

## EJERCICIOS TEMA 6

1. Crea la clase Vehiculo, así como las clases Bicicleta y Coche como subclases de la primera. Para la clase Vehiculo, crea los atributos de clase vehiculosCreados y kilometrosTotales, así como el atributo de instancia kilometrosRecorridos. Crea también algún método específico para cada una de las subclases. Prueba las clases creadas mediante un programa con un menú como el que se muestra a continuación ; instanciar un objeto tipo Bicicleta y otro tipo Coche:

### VEHÍCULOS

=====

1. Anda con la bicicleta
2. Haz el caballito con la bicicleta
3. Anda con el coche
4. Quema rueda con el coche
5. Ver kilometraje de la bicicleta
6. Ver kilometraje del coche
7. Ver kilometraje total
8. Ver número total de vehículos creados.
9. Salir

Elige una opción (1-9):

Crear otros dos paquetes, para dar solución al mismo problema con sus correspondientes modificaciones:

1-a) packBIS: Probar el mismo menú de opciones usando un array de tipo Vehiculo, con dos elementos, el primero de tipo Bicicleta y el segundo de tipo Coche.

1-b) packBISBIS: añadir una clase ConjVehiculo, y solucionar el problema para las siguientes opciones del menú:

1. Crear Bicicleta
2. Anda con la bicicleta
3. Haz el caballito con la bicicleta
4. Crear Coche
5. Anda con el coche

6. Quema rueda con el coche
7. Ver kilometraje de las bicicletas
8. Ver kilometraje de los coches
9. Ver kilometraje total de todos los vehículos
10. Ver número total de vehículos creados.
11. Salir

2. Crea las clases Animal , Mamífero , Ave , Gato , Perro ,Canario ,Pingüino y Lagarto . Crea al menos, tres métodos específicos de cada clase y redefine el/los métodos cuando sea necesario. Prueba las clases creadas en un programa en el que se instancian objetos y se aplican dichos métodos.

3. Crea la clase Fracción . Los atributos serán numerador y denominador . Y sus métodos son: invierte (devuelve la fracción invertida), simplifica (Para simplificar una fracción, se comprueba si numerador y denominador son divisibles entre el mismo número. En tal caso, los dos se dividen. Se repite el proceso hasta que la fracción que se obtiene es irreducible (no se puede simplificar más),multiplica por un entero (devuelve la fracción multiplicada por un entero), divide entre un entero (devuelve la fracción dividida por un entero), divide dos fracciones y devuelve la fracción resultado, multiplica dos fracciones y devuelve la fracción resultado. Prueba todos los métodos de la clase Fracción.

4. Crea la clase Tiempo con los métodos suma y resta . Los objetos de la clase Tiempo son intervalos de tiempo y se crean de la forma `Tiempo t = new Tiempo(1, 20, 30)` donde los parámetros que se le pasan al constructor son las horas, los minutos y los segundos respectivamente. Crea el método `toString` para ver los intervalos de tiempo de la forma `10h 35m 5s` . Permite la suma de dos objetos de esta clase: por ejemplo `30m 40s` y `35m 20s` el resultado debería ser `1h 6m 0s` . Realiza un programa de prueba para comprobar que la clase funciona bien.

5. Crear la clase Profesor (identificador de profesor y dos constructores) que hereda de Persona (nombre, apellidos y edad), y que a su vez puede ser ProfesorInterino ( atributo fecha de comienzo de interinidad de tipo Calendar y dos constructores que invocan a la clase Profesor: uno con todos los atributos y otro con la fecha de comienzo de interinidad) o ProfesorTitular. Probarlo en una clase test.

6. Crear un proyecto para organizar a un conjunto de personas que se apuntan a una asociación. De cada persona interesa saber sus datos básicos: NIF, nombre completo y fecha de nacimiento. Cuando cada nuevo socio se da de alta, se le asigna un código de asociado único y se anota la fecha de alta (usar la clase de java Gregorian Calendar); de forma automática. Crear un menú con las siguientes opciones:

1. Alta de socio (pedir datos: nif, nombre y fecha de nacimiento (crear una clase Fecha propia: día, mes y año)).
2. Modificación de fecha de nacimiento (pedir el número de socio).
3. Listado completo.
4. Listado completo ordenado según el nombre.
5. Listado completo ordenado según la fecha de nacimiento.

## 6. Salir.

7. Se plantea desarrollar un programa Java que permita la gestión de una empresa agroalimentaria que trabaja con tres tipos de productos: productos frescos, productos refrigerados y productos congelados. Todos los productos llevan esta información común: fecha de caducidad y número de lote. A su vez, cada tipo de producto lleva alguna información específica. Los productos frescos deben llevar la fecha de envasado y el país de origen. Los productos refrigerados deben llevar el código del organismo de supervisión alimentaria, la fecha de envasado, la temperatura de mantenimiento recomendada y el país de origen. Los productos congelados deben llevar la fecha de envasado, el país de origen y la temperatura de mantenimiento recomendada.

Hay tres tipos de productos congelados: congelados por aire, congelados por agua y congelados por nitrógeno. Los productos congelados por aire deben llevar la información de la composición del aire con que fue congelado (% de nitrógeno, % de oxígeno, % de dióxido de carbono y % de vapor de agua). Los productos congelados por agua deben llevar la información de la salinidad del agua con que se realizó la congelación en gramos de sal por litro de agua. Los productos congelados por nitrógeno deben llevar la información del método de congelación empleado y del tiempo de exposición al nitrógeno expresada en segundos.

Crear el código de las clases Java implementando una relación de herencia siguiendo estas indicaciones:

- a) En primer lugar realizar un esquema con papel y bolígrafo donde se represente cómo se van a organizar las clases cuando escribamos el código. Estudiar los atributos de las clases y trasladar a la superclase todo atributo que pueda ser trasladado.
- b) Crear superclases intermedias (aunque no se correspondan con la descripción dada de la empresa) para agrupar atributos y métodos cuando sea posible. Esto corresponde a “realizar abstracciones” en el ámbito de la programación, que pueden o no corresponderse con el mundo real.
- c) Cada clase debe disponer de constructor y permitir establecer (set) y recuperar (get) el valor de sus atributos y tener un método que permita mostrar la información del objeto cuando sea procedente.

Crear una clase TestHerencia con el método main donde se creen: dos productos frescos, tres productos refrigerados y cinco productos congelados (2 de ellos congelados por agua, otros 2 por aire y 1 por nitrógeno). Mostrar la información de cada producto por pantalla.

## 8. 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. El código será un identificador de tipo entero, único para cada publicación. Se asignará automáticamente este valor a cada publicación, en el momento de su alta.
- 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

Implementar una superclase Publicacion con tres atributos. Un constructor con los tres atributos. Getters y setters. ToString() que devuelva estos tres atributos concatenados en forma de cadena. Las clases Libro y Revista; ídem. Una interfaz Prestable que declara los métodos indicados sin implementarlos. La clase Libro los implementa.

Se pide realizar un proyecto que permita realizar las siguientes opciones:

1- ALTA DE REVISTA

2- ALTA DE LIBRO

3- PRESTAR UN LIBRO

4- DEVOLVER UN LIBRO

5- LISTADO COMPLETO

6- LISTADO DE REVISTAS MOSTRANDO NÚMERO TOTAL DE REVISTAS

7- LISTADO ORDENADO DE LIBROS MOSTRANDO SI ESTÁ O NO PRESTADO

8- SALIR

9. La empresa informática "IPM Tech" necesita llevar un registro de todos sus empleados que se encuentran en la oficina central, para eso ha creado un diagrama de clases que debe incluir lo siguiente:

**1. Empleado:**

Atributos:

- nombre: tipo cadena (Debe ser nombre y apellido)
- identificador: tipo int
- edad : entero (Rango entre 18 y 45 años)
- casado: boolean
- salario: tipo numérico doble

Métodos:

- Constructor con y sin parámetros de entrada
- Método que permita mostrar la clasificación según la edad de acuerdo al siguiente algoritmo:  
Si edad es menor o igual a 21, Principiante  
Si edad es  $\geq 22$  y  $\leq 35$ , Intermedio  
Si edad es  $> 35$ , Senior.
- Imprimir los datos del empleado por pantalla (utilizar salto de línea: \n para separar los atributos).
- Un método que permita aumentar el salario en un porcentaje que sería pasado como parámetro al método.

**2. Programador**(Especialización de Empleado). Clase que hereda de Empleado todos los atributos y métodos.

Atributos:

- lineasDeCodigoPorHora : tipo entero
- lenguajeDominante: tipo cadena

Metodos:

- Constructor con y sin parámetros de entrada.

**3. Contable:** Clase que hereda de Empleado todos los atributos y métodos.

Atributos:

- asientosPorHora: Asientos contables por hora (int)

Metodos:

- Constructor con y sin parámetros de entrada.

Se pide realizar un proyecto que permita realizar las siguientes opciones:

1- ALTA DE PROGRAMADOR

2- ALTA DE CONTABLE

3- AUMENTAR SALARIO DE UN EMPLEADO

4- LISTADO COMPLETO DE EMPLEADOS

5- LISTADO COMPLETO DE EMPLEADOS CON CLASIFICACIÓN POR EDAD

6- LISTADO DE PROGRAMADORES AGRUPADOS POR LENGUAJE DOMINANTE

7- LISTADO DE CONTABLES ORDENADO EN ORDEN DESCENDENTE SEGÚN ASIENTOS POR HORA

8- SALIR

10. Para construir este proyecto:

A. Defina una interfaz que denominará **lpunto** con los siguientes métodos públicos:

- mover que devuelve void y recibirá dos valores en doble precisión
- mover que devuelve void y recibirá un lpunto
- borrar que devuelve void y no recibirá ningún parámetro.
- getX y getY que devuelven double.
- setX y setY que reciben un double y devuelven void.
- distancia que recibe un objeto lpunto y devuelve un double.
- distanciaorigen sin parámetros. Devuelve un double.
- imprime que devuelve void y no recibirá ningún parámetro.

B. Construya una clase Punto que implemente la interfaz lpunto y tenga dos atributos protegidos de tipo double denominados x e y (las coordenadas del punto), y un atributo estático protegido

llamado origen de tipo Punto con los valores 0 y 0. La clase Punto implementará los siguientes métodos:

- a. Un constructor sin parámetros que defina el punto (0,0).
  - b. Un constructor que reciba dos valores de tipo double para cada atributo.
  - c. Un método borrar que ponga a 0 los dos atributos.
  - d. Un método mover que dado los dos valores pasados como parámetros cambie los atributos x e y.
  - e. Un método denominado mover que cambie las coordenadas del Punto que invoca por las de un Punto pasado como argumento.
  - f. Un método distancia que calcule la distancia entre el Punto que invoca y otro que es pasado como argumento. Utilice el método sqrt de la clase Math.
  - g. Un método estático cambiarorigen que cambie el atributo origen por el punto que se le pasa como argumento.
  - h. Un método distanciaorigen que calcula la distancia al origen utilizando el método distancia.
  - i. Un método imprime que saca por pantalla los valores de los atributos con el siguiente formato ( x , y )
- C. Defina una interfaz que denominará Ipixel, que herede los métodos de la interfaz lpunto y que cuente además con los siguientes métodos públicos
- a. borrar que devuelve void y no recibirá ningún parámetro.
  - b. cambiarcolor que recibe un entero y devuelve void.
  - c. imprime que devuelve void y no recibirá ningún parámetro
- D. Cree una clase Pixel que extienda la clase Punto anterior y que implemente el interfaz IPixel, añadiendo un atributo privado de tipo entero que denominaremos color. Los métodos propios de la clase Pixel serán:
- a. Un constructor sin parámetros que pondrá a 0 todos los atributos.
  - b. Un constructor con parámetros que reciba un Punto y un entero para el color.
  - c. Un método imprime que redefina el método de la clase Punto escribiendo además el color.
  - d. Un método borrar que redefina el de la clase Punto poniendo el color a 0.
  - e. Un método cambiarcolor que cambie el color al valor que se le pasa como argumento.
- ¿Qué cambios deberá hacer en los atributos Punto?
- Haga un programa principal para probar todos los métodos contruidos, tanto para la clase Punto, como para la clase Pixel. Compruebe cómo se pueden invocar los métodos de la clase Punto desde objetos de la clase Pixel.