");
        sistema.registrarParticipante(777777, "Dante", "Tradicional");
        sistema.comprarFiguritas(222222);
        // El participante 111111 tiene album Web entonces tiene un codigo
        // promocional para solicitar 4 figuritas sin costo.
        sistema.comprarFiguritasConCodigoPromocional(111111);

        sistema.comprarFiguritas(222222);
        sistema.comprarFiguritas(333333);
```

```java
// El participante 333333 tiene un album tradicional y por eso
// puede participar en un sorteo por un premio instantaneo.
System.out.println(
            sistema.darNombre(333333) +
            " recibio por sorteo instantaneo: " +
            sistema.aplicarSorteoInstantaneo(333333)
      );
System.out.println();

List<String> pegadas = sistema.pegarFiguritas(222222);

if(pegadas.isEmpty()) { //o sea... no pego ninguna
      sistema.comprarFiguritas(222222);
      sistema.intercambiar(
            222222,
            sistema.buscarFiguritaRepetida(222222)
      );
}

sistema.pegarFiguritas(333333);

// Simulamos un uso prolongado del sistema.
for (int i =0;i<2000;i++) {
      sistema.comprarFiguritas(222222);
      sistema.pegarFiguritas(222222);
      sistema.comprarFiguritas(555555);
      sistema.pegarFiguritas(555555);
}
for (int i =0;i<500;i++) {
      sistema.comprarFiguritas(666666);
      sistema.pegarFiguritas(666666);
      sistema.comprarFiguritas(777777);
      sistema.pegarFiguritas(777777);
}

if(sistema.llenoAlbum(222222)) {
      System.out.println(
            sistema.darNombre(222222) +
            " recibio: " +
            sistema.darPremio(222222)
      );
System.out.println();
}
```

```
        System.out.println("Llenaron album:");
        System.out.println(sistema.listadoDeGanadores());
        System.out.println();

        System.out.println("Participantes que Llenaron el Pais Argentina:"
                           );
        for (String item:
              sistema.participantesQueCompletaronElPais("Argentina"))
              System.out.println(item);


              System.out.println();
              System.out.println("==========================================
        ========");

        System.out.println(sistema);
        }
 }
```

Se debe respetar la siguiente Interfaz:

```
public interface IAlbumDelMundial {

/**
* Registra un nuevo participante y devuelve el codigo unico del
* album asociado.
*
* Si el participante ya está registrado o el tipo de album es
* invalido, se debe lanzar una excepcion.
*/
int registrarParticipante(int dni, String nombre, String tipoAlbum);

/**
* Se generan 4 figuritas al azar y se asocia al participante
* correspondiente identificado por dni
* Si el participante no está registrado, se debe lanzar una excepción.
*/
void comprarFiguritas(int dni);

/**
* Se generan 4 figuritas top 10 al azar y
* se asocia al participante correspondiente identificado por dni
*
```

```
* Si el participante no está registrado, se debe lanzar una excepción.
* Si el participante no tiene album top10, se debe lanzar una excepción.
*/
void comprarFiguritasTop10(int dni);

/**
* Compra por única vez un grupo de 4 figuritas con el codigo promocional
* asociado a su album web.
*
* Si el participante no está registrado, se debe lanzar una excepción.
* Si el participante no tiene codigo de sorteo o el mismo ya fué usado,
* se debe lanzar una excepcion.
*/
void comprarFiguritasConCodigoPromocional(int dni);

/**
* Busca entre las figuritas del participante cuales aún no están en el
* album y las asocia.
* Devuelve una lista con las figuritas asociadas.
* De cada figurita se devuelve un string "$pais-$numeroJugador"
*
* Si el participante no está registrado, se debe lanzar una excepción.
*/
List<String> pegarFiguritas(int dni);

/**
* Verifica si el participante identificado por dni ya completó el album.
* Devuelve true si está completo, sino false.
* Este metodo debe resolverse en O(1)
*
* Si el participante no está registrado, se debe lanzar una excepción.
*/
boolean llenoAlbum(int dni);

/**
* Realiza el sorteo instantaneo con el codigo asociado al album
* tradicional del participante y devuelve algun premio al azar.
*
* Si el participante no está registrado, se debe lanzar una excepción.
* Si no tiene codigo para el sorteo o ya fue sorteado, se debe lanzar
* una excepcion.
*/
String aplicarSorteoInstantaneo(int dni);

/**
* Busca si el participante tiene alguna figurita repetida y devuelve
```

```
* el codigo de la primera que encuentre.
* Si no encuentra ninguna, devuelve el codigo -1.
*
* Si el participante no está registrado, se debe lanzar una excepción.
*/
int buscarFiguritaRepetida(int dni);

/**
* Dado el dni de un participante A y el codigo de una figurita,
* se busca entre los participantes con mismo tipo de album
* si alguno tiene repetida alguna figurita que le falte al participante A
* con un valor menor o igual al de la figurita a cambiar.
* En caso de encontrar alguno, realiza el intercambio y devuelve true.
* Si no se encuentra ninguna para cambiar, devuelve false.
*
*
* Si el participante no está registrado o no es dueño de la figurita a
* cambiar, se debe lanzar una excepción.
*/
boolean intercambiar(int dni, int codFigurita);

/**
* Dado el dni de un participante, busca una figurita repetida e intenta
* intercambiarla
* Devuelve true si pudo intercambiarla. Sino, devuelve false.
*
* Si el participante no está registrado, se debe lanzar una excepción.
*/
boolean intercambiarUnaFiguritaRepetida(int dni);

/**
* Dado el dni de un participante, se devuelve el nombre del mismo.
*
* Si el participante no está registrado, se debe lanzar una excepción.
*/
String darNombre(int dni);

/**
* Dado el dni de un participante, devuelve el premio correspondiente
* por llenar el album.
*
* Si el participante no está registrado, se debe lanzar una excepcion.
* Si no tiene el album completo, se debe lanzar una excepcion.
*/
String darPremio(int dni);
```

```
/**
* Devuelve un string con la lista de todos los participantes que
* tienen su album completo y el premio que les corresponde.
* El listado debe respetar el siguiente formato para cada ganador:
*     " - ($dni) $nombre: $premio"
*/
String listadoDeGanadores();

/**
* Devuelve una lista con todos los participantes que llenaron el pais
* pasado por parametro.
*
* De cada participante se devuelve el siguiente String:
*     "($dni) $nombre: $tipoAlbum"
*/
List<String> participantesQueCompletaronElPais(String nombrePais);
}
```