

Trabajo Práctico N°2 Módulo 2 - Clase 3 y 4

Métodos de clases y métodos especiales.

Ejercicio N° 1

Vamos a realizar una clase llamada Raíces, donde representaremos los valores de una ecuación de 2º grado. Tendremos los 3 coeficientes como atributos, llamémosles a, b y c. Hay que insertar estos 3 valores para construir el objeto.

Las operaciones que se podrán hacer son las siguientes:

- obtenerRaices(): imprime las 2 posibles soluciones
- obtenerRaiz(): imprime una única raíz, que será cuando solo tenga una solución posible.
- getDiscriminante(): devuelve el valor del discriminante (double), el discriminante tiene la siguiente fórmula, $(b^2)-4ac$
- tieneRaices(): devuelve un booleano indicando si tiene dos soluciones, para que esto ocurra, el discriminante debe ser mayor o igual que 0.
- tieneRaiz(): devuelve un booleano indicando si tiene una única solución, para que esto ocurra, el discriminante debe ser igual que 0.
- calcular(): mostrará por consola las posibles soluciones que tiene nuestra ecuación, en caso de no existir solución, mostrarlo también.

Fórmula ecuación 2º grado: $(-b \pm \sqrt{(b^2)-(4ac)})/(2*a)$

Solo varía el signo delante de -b

Introducción a la Programación en Java

Ejercicio N° 2

Crear una clase llamada Cuenta que tendrá los atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:

- Un constructor;
- mostrar(): Muestra los datos de la cuenta;
- ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada;
- retirar(cantidad): se retira una cantidad a la cuenta. La cuenta puede estar en números rojos.

Ejercicio N° 3

Usando la clase persona del ejercicio 1 crear una clase empleado que deberá tener el atributo sueldo básico. Implementar los métodos:

- calculoSuelto(): este se calcula como la cantidad de horas trabajadas por el sueldo básico
- verEmpleado(): visualiza los datos del empleado.

Ejercicio N° 4

Crear una clase Productos con atributos: Nombre, precios, vencimiento, con 2 tipos de productos perecederos y no perecederos.

- Crear sus constructores getters y setters.

Introducción a la Programación en Java

- crear una función llamada calcular: en producto simplemente seria calcular precio por cantidad, en perecederos el precio se reducirá según los días a caducar, si le queda 1 día se reducirá 4 veces el precio, si le quedan 2 días se reducirá el precio 3 veces, si le quedan 3 días se reducirá la mitad el precio. Crear una clase ejecutable y crear un array de productos y muestra el precio total de vender 5 productos de cada uno.

Ejercicio N° 5

Crear una agenda telefónica de contactos:

Cada contacto tiene nombre y número de teléfono, un contacto es igual a otro cuando sus nombres son iguales Una agenda de contactos está formada por una cantidad n de contactos. Podemos definir nosotros la cantidad o por defecto serán 10.

Los métodos de la agenda serán los siguientes:

- añadirContacto(Contacto c): añade un contacto a la agenda, si no se pueden meter más a la agenda se indicará por pantalla. No se pueden duplicar contactos existentes. -existeContacto(Contacto c): indica si el contacto pasado existe o no.
- listarContactos(): lista todos los contactos.
- buscaContacto(String nombre): busca el contacto por el nombre y muestra su teléfono
- eliminarContacto(String nombre): elimina el contacto y muestra un mensaje de contacto eliminado.
- agendaLlena(): indica si la agenda está llena

Introducción a la Programación en Java

- `contactosLibres()`: indica cuántos contactos libres quedan
- Crea un menú con opciones por consola para probar todas estas funcionalidades.

Ejercicio N° 6

Crear una Clase Taxi que contenga los siguientes atributos: id, número Taxi, tipoDeTarifa (1 o 2), precio de bajada de bandera(

58o

70). Crear un objeto taxi utilizando un constructor a su elección. Dentro de la clase crear los siguientes métodos:

- crear un método `cambiarTarifa` que permita cambiar el tipo de tarifa del taxi, si vale 1 debe cambiar a 2 y el precio debe ser 70 y si el tipo de tarifa es 2 debe cambiar a 1 y el precio a 58, cuando realice el cambio mostrar un mensaje "cambio de tarifa exitoso".
- crear un método `calcularTotalAPagar` que permita ingresar la cantidad de metros recorridos, si el tipo de tarifa es 1 se debe cobrar el precio de bajada de bandera apenas sube la persona y por cada metro se le va a cobrar 5 mas, si el tipo de tarifa es 2 se debe cobrar el precio de bajada de bandera apenas sube la persona y por cada metro se le cobra 8 más. El método debe devolver el monto que se le debe pagar al taxi.

Los métodos deben ser probados en el objeto creado ingresando valores a su elección y mostrando los resultados.

Ejercicio N° 7

Crear una Clase Producto que contenga los siguientes atributos: id, nombre, categoría, precio, stock. Crear un objeto Producto utilizando un constructor a su elección. Dentro de la clase crear los siguientes métodos:

- Crear un método `validarStock` este método debe devolver un `true` si es que tiene stock disponible, sino lo tienen debe devolver `false`.

Introducción a la Programación en Java

- Crear un metodo vender este método debe recibir una categoría y la unidades que se desean vender, el método debe validar usando el método anterior si hay stock antes de realizar la venta, si se puede vender debemos comprobar que la categoría ingresada sea igual a la del producto si es asi recien se puede realizar la venta por lo cual debe mostrar el total a pagar por la venta y actualizar el stock del producto, en cualquiera de los otros casos que no se pueda vender mostrar un mensaje de error a su elección.
- Crear un método comprar este método debe recibir como parámetro las cantidad de producto y la categoría, si la categoría coincide con la del producto actual y el stock está por debajo de 100 unidades esa cantidad de producto debe ser sumada al stock actual, mostrar un mensaje con todos los datos del objeto. En caso contrario mostrar un mensaje "no se pudo agregar la compra". Los métodos deben ser probados en el objeto creado ingresando valores a su elección y mostrando los resultados. Cuando quiera probar el método validar en la consola debe mostrar en vez de true "si tiene stock", y en vez de false "no tiene stock".