

EJERCICIOS C# VIERNES CON GITHUB

Se pide crear los siguientes ejercicios en VISUAL STUDIO Y SUBIRLOS A TU CUENTA DE GITHUB

1º) Un banco requiere controlar las cuentas de sus clientes y para ello las clasifica en Cuentas Bancarias:

Todas las cuentas del banco tienen los siguientes datos:

- Número de cuenta (entero).
- Nombre del cliente (cadena).
- Saldo (numérico real).

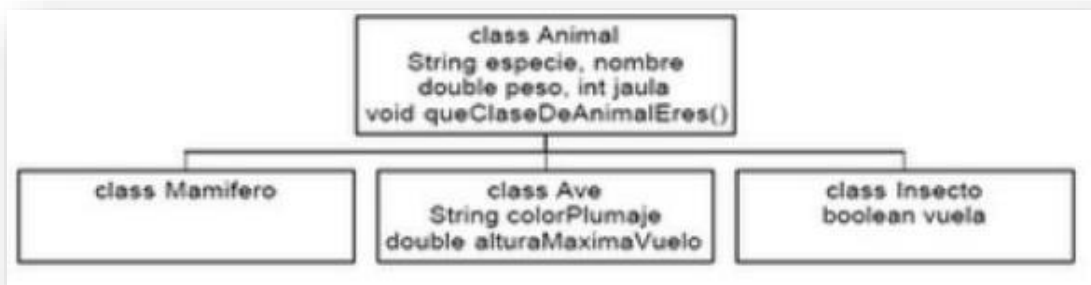
Además se realizan las siguientes operaciones con ellas:

- Consultar datos: A través de sus propiedades.
- Depositar: Incrementa el saldo con la cantidad de dinero que se deposita.
- Retirar: Antes de hacer el retiro, se debe verificar la suficiencia de saldo y en caso de aprobarlo, se disminuye el saldo.

Se pide además un menú similar al mostrado en donde el usuario pueda entre otras operaciones:

- 1: Crear cuenta
 - 2: Depositar una cantidad
 - 3: Retirar dinero
 - 4: Consultar Saldo
 - 5: Salir de la App

2º) Gestión de un Zoológico. A continuación, se presenta la jerarquía de clases que representa los animales de un posible zoológico:



La clase Animal es una clase abstracta con cuatro atributos miembros protected:

- a) Una cadena indicando la especie (león, águila, abeja).
- b) Una cadena indicando el nombre del animal concreto.
- c) Un dato numérico real indicando el peso en kg.
- d) Un dato numérico entero indicando el número de jaula que se asigna al animal.

Además, la clase Animal declara un método virtual queClaseDeAnimalEres() que habrá que definir en las clases derivadas.

La clase Mamifero no añade nuevos atributos miembro, aunque deberá implementar el método queClaseDeAnimalEres().

La clase Ave tiene dos nuevos atributos protected:

- a) Una cadena colorPlumaje indicando el color predominante...
- b) Un dato numérico real indicando la alturaMaximaVuelo.

La clase Insecto tiene un nuevo atributo miembro protected de tipo booleano llamado vuela que indica si el insecto vuela o no.

SIGUE

Para realizar este ejercicio se pide lo siguiente:

- Crear las cuatro clases indicadas, con los correspondientes constructores y sobrecarga de constructores, Como ayuda, se indica que el orden de los argumentos en el constructor parametrizado de la clase base es:

public Animal(String especie, String nombre, double peso, int jaula) {...}

- Definir los métodos llamados queClaseDeAnimalEres() en cada una de las clases derivadas de Animal. Este método no tiene valor de retorno (es void) ni argumentos. Debe ser capaz de mostrar por la pantalla la información correspondiente al animal de que se trate (ver el ejemplo), utilizando para ello la información almacenada en las variables miembro.

Ejemplo:

```
Soy un mamífero llamado: xxxxxxxx
de la especie: xxxxxxxx
Peso en Kg: xxx
Estoy en la jaula: xx
```