

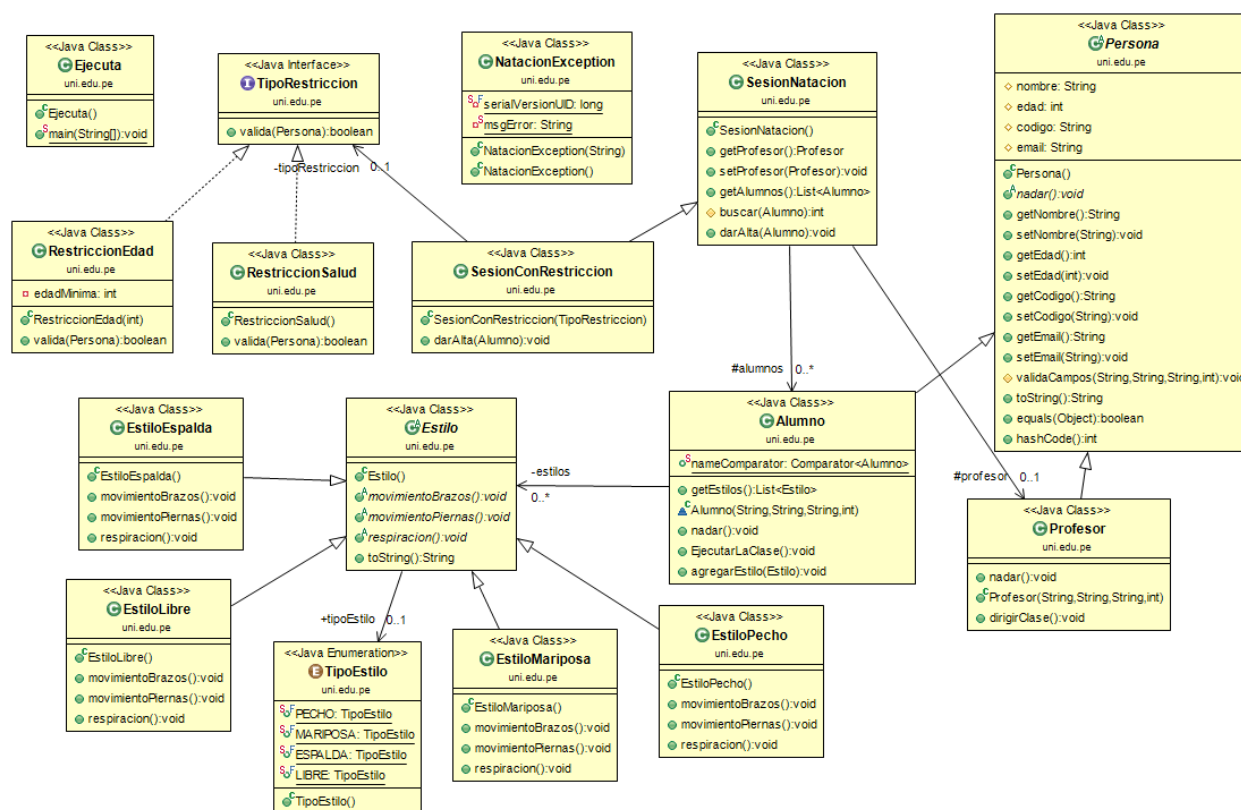
## PRACTICA CALIFICADA II

Curso : Programación orientada a objetos  
Código : SI302V  
Alumno :

CICLO: 2021-I

### Caso sesión de Natación

Luis es un profesor de natación que espera llevar un control de las sesiones de natación de sus alumnos, para ello ha elaborado el siguiente diagrama:



Se desea desarrollar una aplicación para la gestión de las sesiones de natación, se tiene en cuenta las siguientes consideraciones:

- (1 punto) La clase **NatacionException** es una excepción no checkeada que se utilizará para tratar las distintas situaciones excepcionales.
- (2.5 puntos) La clase **Estilo** es una clase abstracta que contiene un atributo **tipoEstilo**, el cual es un enum con los siguientes valores (**PECHO, MARIPOSA, ESPALDA Y LIBRE**). Adicionalmente, posee los siguientes métodos abstractos: **movimientoBrazos()**, **movimientoPiernas()** y **respiración()**;

- Las clases: **EstiloPecho**, **EstiloMariposa**, **EstiloEspalda** y **EstiloLibre**; son clases concretas, hijas de la clase **Estilo**, que contienen un constructor por defecto que se encarga de inicializar el atributo **tipoEstilo** (según el valor que corresponda). Dichas clases además implementa el comportamiento de los métodos abstractos imprimiendo un mensaje según sea el caso:

**respiracion():**

- **Libre:** Respiración Lateral
- **Espalda:** respiración boca arriba
- **Mariposa y Pecho:** respiración Plano frontal

**movimientoPiernas():**

- **Libre, Espalda y Mariposa:** Batido alternado de piernas
- **Pecho:** Patada de Rana

**movimientoBrazos()**

- **Libre y Espalda:** Alternancia de Brazos
- **Mariposa:** Movimiento simultaneo de brazos
- **Pecho:** Brazada de Pecho.

3. (5 puntos) La clase **Persona** es una clase abstracta que contendrá la información correspondiente a edad (int), nombre (String), codigo (String) y el email (String). También se contará con el método abstracto nadar.

- La clase **Profesor** es una clase concreta, hija de la clase **Persona**, que tiene como el método **void dirigirClase()** que imprime en pantalla un mensaje. Para crear un profesor se tiene que indicar nombre, código, edad y email.
- La clase **Alumno** es una clase concreta, hija de la clase **Persona**, que tiene como atributo una relación de los estilos que va a llevar ese alumno **estilos** y los métodos:
  - **public void ejecutarLaClase** (que sólo imprime en pantalla un mensaje)
  - **public agregarEstilo(Estilo e)** que permite agregar un nuevo estilo a la relación de estilos a ejecutar ese día. Tener presente que se debe validar que un alumno debe llevar al menos un estilo, y puede realizar máximo 3 estilos en la sesión, pero que si alguno de esos estilos seleccionados es Mariposa podrá ejecutar máximo 2 estilos (por la dificultad que supone este estilo).
- Tanto para el profesor, como para el alumno, si el nombre, el código o el email son nulos o bien la edad no es un entero positivo se lanzará una excepción del tipo **NatacionException**.
- La representación textual (toString) de una persona viene dado por su nombre, codigo, edad, y correo electrónico (entre corchetes) separados por espacio en blanco. Por ejemplo:  
Nestor Audante 20031425G 25 [naudante@contoso.com]
- Se pide encapsular los atributos del Profesor y del Alumno.
- Se pide garantizar que 2 objetos descendientes de la clase **Persona** (Profesor o Alumno) sean iguales si coinciden en su codigo y email.

4. (3 puntos) Una Sesión de Natación "**SesiónNatación**" está compuesta por un profesor y una relación de Alumnos "**alumnos**"(máximo 10), la cual será representada con un ArrayList.

- Definir un método **protected int buscar**(Alumno) que determina si el alumno que ha sido pasado como argumento se encuentra en el listado de alumnos "**alumnos**" y devuelve la posición en la que se encuentra, caso contrario devuelve -1.
  - Definir el método para dar de alta en la sesión a un nuevo alumno (**void darAlta**(Alumno a)), que se pasa como argumento. Si la persona que se desea añadir ya se encuentra en el listado no se hará nada. Se lanzará una excepción del tipo **NatacionException** si el objeto a es nulo. Puedes usar el método buscar que implementamos en líneas anteriores.
5. (1 punto) La interfaz "**TipoRestricción**" especifica un método **public boolean valida**(Persona p) que sirve para conocer si la persona p cumple con las restricciones establecidas por la piscina para llevar una sesión.
  6. (2.5 puntos) La clase **SesionConRestriccion** se comporta como la clase **SesionNatacion**, con la diferencia de que los alumnos deben cumplir una serie de restricciones antes de acudir a esta sesión. Cada objeto de esta clase tiene una variable de instancia restricción (cuyo tipo es la interfaz **TipoRestriccion**) que servirá para implementar las restricciones pertinentes. Se deben implementar los métodos necesarios para:
    - Construir un objeto de la clase, dado el objeto de la clase **TipoRestriccion**
    - Redefinir el método **public void darAlta**(Persona p) para que añada un nuevo alumno, sólo si cumple la restricción previamente definida. En caso p sea nulo debe enviarse una excepción del tipo **NatacionException**.
  7. (2 puntos) Se va a implementar la lógica sólo de la restricción de edad: La clase **RestriccionEdad** implementa la interfaz **TipoRestriccion**. Tiene una variable de instancia llamada **edadMinima(int)**. Dispondrá de los métodos siguientes:
    - Un constructor al que se le pasará como argumento un valor entero con la edad e inicializará edad mínima que un usuario debe tener para poder asistir a una sesión de natación. Recuérdese que si se recibe una edad negativa se lanzará una **NatacionException**
    - El método **boolean valida**(Persona p) para que cualquier persona que tenga una edad menor que la señalada por **edadMinima** no sea válida.
  8. (2.5 puntos) Ahora, cree una clase ejecutable que haga lo siguiente:
    - Genere una Sesión (En base a la **SesionConRestricción**) en la que ingrese la información de un profesor y "N" Alumnos (Cada uno con sus estilos). El valor de N será un valor decidido por usted y los valores de los alumnos y sus estilos serán seteados en duro.
    - Ordene la relación de alumnos (en base a su Nombre), muestre los estilos que llevó en la sesión y ejecute sus métodos para respirar y mover brazos y piernas.
  9. (0.5) Suba la carpeta con el código fuente y el enunciado de esta práctica en un directorio privado en github llamado practica-2 y brinde acceso al profesor del curso.