

Solución:

a) 3, En el Código se están definiendo 3 clases: La clase "Animal", la clase "Familia" y la clase "FamiliaAnimales".

b) El programa principal se define en la clase "FamiliaAnimales", al correr el programa este genera una secuencia de eventos, donde primero nos hablan sobre el matrimonio de Cebra con Caballo; posteriormente, se nos dice que estos van a tener un hijo, el cual tiene de nombre "cebrallo".

El programa nos dice que la madre de cebrallo es Cebra, y el padre es Caballo, y que cebrallo es de género femenino; luego, cebrallo menciona su nombre y nos dice su peso, el cual es de "1.0".

El programa nos imprime el total de animales en el momento, el cual, es de 3, para acto seguido, anunciar la muerte de cebrallo.

La última línea impresa por el programa indica que quedan 2 animales.

c) 3 objetos, animal1, Cebrallo mediante el método procreación, y animal2.

d) Se crean los objetos cebra, caballo y cebrallo.

e) La clase animal tiene en total 5 atributos: nombre de tipo String, género de tipo String, peso de tipo double, pareja de tipo animal, y totalAnimales de tipo static int.

f) Los atributos de tipo primitivo son peso (double), y totalAnimales(static int).

g) Se hace referencia a objetos en los atributos nombre(String), genero(String) y pareja(Animal).

h) El atributo pareja se inicializa con el valor null.

i) Los nombres en animal1 y animal2, por ser Strings, se inicializan con el valor null.

j) El peso de animal 1 no se establece de forma específica, por ende, su valor es de 0.0.

k) revisar el archivo adjunto "dibujo punto k".

l) El valor de genero en animal2 es de m, el cual es el valor asignado por defecto a cada objeto de la clase animal.

m) - En caso de comentar la línea 15, Cebra no tendrá una pareja asignada.

- En caso de comentar la línea 16, El caballo no tendrá una pareja asignada.

- Al comentar las dos líneas, ambos objetos(cebra y caballo) van a carecer de pareja, generando el error "cebra no tiene pareja" al usar el método procrear.

n) el método procrear se ejecuta sobre el objeto animal1(cebra).

o) en caso de colocar el modificador final en pareja, una pareja asignada no podría ser modificada, y a la vez, el método setPareja no funcionaría bien.

p) No se puede eliminar el modificador static del método morir sin afectar el programa porque el método morir se llama como Animal.morir(animal2), y además, sin static, tendría que llamarse desde una instancia de Animal.

q) En este caso, todos los animales serían machos, además, está el problema con el método de procrear, pues este requiere géneros diferentes.

r) Debido al valor por defecto que tiene esta variable, el cual es 0, no es necesario tener un valor inicial.

s) Al animal1 y animal apuntar al mismo objeto (cebra), no cambia la salida del programa.

t) Al igual que en s, se hace referencia al mismo objeto independientemente de si usamos animal o animal1.

u) La clase Animal tiene 5 métodos: setPareja(), tenerHijo(), procrear(), morir() y toString()

v) procrear() retorna un objeto de tipo familia, mientras que tenerHijo() retorna un objeto de tipo animal.

w) Se pasa por referencia, al ser un objeto de tipo animal.

x) Se hace referencia al objeto animal1(cebra), y por la ambigüedad entre el parámetro pareja y el atributo pareja en clase, no se puede omitir el uso del this.

y) Debido a la referencia del objeto actual (la instancia actual de familia), el uso del this es redundante, tener o no tener a este no hace diferencia.

z) No se puede quitar el static en morir(), pues se llama como Animal.morir(animal2), además, Sin static, se debería llamar desde una instancia de animal.

aa) El this en las líneas de 44 y 46 en el programa principal hace referencia al objeto que está procreando(animal1), es necesario para distinguir entre el objeto actual y las variables locales.

bb) No hay un punto "bb" en el taller.

cc) Al ser este un atributo de clase, mantiene un contador global de los animales que existen, pues en caso de este ser de instancia, cada animal tendría su propio contador, además, este atributo cumple la función adicional de tener un registro centralizado y accesible a todas las instancias.

dd) No es posible agregar el modificador static al método tenerHijo() sin afectar su funcionamiento, pues

necesita acceder a atributos de instancia, atributos que no son static, y operar en un animal específico.
ee) toString() se usa con el proposito de mostrar información del objeto al imprimir.