Herencia

1.- Mundo animal.

Vamos a realizar una estructura de clases en la que queden reflejados los dos tipos de animales domésticos principales: **perros** y **gatos**.
Todos tendrán un ***nombre***, una ***raza***, un ***peso*** y un ***color***. En cuanto a métodos tendrán los siguientes: ***vacunar***, ***comer***, ***dormir***, ***hacerRuido***, ***hacerCaso***. Los métodos *vacunar*, *comer* y *dormir* son comunes. Los métodos *hacerRuido* y *hacerCaso* serán sobreescritos en las especializaciones: *hacerRuido* para los perros será un ladrido y para los gatos un maullido; *hacerCaso* para los perros será un método *boolean* que devolverá true el 90% de las veces y para los gatos el 5%. Los perros tendrán un método particular: ***sacarPaseo***. Los gatos tendrán otro método que será: ***toserBolaPelo***.

2.- Vehículos.

Usando como base la clase realizada en el método anterior; renómbrala como vehículo. Ahora tendremos tres tipos de vehículos: ***turismos***, ***camiones*** y ***motocicletas***.
Para los **turismos** nos interesa saber: ***número de plazas*** y el método propio ***tipodeUso*** (profesional o particular).
Para los camiones: atributo ***peso máximo autorizado*** y el método ***esMercanciaPeligrosa*** (tipo boolean).
Para las motos: atributo ***cilindrada*** y el método ***necesitaCarnet*** (tipo boolean; a partir de 125 CC necesita carnet).

3.- Ítems de biblioteca.

Debemos definir la clase Ficha diseñada para manipular objetos de una biblioteca, como libros, revistas, DVD, etc. Todos estos objetos pueden tener en común un ***número*** que los identifique y un ***título***. La clase por tanto debe contener esa información y los métodos para manejarla. La idea es disponer de una clase para definir otras que amplíen su funcionalidad.
Pensemos ahora en un objeto particular.

- Un **libro** tendrá las características definidas por la clase ficha más algunas otras; por ejemplo: ***autor*** y ***editorial***.
- También las **revistas** tendrán información particular como: ***número de la revista*** y ***año de publicación***.
- Y finalmente para los **DVD** necesitaremos saber: ***director***, ***año*** y ***tipo*** (“película”, “documental”, etc...).

Define la clase base y clases derivadas necesarias; así como los métodos que creas convenientes.

4.- Currantes.

Tenemos tres tipos de trabajadores en nuestra empresa: **Fijos**; **PorHoras** y **AComisión**. Los atributos principales para ellos son:

- **Fijos**: *nombre*, *apellidos* y *sueldo*.
- **PorHoras**: *nombre*, *apellidos*, *horas*, *sueldohora*.
- **AComision**: *nombre*, *apellidos*, *ventas* y *porcentaje*.

Crea las clases que creas necesarias. Observa que existen una serie de atributos comunes. Podría interesar hacer una clase general Trabajador que las maneje y que sirva de base para los tres tipos de trabajadores.

5.- Cuentas bancarias.

Heredando de la clase "**cuenta**" que creamos en la relación de ejercicios anterior. Vamos a crear dos nuevas clases derivadas: **cuentaahorro** y **cuentacorriente**.
Además de los atributos de la clase cuenta, cada una de estas clases debe incluir:

- **Cuentaahorro**: Tienen una *cuota de mantenimiento* y un *interés anual*. Debemos implementar métodos que me permitan saber el saldo nuevo si aplicamos ese interés y que resten al saldo, la cuota de mantenimiento cuando sea necesario.
- **Cuentacorriente**: Nos permite hacer *transacciones* y *movimientos* de cuenta (por ejemplo domiciliaciones). Tenemos que manejar la siguiente información: *porcentaje cobrado* por transacción, el *número de transacciones* y las **transacciones** en sí (*día*, *mes* y *año* de la transacción, *concepto* e *importe*). Para esta última clase quizá te haga falta definir una nueva clase transacción parecida a la que hicimos cuando implementamos un vector dinámico.

Clases abstractas e interfaces

6.- Animales.

Sobre el ejercicio de animales domésticos realiza las siguientes variaciones: los métodos ***comer***, ***dormir*** y ***hacerRuido***. deben ser obligatorios para cualquier nuevo animal doméstico que se añada nuevo. Crea la clase **Elefante** que herede de la anterior y que incluya los métodos obligatorios (el elefante barrita). Realiza el ejercicio en los supuestos de que la clase Animal no se instancie nunca y en el supuesto que sí.

7.- Currando otra vez.

Para el ejercicio cuatro, diseña el método *calculaSalario* que será obligatorio para cualquier currante que se añada. Incluye un nuevo tipo de currante: becario, que lo tenga. Para el caso del becario el método devolverá 100, siempre.
Igual que anteriormente diseña el ejercicio en el supuesto de que haga falta instanciar la clase base y en el supuesto que no.
En cualquier caso los currantes *porHoras* y *becarios* tendrán un valor común: TIEMPO.

Polimorfismo

8.- Animales.

Pues otra vez con los animales domésticos. Vamos a diseñar una simulación de un parque. En este parque, dividido en sectores solo cabe un animal doméstico por sector.
Cada 10 segundos aparece un animal nuevo que se sitúa en una posición libre del parque; si no hubiera el animal se va.
Cada 2 segundos los animales del parque hacen algunas de las acciones habituales: comer, dormir o hacerRuido; al azar.
Cada 15 segundos un animal cambia de posición a otra adyacente. Si no hay hueco libre no se mueve.
Cada 20 segundos alguno de los animales abandona el parque con una probabilidad del 50%.

9.- Biblioteca.

Define la estructura necesaria que permita manejar los datos de una biblioteca. En dicha biblioteca tendremos libros, DVD's y revistas (tal y como definimos en el ejercicio 3). Diseña el siguiente menú:

1.- Añadir ítem (preguntará que tipo de ítem es y lo añadirá con sus datos a la biblio).
2.- Buscar ítem.
3.- Eliminar ítem.
4.- Listado de la biblioteca
5.- Salir