Quiz 31/7/2025 Versión A

Dado el material entregado: Escriba en la carpeta `src` una clase `Boda` que sea usada por la clase `boda.test.TestBoda` para que esta genere los resultados mostrados y explicados adelante.
Tareas:

a. Escriba en `src` `Boda.java` que permita que `TestBoda` funcione. Estos son los `imports` de esa `TestBoda`:

```
import java.util.*;
import boda.model.Boda;
import static boda.model.Boda.*;
```

Nota: Ud. no recibe el fuente de `TestBoda` solo el `.class`.

b. Compile su clase `Boda.java` usando una consola ubicada en su carpeta de trabajo según se muestra:

```
javac -d build src\Boda.java
```

c. Una vez corregido cualquier error de compilación, ejecute su clase `boda.test.TestBoda`:

```
java -cp build boda.test.TestBoda
```

d. Compare con la salida esperada (se adjunta esa salida en archivo con la salida en `output.txt`). Salida (truncada en este documento):

```
*** Casting ***
Pedro:
        .sin Rol
Luis:
        .como Pareja
        .como Bailante
Refugios:
        .sin Rol
María:
        .como Pareja
        .como Bailante
Ana:
        .como Discursante
        .como Testificante

*** Probando Evento; Casamiento ***
Pedro llora
Luis da el sí
Refugios llora
María da el sí

*** Probando Evento: Firma ***
```

Esa salida es el producto del siguiente código en `boda.test.TetBoda`:

```
static void testAll(){
    //Actores y Roles
    var ana = new Mujer("Ana");
    var luis = new Varon("Luis");
    var maria = new Mujer("María");
    var pedro = new Varon("Pedro");
    var refugios = new Varon("Refugios");

    Boda.registrar(pedro, refugios);
    Boda.registrar(ana, new DaDiscurso(), new FirmaComoTestigo());
    Boda.registrar(Arrays.asList(luis, maria),
                   new Pareja(), new BailaVals());
    Boda.participantes();

    //Eventos

    Boda.atencion("Probando Evento; Casamiento", Evento.JURAMENTO);

    Boda.atencion("Probando Evento: Firma", Evento.FIRMA);

    Boda.atencion("Probando Evento: Brindis", Evento.BRINDIS);

    Boda.atencion("Probando Evento: Vals", Evento.VALS);

}
```

Las clases `Boda`, `Mujer`, `Varon` y los métodos `Boda::registrar` y `Boda::atencion` las debe proveer su `Boda.java`. El método `Boda.participantes` imprime cada participante y sus roles.

```
*** Casting ***
Pedro:
        .sin Rol
Luis:
        .como Pareja
        .como Bailante
Refugios:
        .sin Rol
María:
        .como Pareja
        .como Bailante
Ana:
        .como Discursante
        .como Testificante
```

El método `Boda::atencion` imprime un mensaje y notifica a los participantes de un evento para que estos reaccionen.
Por ejemplo: la llamada:

```
Boda.atencion("Probando Evento; Casamiento", Evento.JURAMENTO);
```

Produce:

```
*** Probando Evento; Casamiento ***
Pedro llora
Luis da el sí
Refugios llora
María da el sí
```

Ante un evento de `JURAMENTO` se ven las reacciones de los participantes a quiés ese evento llama a atención.
Note que `Pedro` no tenía un rol asignado. Cuando una persona no se le ha registrado ningún rol, se asume uno por defecto llamado `None` que hace que la persona llore o ría aleatoriamente (50% c/u). Hay un rol `Pareja` (para que la persona sea novio(a) en la boda). Este se asocia al evento `JURAMENTO`. Por ejemplo, `Luis` tiene rol `Pareja`, reacciona dando el sí. Lo mismo con `María`.

Hay un rol `DaDiscurso` que se asocia al evento `BRINDIS`. Hay `BailaVals` que reacciona al evento `VALS`. Y el rol `FirmaComoTestigo` que reacciona al evento `FIRMA`. La reacción ante un evento de una persona es simplemente imprimir un mensaje.
Por ejemplo:
Si una persona llamada `Ana`, tiene el rol `daDiscurso` ante el evento `BRINDIS` simplemente se imprime:

```
Ana da un emotivo discurso.
```

El método `Boda.registrar` permite tres tipos de registros en la boda: registrar uno o varias personas de la `Boda`. O registrarle uno o varios roles a una persona. O registrarles unos roles a todas las personas de una lista de personas.
Por ejemplo:

```
Boda.registrar(pedro, refugios);
```

Acá se registran `pedro` y `refugios` dos personas como participantes (sin rol definido aún). Tendrán rol `None`.
Acá:

```
Boda.registrar(ana, new DaDiscurso(), new FirmaComoTestigo());
```

se registra a `ana` con roles `DaDiscurso` y `FirmaComoTestigo`. Pueden ser más de dos o solo uno.
Y con:

```
Boda.registrar(Arrays.asList(luis, maria),
               new Pareja(), new BailaVals());
```

se registran `luis` y `maría` ambos como `Pareja` y como `BailaVals`.

Indicaciones

- En una **hoja en blanco** (la hoja de revisión) escriba en la parte superior (ud. pone su horario 10am, 1pm o 6pm).

        - `EIF400-II-2025 Quiz 1 31-julio Boda HORARIO:`

        - Pone los **nombres completos** de los autores y sus **cédulas**
        - En esa hoja se pondrá la nota y observaciones por el profesor. **Se entre una sola hoja por grupo**. Me la entrega al inicio del quiz.
    - Al finalizar **un único miembro de cada grupo** de trabajo me envía un correo a mi correo desde su correo de la una así (`HH` su horario)
        - `Asunto: EIF400-II-2025 Quiz 1 31-julio Boda HH`
        - `Adunto: Boda.java`

    - Su archivo Boda.java tiene al inicio un comentario con los nombres de los autores. Ejemplo. Ponga sus datos y horario (10am, 1pm, 6pm)

```
/**
Quiz #1 Boja Horario: 10am
@since 31/07/2025
@author Juan Perez 11111
@author Ana Kareinina 22222
@author Jafet Lonis 33333
@author Ronald Diño 444444
*/
```

    - No puede usar IA.

    - Compartir este examen y/o sus respuestas con otros fuera del grupo oficial (mismo horario u otro horario) se considerará copia y/o plagio. Rige a partir de leído este documento.

    - Evaluación:

        - Entrega la hoja en de revisión (reciben cero si no la entregan o no cumple)
        - 65% Si su clase `Boda.java` **funciona** y `TestBoda` produce **exactamente** la salida esperada (`output.txt`). Tiene el comentario al inicio.
        - 35% Criterio del profesor sobre el modelo OOP implementado