

Ejercicio 1

Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Crea dos constructores que cumpla lo anterior.

Crea sus métodos constructor, get, set y toString.

Tendrá dos métodos especiales:

- ingresar(double cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- retirar(double cantidad): se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0.

Ejercicio 2

Haz una clase llamada Persona que siga las siguientes condiciones:

- Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo.
- Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo sera hombre por defecto, usa una constante para ello.
- Se implantaran varios constructores:
 - Un constructor por defecto.
 - Un constructor con el nombre, edad y sexo, el resto por defecto.
 - Un constructor con todos los atributos como parámetro.

Los métodos que se implementaran son:

- `calcularIMC()`: calcula si la persona esta en su peso ideal (peso en kg/(altura^2 en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que esta por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1.
- `esMayorDeEdad()`: indica si es mayor de edad, devuelve un booleano.
- `comprobarSexo(char sexo)`: comprueba que el sexo introducido es correcto. Si no es correcto, sera H. No sera visible al exterior.
- `toString()`: devuelve toda la información del objeto.
- `generaDNI()`: genera un número aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método sera invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
- Métodos set de cada parámetro, excepto de DNI.

Ahora, crea una clase main que haga lo siguiente:

1. Pide por teclado el nombre, la edad, sexo, peso y altura.
2. Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
3. Para cada objeto, deberá comprobar si esta en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
4. Indicar para cada objeto si es mayor de edad.
5. Por último, mostrar la información de cada objeto.

Ejercicio 3

Haz una clase llamada Password que siga las siguientes condiciones:

- Que tenga los atributos longitud y contraseña . Por defecto, la longitud sera de 8.
- Los constructores serán los siguiente:
 - Un constructor por defecto.
 - Un constructor con la longitud que nosotros le pasemos. Generara una contraseña aleatoria con esa longitud.

Los métodos que implementa serán:

- esFuerte(): devuelve un booleano si es fuerte o no, para que sea fuerte debe tener mas de 2 mayúsculas, mas de 1 minúscula y mas de 5 números.
- generarPassword(): genera la contraseña del objeto con la longitud que tenga.
- Método get para contraseña y longitud.
- Método set para longitud.

Ahora, crea una clase clase main:

- Crea un array de Passwords con el tamaño que tu le indiques por teclado.
- Crea un bucle que cree un objeto para cada posición del array.
- Indica también por teclado la longitud de los Passwords (antes de bucle).
- Crea otro array de booleanos donde se almacene si el password del array de Password es o no fuerte (usa el bucle anterior).
- Al final, muestra la contraseña y si es o no fuerte (usa el bucle anterior). Usa este simple formato:

- contraseña1 valor_booleano1
- contraseña2 valor_bololeano2

Ejercicio 4

Se requiere desarrollar un programa que modele una tienda de computadores.

Los computadores de la tienda tienen los siguientes atributos:

- Marca del computador.
- Cantidad de memoria.
- Sistema operativo.
- Precio del computador.

La tienda posee los siguientes atributos:

- Nombre de la tienda.
- Propietario de la tienda.
- Identificador tributario de la tienda.
- Array de 5 computadores

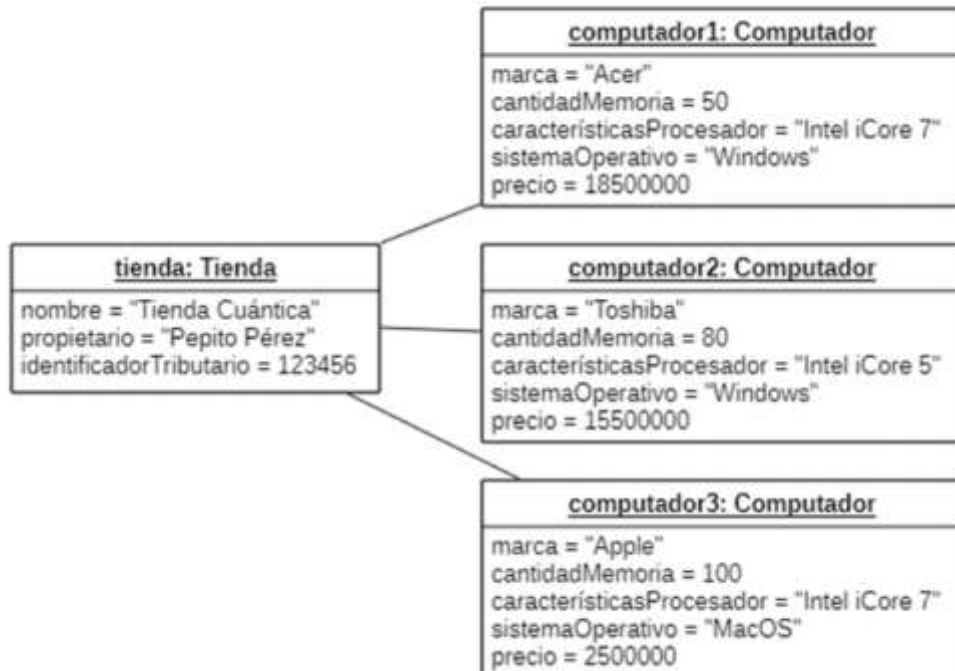
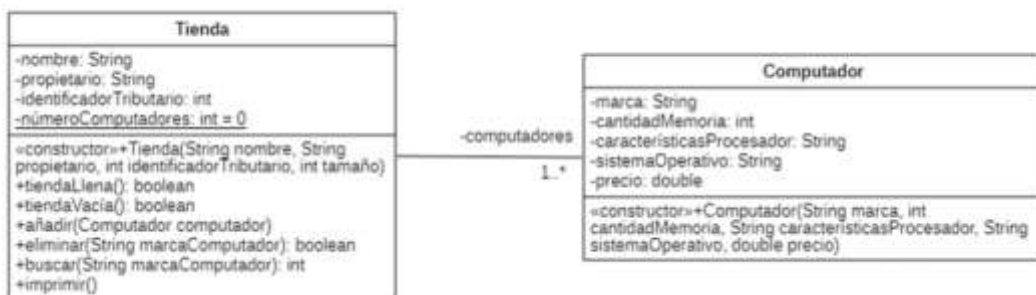
La clase tienda debe tener determinados métodos que permitan añadir la siguiente funcionalidad

- Agregar un computador a la tienda.
- Eliminar un computador de la tienda dada su marca.
- Buscar un computador en la tienda dada su marca.
- Listar la información de todos los computadores que tiene la tienda

```
public static void main() {  
    Tienda tienda = new Tienda("Tienda Cuántica", "Pepito Pérez", 123456, 5);  
    Computador computador1 = new Computador("Acer", 50, "Intel iCore 7", "Windows",  
18500000);  
    Computador computador2 = new Computador("Toshiba", 80, "Intel iCore 5", "Windows",  
15500000);  
}
```

```

Computador computador3 = new Computador("Mac", 100, "Intel iCore 7", "Mac", 2500000);
tienda.añadir(computador1);
tienda.añadir(computador2);
tienda.añadir(computador3);
Scanner sc = new Scanner(System.in);
String marca = sc.next();
System.out.println("El computador a buscar: " + marca + " se encuentra en la posición " +
    tienda.buscar(marca));
tienda.imprimir();
}
    
```



Ejercicio 5

Se tiene un curso universitario el cual contiene un array de estudiantes. Para cada estudiante se tienen los datos: nombre y apellidos del estudiante, código, número de semestre y nota final del estudiante. Se requiere implementar los siguientes métodos:

- Añadir un estudiante al curso: se ingresan por teclado los datos del estudiante. El código del estudiante debe ser único, si el código ya existe se debe generar el mensaje correspondiente.
- Buscar un estudiante de acuerdo con su código ingresado por teclado: si se encuentra muestra los datos del estudiante. De lo contrario, debe mostrar el mensaje correspondiente.
- Eliminar un estudiante de acuerdo con su código ingresado por teclado: si se encuentra muestra los datos del estudiante y se solicita una confirmación de la eliminación. Si no, debe mostrar el mensaje correspondiente.
- Calcular promedio del curso: sumar las notas de los estudiantes y dividirlos por la cantidad de estudiantes que tiene el curso.
- Obtener la cantidad de estudiantes que aprobó el curso: calcular el número de estudiantes que obtuvo un promedio mayor o igual a 3.0 y mostrarlo en pantalla. También se debe calcular el porcentaje de estudiantes que aprobó el curso.

Ejercicio 6

Se quiere diseñar aplicación para gestionar todos los datos de un campeonato de fútbol sala que se organiza este año en la ciudad. Aquellos que quieran participar deberán formar un equipo (nombre, patrocinador, color de la 1ª camiseta, color de la 2ª camiseta, categoría, . . .) e inscribirse en el campeonato. A medida que transcurran los partidos se irán almacenando los resultados de éstos, así como qué equipos lo jugaron, en qué campo se jugó, quién lo arbitró y alguna incidencia que pudiera haber ocurrido (en caso de que no ocurran incidencias no se anotará nada. Además, los participantes deberán rellenar una ficha de suscripción con algunos datos personales (nombre, apellidos, edad, dirección, teléfono, . . .)

Ejercicio 7

Se quiere diseñar una aplicación para controlar el acceso a las pistas deportivas de Gijón. Se tendrán en cuenta los siguientes supuestos:

- Todo aquel que quiera hacer uso de las instalaciones tendrá que registrarse y proporcionar su nombre, apellidos, email, teléfono, dni y fecha de nacimiento
- Hay varios polideportivos en la ciudad, identificados por nombre, dirección, extensión (en m2)
- En cada polideportivo hay varias pistas de diferentes deportes. De cada pista guardaremos un código que la identifica, el tipo de pista (tenis, fútbol, pádel, . . .), si está operativa o en mantenimiento, el precio y la última vez que se reservó
- Cada vez que un usuario registrado quiera utilizar una pista tendrá que realizar una reserva previa a través de la web que el ayuntamiento ha creado. De cada reserva queremos registrar la fecha en la que se reserva la pista, la fecha en la que se usará y el precio. Hay que tener en cuenta que todos los jugadores que vayan a hacer uso de la pista deberán estar registrados en el sistema y serán vincula-dos con la reserva

Ejercicio 8

Diseña el diagrama de clases de una aplicación para la gestión de la secretaría de un colegio:

- Cuando un alumno venga a matricularse se le hará rellenar una ficha con sus datos que luego se introducirá a la aplicación: DNI, número de expediente, nombre, apellidos, domicilio, teléfono y e-mail
- Se tendrá ya almacenada información sobre todos los cursos que se imparten en el centro: un código, el nombre, aula donde se imparte y el horario

- Se almacenarán todas las asignaturas de todos los cursos con el fin de conocer en que asignaturas se matricula cada alumno. De cada una guardaremos el nombre, el profesor que la imparte y el número de horas a la semana. Hay que tener en cuenta que todos los alumnos se matricularán al menos de una asignatura en algún curso
- Los profesores también están dentro de la Base de Datos, con la siguiente información: nombre, apellidos, domicilio y e-mail. Un profesor podrá impartir como máximo 6 asignaturas y deberá impartir al menos una. También habrá que almacenar el curso del que un profesor es tutor, teniendo en cuenta que puede que no sea tutor de ningún curso
- Hay que tener en cuenta que es importante almacenar las notas que cada alumno tiene en cada asignatura a lo largo del curso en las distintas evaluaciones, así como las observaciones que los profesores podrán anotar. Además, al principio de curso los alumnos escogerán con que compañero quieren realizar las prácticas o trabajos de cada una de las asignaturas (será siempre con el mismo), información que almacenaremos también en la Base de Datos

Ejercicio 8

Diseña el modelo de clases para la siguiente aplicación de gestión para uno de los hoteles del complejo Gran Scala:

- En este hotel, los clientes se registran por Internet al solicitar su reserva, que podrá incluir una o varias habitaciones. En cualquier caso, la fecha de entrada y salida de todas las habitaciones de una misma reserva será la misma. Además, debido a la exclusividad del hotel, cada habitación dispone de personal propio (uno o varios empleados, dependiendo del tipo de habitación). De los clientes almacenaremos nombre, apellidos, teléfono, email y fecha de nacimiento. Hay que tener en cuenta que guardaremos un histórico de todas las reservas que un

cliente vaya realizando a lo largo del tiempo, pero en un momento dado un cliente sólo puede tener una reserva.

- Es importante almacenar tanto la fecha de entrada y salida de cada una de las reservas, así como las habitaciones que la componen. De cada habitación almacenamos el tipo, precio, extensión y el número de clientes que hasta el momento la han utilizado. Cada habitación tiene asignado un número determinado de empleados, de los que guardamos nombre, apellidos, horario y cargo que desempeñan. Hay que tener en cuenta que cada empleado sólo puede tener asignada una habitación donde trabajar.
- A la hora de realizar el cobro de los servicios utilizados, será importante almacenar cuando un cliente hace un uso de los empleados de alguna de sus habitaciones, puesto que tiene un coste por tarea desempeñada. Habrá que anotar la fecha y hora en la que se hace uso de sus servicios y el precio del mismo. Así, todos esos cargos se añadirán al precio de la propia reserva.
- Por último, hay que tener en cuenta que será interesante tener en cuenta que las habitaciones tienen un precio pero éstas pueden ser reservadas a precios más bajos dependiendo de la oferta en vigor, que deberá guardarse también. Tanto el precio de la habitación como el precio que realmente paga el cliente por ella son datos que deben ser registrados.

Ejercicio 9

Una empresa desea crear un sitio Web de comercio electrónico al que se podrán conectar clientes para realizar sus compras. Se tiene que realizar el diseño de la Base de Datos que soporte la operativo de este sitio Web.

- Cuando un usuario intenta entrar en este sitio, se le pedirá un login y una contraseña. El sistema comprobará si el usuario tiene cuenta y en caso negativo se le pedirán los siguientes datos de alta: NIF, correo, nombre, dirección, teléfono, login y password.

Se comprobará si ya existía con distinto login para darle un mensaje de error.

- Una vez el usuario se ha dado de alta o ha entrado con su login y password correctos, puede visitar las distintas secciones de la tienda virtual. Nuestra empresa quiere que quede constancia de las secciones visitadas por los usuario y la fecha en la que la visitaron. Hay que tener en cuenta que un usuario podrá visitar varias secciones. De cada sección se almacenará un código, nombre, descripción y fecha de creación.
- Los usuarios pueden realizar sus compras utilizando un carrito virtual. Cuando un usuario decide utilizar el carrito, el sistema creará uno almacenando la fecha de creación. El usuario entonces puede poner productos, detallando cuantas unidades desea o bien eliminarlos. Un carrito puede contener varios productos y un producto puede aparecer en carritos de diferentes usuarios.
- De los productos se almacenará el código de producto, el nombre, la descripción y el precio por unidad. Cuando un usuario decide finalizar su compra, el sistema le pedirá entonces los datos bancarios (si es la primera vez que paga) y dará el carrito por finalizado. El usuario puede dejar un carrito lleno y no completar la compra en esa sesión, para completarla otro día. El usuario debe poder comprobar cuál es el coste total de un carrito antes de pagarlo. Además, podrá comprobar el precio total de todos sus carritos anteriores y su contenido.
- En este sitio Web los productos están organizados en las diferentes secciones teniendo en cuenta que un producto puede aparecer en varias secciones y una sección puede tener varios productos.