*TRABAJO PRÁCTICO N°4*

1. Realizar un programa en el que debamos crear una clase Alumno y como atributos aparte de nombre, apellido y dni tenga tres notas, luego en el main crear 15 alumnos y darle al usuario la opción de calcular:

a) el promedio de toda la clase

b) el promedio de un alumno

c) la nota más baja

d) la nota más alta

1. Hacer una clase llamada Persona que siga las siguientes condiciones:

Sus atributos son: nombre, edad, sexo (H hombre, M mujer, I indeterminado por defecto), peso y altura. No queremos que se accedan directamente a ellos. Determinar que modificador de acceso es el más adecuado, también su tipo. Si se requiere añadir algún atributo es válido.

Por defecto, todos los atributos serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.).

Se implantarán 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 (): calculara si la persona está en su peso ideal (peso en kg/ (altura^2 en mts)), devuelve un -1 si está por debajo de su peso ideal, un 0 si está en su peso ideal y un 1 si tiene sobrepeso. Se recomienda usar constantes para devolver estos valores.

Un IMC por debajo de 18,5 indica desnutrición o algún problema de salud, mientras que un IMC superior a 25 indica sobrepeso.

* esMayorDeEdad (): indica si es mayor de edad, devuelve un booleano.
* comprobar Sexo (char sexo): comprueba que el sexo introducido es correcto. Si no es correcto, será I. No será visible al exterior.
* toString (): devuelve toda la información del objeto.
* Métodos set de cada parámetro.Investigar el método to

Ahora, crea otra clase con método main () que haga lo siguiente:

Crea 3 objetos de la clase anterior, asignar diferentes valores directamente con variables en la codificación.

Para cada objeto:

* Comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
* Indicar para cada objeto si es mayor de edad.
* Por último, mostrar la información de cada objeto.

1. Cree una clase llamada CuentaDeAhorros. Use una variable static llamada tasalnteresAnual para almacenar la tasa de interés anual para todos los clientes. Cada objeto de la clase debe contener una variable de instancia llamada saldoAhorros, que indique la cantidad que el ahorrador tiene actualmente en depósito. Proporcione el método calcularlnteresMensual para calcular el interés mensual, multiplicando el saldoAhorros por la tasalnteresAnual dividida entre 12; este interés debe sumarse al saldoAhorros. Proporcione un método static llamado modificarTasalnteres para establecer la tasalnteresAnual en un nuevo valor. Escriba un programa para probar la clase CuentaDeAhorros. Cree dos instancias de objetos CuentaDeAhorros, ahorradorl y ahorrador2, con saldos de $20000.00 y $30000.00. Establezca la tasalnteresAnual en 4%, después calcule el interés mensual para cada uno de los 12 meses e imprima los nuevos saldos para ambos ahorradores. Luego establezca la tasalnteresAnual en 5%, calcule el interés del siguiente mes e imprima los nuevos saldos para ambos ahorradores.
2. Se desea diseñar un programa para registrar la información de las reservas de una empresa dedicada al alquiler de automóviles, teniendo en cuenta que:

* De cada cliente se desean almacenar su DNI, nombre y apellido, dirección y teléfono. Además, dos clientes se diferencian por un código único.
* Un determinado cliente puede tener en un momento dado hechas varias reservas.
* Una reserva la realiza un único cliente, pero puede involucrar varios autos.
* Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los autos, el precio total de la reserva y un indicador de si el auto o los autos han sido entregados.
* De cada auto se requiere la matricula, la marca, el modelo, el color y las plazas (cantidad de personas que puede transportar).

1. Se desea diseñar un programa que registre libros (técnicos y novelas) para una librería y permita buscarlos, venderlos y verificar su stock.

L as novelas se clasifican como de ciencia ficción, romance, misterio, juveniles y policiales. Los libros técnicos se clasifican como de ingeniería, ciencias naturales o ciencias sociales.

Cada libro tiene un título, uno o más autores, una editorial, un año de edición y formato (tapas duras o edición económica). Los libros tienen además un código ISBN y capítulos, los que tratan una o más materias (en los técnicos) o es una simple división (en las novelas).

La librería obtiene los libros por medio de proveedores que representan a una o más editoriales. De cada libro se tiene un stock (que puede ser cero). Al venderse un libro, el stock se actualiza. Si un cliente requiere un libro cuyo stock es cero, se puede realizar un encargo por parte del cliente. Esto significa que se pide el libro a un proveedor de la editorial del libro.

1. Escribe un programa para una biblioteca que contenga libros y revistas. Las características comunes que se almacenan tanto para las revistas como para los libros son el código, el título, y el año de publicación. Estas tres características se pasan por parámetro en el momento de crear los objetos.

Los libros tienen además un atributo prestado. Los libros, cuando se crean, no están prestados.

Las revistas tienen un número. En el momento de crear. En el momento de crear las revistas se pasa el número por parámetro.

Tanto las revistas como los libros deben tener (aparte de los constructores) un método toString () que devuelve el valor de todos los atributos en una cadena de caracteres. También tienen un método que devuelve el año de publicación, y otro el código.

Para prevenir posibles cambios en el programa se tiene que implementar una interfaz Prestable con los métodos prestar (), devolver () y prestado. La clase Libro implementa esta interfaz.

1. Utilizar la clase Persona.

Realizar una nueva clase llamada EquipoFutbol con un único atributo de nombre del equipo.

Realizar una nueva clase que herede de Persona llamada JugadorFutbol. (esto es porque, aunque no lo parezcan a veces los jugadores de futbol son personas).

* El JugadorFutbol tiene como atributo el nombre de su equipo de tipo EqupoFutbol.
* Incluir método toString que regrese todos sus valores como Persona y como Jugador de futbol juntos.

Realizar una nueva clase que herede de Persona llamada Aficionado. (Los aficionados también son personas).

* El Aficionado tiene como atributo equipoFavorito que es un tipo de EquipoFutbol.
* El Aficionado tiene como atributo un JugadorFavorito de tipo JugadorFutbol.
* Incluir método toString que regrese todos sus valores como Persona y como aficionado juntos.

Cambiar la clase Persona como abstracta. Ahora crear una nueva clase para probar nuestro modelo.

* Crear un solo arreglo llamado asistentesClasico de 5 localidades, el cual contendrá 3 Aficionados (2 al RIVERy 1 de RACING) y 2 JugadorFutbol (uno de RIVER y otro de RACING).
* Imprimir todos los valores de los objetos contenidos en el arreglo.