

Trabajo Práctico Programación II

Segunda parte

Estudiantes:

- **Giannini, Leila**
- **Sanchez, Gastón**

Profesores:

- **Nores, José**
- **Russo, Christian**

Comisión: 01



Informe

En este trabajo práctico realizamos un programa llamado "AlbumDelMundial" para la productora Figurini la cual desea que el programa administre los álbumes que se compran tanto en el formato papel como en el formato web, y que facilite el intercambio de las figuritas entre ellos. La empresa genera tres tipos de álbumes: tradicional, extendido y web, los cuales tienen un premio final. A su vez, cuando el cliente compra figuritas, tiene dos tipos, la común y la TOP10 que son más caras.

Para realizar este programa creamos las clases: Album, AlbumDelMundial, AlbumExtendido, AlbumTradicional, AlbumWeb, Figurita, FiguritaTop10, Pais, Participante y Tupla.

De las recién mencionadas, la clase llamada "AlbumDelMundial" es la clase principal. Y con respecto a la clase Tupla, la usamos para asociar un código de sorteo a un booleano que determine si es ganador o no.

La clase **Álbum** es una clase abstracta ya que es la clase padre del AlbumExtendido, AlbumTradicional y AlbumWeb, por lo tanto sus clases hijas van a utilizar los atributos y métodos heredados de Album. Fue necesario crear la clase Album ya que notamos que los distintos álbumes compartían ciertas características, tales como un código, una lista de países y un booleano que dice si fue completado; y comportamientos como: pegarFiguritas(), marcarCompletado(), entre otros. De esta manera, se evita repetir o desarrollar el mismo código en distintas clases, e, otras palabras se logra ahorrar y reutilizar código mediante el uso de herencia, uno de los principios más básicos e importantes de la programación orientada a objetos.

A su vez, como se mencionó previamente, las clases hijas utilizan algunos métodos que comparten entre ellas a través de su clase padre. A este comportamiento se lo conoce como polimorfismo, ya que un mismo método puede ser utilizado por varias clases y de distintas maneras.

Además, en la clase Album hicimos que el método tieneCodigoPromocional() sea abstracto ya que tiene que ser sobrescrito en sus clases hijas, es una función que es utilizada en cada una de ellas. De esta forma, utilizamos un concepto visto en clase conocido como sobrescritura. Y además, al ser sobrescrito en cada una de las clases hijas, terminan usando el mismo método pero cada una a su manera, utilizando un concepto conocido como polimorfismo.

Otro concepto trabajado en clases es sobrecarga, el cual fue usado en la clase Participante en el método agregarFiguritaAColeccion(). Este puede recibir dos tipos de argumentos:

1. un sobre de figuritas en forma de lista.
2. una figurita de tipo Figurita.

Es decir, agregarFiguritaAColeccion(List<Figurita> sobre) y agregarFiguritaAColeccion(Figurita figurita).

En cuanto a herramientas de Tecnologías Java se utilizaron: StringBuilder y foreach.

El StringBuilder fue de bastante utilidad para hacer la lista de ganadores en forma de un String y que se vaya actualizando esa cadena con los ganadores. Se utilizó en el método listadoDeGanadores() de la clase AlbumDelMundial. Finalmente, usamos varios foreach por ejemplo en el método participantesQueCompletaronElPais() para recorrer una lista de participantes y otro para recorrer una lista de países. Y de esa manera buscar todos los participantes que completaron algún país.

A continuación se provee una lista a modo resumen para una visualización más práctica:

- Stringbuilder, cuyo uso debe basarse en la necesidad de modificar el string.

Se utilizó en listadoDeGanadores() de la clase AlbumDelMundial.

- Foreach para recorrer las colecciones de Java

Foreach utilizado en participantesQueCompletaronElPais() para recorrer una lista de participantes y otro para recorrer una lista de paises. Y de esa manera buscar todos los participantes que completaron algun pais.

- Herencia

Concepto utilizado en Album y sus Hijos AlbumExtendido, AlbumTradicional y AlbumWeb.

- Polimorfismo

Las clases hijas de Album usan el metodo tieneCodigoPromocinal().

- Sobreescritura

Concepto utilizado en tieneCodigoPromocinal() y en toString()

- Sobrecarga

Concepto utilizado en agregarFiguritaAColeccion() de Figurita donde recibe una lista de figuritas (List<Figurita> sobre) y otra donde solo recibe una Figurita(Figurita figurita).

- Clases abstractas

Concepto utilizado a la hora de implementar Album ya que la misma contiene atributos y métodos que van a ser utilizados por sus clases hijas.

Por último, una de las consignas solicita explicar la complejidad del método:

int buscarFiguritaRepetida(int dni);

En este caso es $O(1)$ ya que al utilizar un hashmap, se encuentra el participante mediante su dni, la key del diccionario, y simplemente se retorna el número de la figurita repetida de la lista de figuritas repetidas que posee el participante. Esto significa que el método tiene una baja complejidad.

IREP

AlbumDelMundial

listaDeParticipantes>0

listaDeParticipantesCompleto>=0

0<codigoGanadorSorteo<11 obtenido de random

Album

codigo >0 : unico

listaDePaises = 32

completado = true, si cada pais tiene completado=true, y si cada pais tiene completado=true

AlbumExtendido

listaTop10 =10

AlbumTradicional

0< numSorteo <11 , obtenido de la clase Fabrica, la cual genera un numero random entre el 1 y el 10 inclusive.

AlbumWeb

CodigoPromocional= true si aun no lo cambi6 por un sobre de figuritas
cuando se le asigna el sobre: CodigoPromocional=false

Figurita

0< numero <13

valorBase >0, obtenido de la clase Fabrica, la cual genera un numero random entre el 1 y el 10 inclusive.

nombreDelPais no null

valorFinal = ranking del pais * valorBase

pegada: en el metodo pegarFiguritas() : si figurita.consultarRepetida()==false y

figurita.estaPegada()== false → marcarComoPegada() → pegada= true

repetida: si pais.tieneEstaFigurita() y figurita.estaPegada()== false → repetida=true

FiguritaTop10

0< numero <11

valorBase >0, obtenido de la clase Fabrica, la cual genera un numero random entre el 1 y el 10 inclusive.

sede: se obtiene de listadoDeMundialesTop10, de la Fábrica, utilizando un random con un rango del 1 al 10 inclusive

tipoBalon: del pais sede se obtienen los finalistas. Se genera random del 1 al 2 inclusive para asignar un puesto(campeón o subcampeón).

si puesto == 1 → tipoBalon = "Oro"

si puesto == 2 → tipoBalon = "Plata"

valorFinal = si tipoBalon = "Oro" → valorFinal = ranking del finalista * valorBase + 20%

si tipoBalon = "Plata" → valorFinal = ranking del finalista * valorBase + 10%

Pais

posiciones = 12

nombre: no null

0<ranking<212 obtenido de la Fabrica

0< listaDePosiciones<13

completado : si listaDePosiciones ==12 → completado =true

Participante

dni>0 , unico

nombre: no null

tipoAlbum no null. tipoAlbum= "AbumExtendido" o "AlbumTradicional" o "AlbumWeb"

coleccionFiguritas: no repetidas

coleccionFiguritasRepetidas: repetidas

premio: si participante.completoElAlbum()= true:

si TipoAlbum() == "Tradicional" → premio= "Pelota"

si TipoAlbum() == "Extendido" → premio= "Pelota Y Viaje"

si TipoAlbum() == "Web" → premio= "Camiseta de la selección"

Tupla.

s1: no null

s2: no null

Diagrama de clases (modificado)

