

Trabajo Práctico 2: Tipos de datos abstractos

Ejercicio 1

Implementar el TDA "Propiedad" que modela un inmueble, con una estructura definida por los siguientes componentes:

- Calle
- Número
- Localidad
- Año de construcción
- Cantidad de ambientes

Implementar las siguientes operaciones:

- Constructor: Debe incluir las validaciones necesarias, teniendo en cuenta que solo se almacenan propiedades construidas luego de 1870.
- `_str_`: Al usar la función *print* con una variable del tipo propiedad debe mostrar: **'calle' 'numero' ('localidad')**.
- `mismaLocalidad`: Operación que recibe dos propiedades y retorna *True* si estan en la misma localidad y *False* en caso contrario.
- `mayorNumeración`: Operación que recibe dos propiedades y si estan en la misma calle, retorna la que posee mayor numeración. Si estan calles diferentes debe lanzar una excepción.
- `calculaImpuestoARBA`: Operación que retorna el porcentaje de impuesto inmobiliario de una propiedad, según la siguiente regla:
 - Propiedades entre 1870 y 1949:
 - Entre 1 y 3 ambientes: 5% de impuesto
 - Entre 4 y 6 ambientes: 10% de impuesto
 - Más de 6 ambientes: 25 % de impuesto
 - Propiedades desde 1950 hasta la actualidad:
 - Entre 1 y 5 ambientes: 5% de impuesto
 - Más de 5 ambientes: 35 % de impuesto

Ejercicio 2

Implementar el TDA "Quiniela" que modela un juego de quiniela con dos números premiados. La estructura contiene:

- Primer número premiado
- Segundo número premiado
- Multiplicador (cuánto se paga por cada peso apostado)

Implementar las siguientes operaciones:

- Constructor: Debe incluir las validaciones necesarias, teniendo en cuenta que los números que participan se encuentran entre 0 y 999.
- `_str_`: Al usar la función *print* con una variable del tipo quiniela debe mostrar:
Primer número ganador: 'numero' - Segundo número ganador: 'numero' - Paga: 'multiplicador'X.
- `esNumeroGanador`: Operación que recibe un número por parámetros y retorna *True* si el número resulto ganador o *False* en caso contrario.
- `importeAPagar`: Operación que recibe un número y el monto apostado por parámetros y retorna el importe a pagar si la apuesta es ganadora o 0 en caso contrario. Si el número es el primer premio, se paga 'mutiplicador' por cada peso apostado, si es el segundo premio se paga la mitad. Solo se aceptan apuestas hasta \$1000.
- `premiadosCercanos`: Operación que retorna *True* si los números premiados están a menos de 10 números de distancia y *False* en caso contrario.

Ejercicio 3

Implementar el TDA "Cuenta" que modela una cuenta bancaria, la estructura de datos esta compuesta por los siguientes componentes:

- Número de cuenta
- DNI del titular
- Saldo de cuenta actual
- Interés anual

Implementar las siguientes operaciones:

- Constructor: Debe incluir las validaciones necesarias.
- `_str_`: Al usar la función *print* con una variable del tipo cuenta debe mostrar: **Cuenta Nro: 'numero' - Titular: 'dni' (\$'saldo').**
- `actualizarSaldo`: Operación que actualiza el saldo de la cuenta aplicándole el interés diario (interés anual dividido entre 365).
- `ingresarDinero`: Operación que recibe un número e ingresa esa cantidad en la cuenta.
- `retirarDinero`: Operación que recibe un número y extrae esa cantidad de la cuenta (si hay saldo disponible), sino debe lanzar una excepción.

Ejercicio 4

Implementar el TDA "Tiempo" que modela una duracion en horas, minutos y segundos.

Se deben implementar las siguientes operaciones:

- Constructor: Debe incluir las validaciones necesarias, la hora debe ser un número positivo y los minutos y segundos deben ser números positivos entre 0 y 59.
- `_str_`: Al usar la función *print* con una variable del tipo tiempo debe mostrar: **'horas': 'minutos': 'segundos'.**

- tiempoASegundos: Operación que toma una variable de tipo tiempo y retorna la cantidad en segundos.
- tiemposDesdeSegundos: Operación que recibe un tiempo en segundos como parámetro y retorna una variable de tipo tiempo, en horas minutos y segundos.
- mayorDuracion: Operación que recibe dos variables de tipo tiempo y retorna la de mayor duración.

Ejercicio 5

Las plataformas de música online como **YouTube** y **Spotify** almacenan la información asociada a las canciones en estructuras de datos complejas para hacer las búsquedas de manera eficiente. Para esto se deben modelar las canciones. Implementar el TDA

"Cancion" con los siguientes componentes:

- Nombre
- Artista
- Duración
- Género musical (6 posibles: Rock, Jazz, Blues, Funk, Reggae y Rap).
- Año de edición
- Número de likes

Implementar las siguientes operaciones:

- Constructor: Debe incluir las validaciones necesarias.
- `_str_`: Al usar la función *print* con una variable del tipo canción debe mostrar: **'nombre' - 'artista' ('duracion')**.
- mayorDuracion: Operación que recibe dos canciones por parámetros y retorna la de mayor duración.
- agregaLikes: Operación que recibe un número e incrementa la cantidad de likes de la canción en ese número.
- masVotada: Operación que recibe dos canciones y si son del mismo artista y del mismo género musical, retorna la que tiene mayor cantidad de likes. En caso contrario debe lanzar una excepción.
-