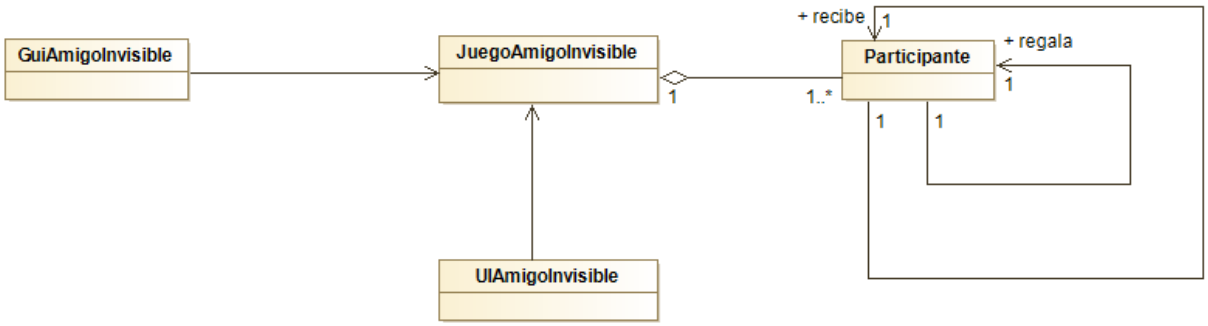
Ejercicio UT5 "*El amigo invisible*" (2021-2022) – 2ª evaluación

Ejercicio.

En este ejercicio vamos a simular el juego del amigo invisible que permite asignar "*un amigo*" a quien regalar en las próximas fiestas.

Ejecuta el fichero ***amigoinvisible.jar*** desde línea de comandos para ver como funciona la aplicación. Lee previamente el ***Anexo – Cómo ejecutar un jar desde línea de comandos con parte gráfica JavaFX***".

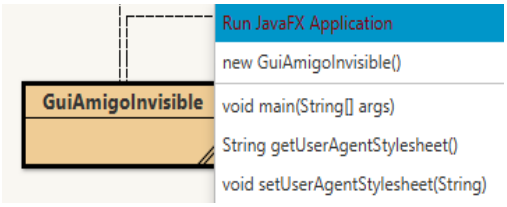


Has de completar la clase JuegoAmigoInvisible. El resto se dan completas.

La clase UIAmigoInvisible representa un interfaz de texto (por consola) para interactuar con el usuario.

La clase GuiAmigoInvisible representa un interfaz gráfico para interactuar con el usuario.

Puedes usar cualquiera de las dos para lanzar la aplicación una vez completada.

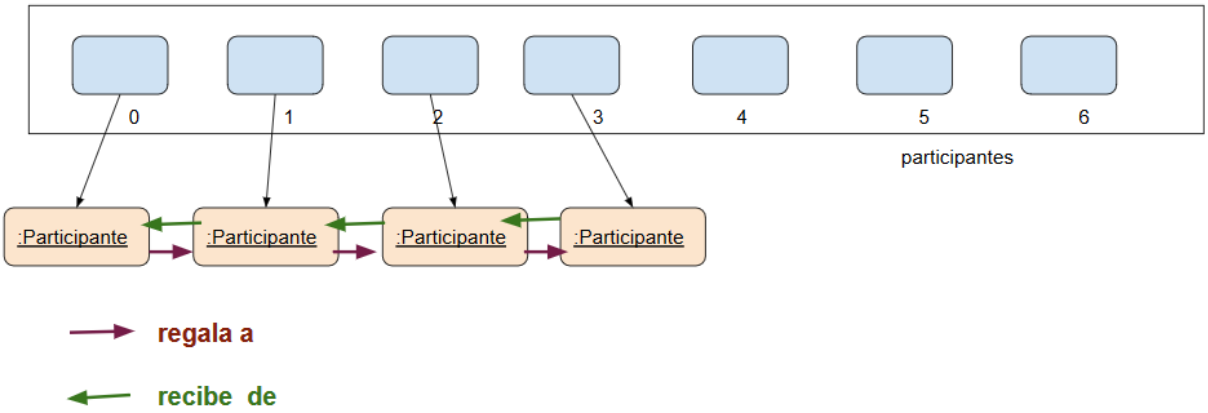


La clase Participante representa a un participante en el juego y se da también completa (léela y asegúrate de que entiendes bien como está modelada).

En el juego del amigo invisible hay una bolsa de participantes entre los cuales se realizará la asignación del *amigo*. Cada participante regala a otro participante y recibe a su vez de un participante. Un participante no puede regalarse a sí mismo.

Aunque hay varias maneras de resolver este problema nosotros lo haremos así:

- utilizaremos un array unidimensional de objetos Participante que simulará la bolsa de participantes
- *revolveremos los participantes en la bolsa*, es decir, moveremos aleatoriamente los elementos de este array antes de hacer el reparto
- una vez *revueltos* asignaremos los amigos de la siguiente forma
  - cada participante regala al que le sigue en el array
  - cada participante será regalado por el que le precede en el array
  - el siguiente al último es el primero
  - el anterior al primero es el último



¿Qué funcionalidad ha de tener la clase JuegoAmigoInvisible?

- ✗ hay que poder añadir participantes
- ✗ ver si está ya un participante añadido
- ✗ simular que se mueven los participantes antes de hacer el reparto de amigos
- ✗ repartir los amigos
- ✗ dado un participante conocer a quién regala y de quién recibe
- ✗ borrar un participante del juego
- ✗ borrar todos
- ✗ ver el resultado de la asignación de amigos
- ✗ .....

Completa los métodos de la clase JuegoAmigoInvisible:

a) el constructor que crea la bolsa de participante al tamaño máximo

b) add() – añade un nuevo participante a la bolsa al final de la misma (si no está ya completa)

c) totalParticipantes() – devuelve la cantidad de participantes reales en el juego

d) Participante[] getParticipantes() – devuelve una copia con el número real de participantes

e) buscarParticipante() - dado un nombre de participante devuelve la posición donde se encuentra o -1 si no está

   - **Nota** – para comparar cadenas haremos *cadena1.equalsIgnoreCase(cadena2)* que devuelve *true* si *cadena1* es igual a *cadena2* (sin importar mayúsculas o minúsculas)

f) boolean estaParticipante() – dado un nombre de participante indica si está añadido ya en la bolsa (usa el método anterior)

g) datosDe() - dado un nombre de participante devuelve el objeto asociado a él (para disponer de la información de ese participante)

h) toString() - representación textual de todos los participantes

i) revolverParticipantes() - *revuelve* (*baraja*) los participantes. Se puede hacer de forma sencilla intercambiando un nº determinado de veces pares de elementos del array *participantes* de posiciones aleatorias

j) reset() - elimina todas las asignaciones de emisor y receptor de todos los participantes.

k) asignarAmigos() - asigna los amigos invisibles previo *barajeo* de la bolsa inicial

l) borrarParticipante() - dado un participante lo borra. Se devuelve *true* si la operación ha tenido éxito (estaba el participante) o *false* si no ha habido éxito (el participante no existía). Cada vez que se borra un participante hay que eliminar todas las asignaciones de emisor y receptor.

m) borrarTodos() - borra todos los participantes

Prueba esta clase con el interfaz gráfico desde BlueJ.

Prueba esta clase con el interfaz de consola haciendo la llamada desde línea de comandos.