

Este boletín de ejercicios está orientado a practicar con

- programación orientada a objetos
1. Queremos implementar una clase para gestionar un juego de Pokemon con las siguientes características:
 - Los atributos base que manejaremos serán código, nombre y tipo
 - Sólo trabajaremos con pokemon de primera generación por lo que el código estará entre el 1 y el 151, ambos incluidos
 - Los posibles tipos son Normal, Agua, Fuego, Planta, Volador, Lucha, Veneno, Eléctrico, Tierra, Roca, Psíquico, Hielo, Bicho, Fantasma y Dragón.
 - Cada pokemon debe de ser de un tipo pero podría ser de dos. Nunca mas
 - No necesitamos setters (ya que un pokemon una vez creado no puede modificar sus características) pero si getters apropiados para todas ellas
 - Además, crearemos un método que se llame evolución que permitirá que un pokemon evolucione en otro diferente. Para ello si un pokemon puede evolucionar en otro debe de tener de alguna forma una referencia al pokemon en el que evoluciona.
 2. Queremos implementar una clase para gestionar nuestra colección de mangas con las siguientes características:
 - Por cada manga guardaremos el nombre del mangaka (autor) el título de la colección (en japonés, obligatorio y en español, opcional), el género principal (shonen, shojo, seinen, josei, kodomo, yuri, spokon, isekai y hentai) y el último número publicado en la colección. Crea getters para todos ellos y setter para el título en castellano (por si originalmente no lo sabemos y luego lo queremos añadir) y para el número por el que va la colección.
 - Queremos, además, poder actualizar los números que tenemos y saber que números nos faltan. Para ello crearemos dos métodos: uno que nos permitirá introducir los números que vamos comprando (permitiendo una entrada variable de argumentos para cuando compramos mas de uno a la vez) y otro que nos diga que números nos faltan para completar la colección.
 - Si cuando introducimos los números que compramos resulta que ya tenemos alguno de ellos repetido debería de advertirnos
 - También necesitaremos un método que nos permita eliminar un número (lo hemos perdido, etc.). Si tratamos de eliminar un número que no tenemos debería de advertírsenos