



Programación Java

▶ Introducción al lenguaje Java

// Práctica integradora - Programación orientada a Objetos

Objetivo

El objetivo de esta guía práctica es que podamos afianzar y profundizar los conceptos de programación orientada o objetos aplicados a Java. Para esto vamos a plantear una serie de ejercicios simples e incrementales (ya que vamos a ir trabajando y agregando lógica a las clases que tenemos que construir), lo que nos permitirá repasar los temas que estudiamos.

¿Are you ready? 😎👍



Ejercicio 1

Creá una clase Persona, la cual tendrá los siguientes atributos: nombre, edad, dni (en este caso vamos a representarlo como una cadena de caracteres), peso y altura ¿Qué tipo de dato le asignarías a las variables de instancia? ¿Cómo sería la estructura básica de tu clase?



Ejercicio 2

Vamos a crear diferentes constructores en la clase Persona, uno sin parámetros, el segundo debe recibir como parámetro nombre, edad y dni; por último creamos un tercero que reciba todos los atributos de la clase como parámetro.



Ejercicio 3

Creá una clase nueva llamada Main, donde declares un método main como te enseñamos anteriormente. Esto nos permitirá ejecutar nuestra aplicación.



Ejercicio 4

En la clase Main que acabamos de crear, dentro del método main() te pedimos que crees un objeto de tipo Persona por cada constructor que hayamos definido en la clase, recuerda poner un nombre significativo a las variables donde vas a asignar cada objeto. ¿Cómo lo harías? A continuación vamos a crear otro objeto de tipo persona y vamos a construirlo pasando solamente un valor para el nombre y otro para la edad en el constructor. ¿Es esto posible? ¿Qué sucede si tratamos de hacer esto?



Ejercicio 5

En la clase Persona implementaremos los siguientes métodos: `calcularIMC()`, la fórmula para calcularlo es: $\text{peso}/(\text{altura}^2)$ - (peso expresado en kg y altura en mts), si este cálculo devuelve un valor menor que 20, la función debe retornar -1, si devuelve un número entre 20 y 25 **inclusive para los dos valores**, el método debe retornar un 0, por último, si devuelve un número mayor que 25 debe retornar un 1. Una vez creado el método anterior, agreguemos el método `esMayorDeEdad()` el cual debe retornar un valor booleano, teniendo en cuenta que la mayoría de edad será considerada en este caso, a partir de los 18 años. Finalmente agregar un método `toString()` que va a devolver toda la información de la persona.

¡Recuerda! Puedes ayudarte de los métodos de la clase `java.lang.Math` para calcular la potencia.



Ejercicio 6

Desde la clase Main vamos a calcular el IMC de la última persona que creamos (la que creamos correctamente mediante el constructor que recibe todos los atributos como parámetro). También vamos a averiguar si es mayor de edad o no; ten en cuenta que en ambos casos, dependiendo de los resultados retornados por los métodos, debes imprimir un mensaje acorde para el usuario. Finalmente queremos mostrar todos los datos de esa persona imprimiendo dicha información por consola. El formato en que vas a mostrar los datos y los mensajes quedan a tu criterio, pero debe ser legible y descriptivo para quien ve la salida del programa.

Referencias:

Índice de masa corporal (IMC)	Nivel de peso
Por debajo de 20	Bajo peso
Entre 20 y 25 inclusive	Peso saludable
Mayor de 25	Sobrepeso