

Solucion Taller 3

Cristhian David Amaya 25481092

1. Responda las siguientes preguntas: (Sacadas de H.M. Deitel and P.J. Deitel, Java How to Program)

a. ¿Cuál cree que es rol de herencia en un programa de Java?

El rol de la herencia en un programa es determinar las diferentes jerarquías que puedan desprenderse de él, evitar la duplicación de código puesto que la clase padre hereda a las clases hijas sus diferentes atributos y métodos dispuestos para ser heredados, estando totalmente disponibles para su uso por parte de las clases hijas y sus instancias quienes se podría decir se especializan en el área de la clase padre, además mediante el uso de esta característica se pueden agrupar diferentes entidades que tengan un rol en común, pudiendo ser llamados mediante el mismo, es decir que si se declara una clase de la cual se pretenden heredar diferentes atributos o métodos se puede crear una variable estática dentro de la misma en donde se guarde una referencia de la instancia de las diferentes clases hijas y de sí misma.

b. ¿Cómo la herencia promueve la reutilización de software?

Mediante la herencia una clase padre hereda a las diferentes clases hijas algunos de sus atributos o métodos, esto quiere decir que los atributos o métodos declarados en la clase padre y dispuestos para ser heredados estarán disponibles para el uso por parte de las clases hijas, de esta forma se evita tener que escribir más de una vez el mismo código que estará disponible para diferentes clases mediante la herencia.

c. ¿Cómo se podría explicar la Jerarquía (Hierarchy) en la programación orientada a objetos?

Se explica mediante el hecho de que una clase agrupe las características comunes a un área específica y que mediante la herencia diferentes clases hereden de la primera estos atributos y métodos pero mediante el poliformismo lo moldeen a su entorno de acuerdo a las necesidades específicas, con ello se puede entender que la primer clase se tiene una mayor jerarquía que a las que hereda algunos de sus atributos y métodos quienes tienen una jerarquía directamente inferior a esta, y de darse el caso de que otras clases derivasen sus atributos o métodos de estas últimas, tendrán igualmente una jerarquía inferior a ella aclarando un poco será algo como Clase Padre - Clase Hija- clase Nieta.

d. Explique la diferencia entre Composición (composition) y herencia. De un ejemplo.

La composición generalmente se usa cuando se quieren las características de una clase existente dentro otra clase, pero no en su interfaz (en el sentido de métodos o atributos accesibles mediante la misma clase). Esto quiere decir que aloja un objeto para implementar características en su clase, pero el usuario de su clase ve la interfaz que se ha definido, en vez de la interfaz de la clase original. Para hacer esto, se sigue el típico patrón de alojar objetos privados de clases existentes en su nueva clase.

```
Ex:   class Motor   {void start() const {};  stop() const {}};  
      class Rueda   {void inflar(int psi) const {}};  
      class Carro   {Motor motor;          Rueda rueda[4];};
```

La herencia hace referencia a una clase padre que hereda a las diferentes clases hijas algunos de sus atributos o métodos, esto quiere decir que los atributos o métodos declarados en la clase padre y dispuestos para ser heredados estarán disponibles para el uso por parte de las clases hijas desde su interfaz, es decir que podrán ser llamados desde la llamada a la clase hija.

```
Ex:    class Mascota{char comer() const {} float dormir(int) };  
        class Perro extends Mascota{void correr(){}; void ladrar(){};};  
        int main() { Perro bob; bob.comer(); bob.dormir(); bob.sleep(1);}
```

e. En java una subclase puede heredar de máximo una superclase. En otros lenguajes como c++ es posible que una clase herede de más de una clase (Herencia múltiple). Explique los pros y contras de esta práctica.

Realmente no existen contras en que una clase pueda heredar de muchas otras sus atributos y métodos, ya que el motivo por el que esto no se hace en java es que de dado el caso en que dos atributos o métodos tengan el mismo nombre, y los mismos parámetros pero pertenezcan a diferentes clases no se sabría de qué clase padre usar el método, pero esto sería fácilmente solucionable mediante la alusión a la clase de la cual se pretende usar el método o atributo como por ejemplo `this.ClasePadre::metodo()`, sin la necesidad de este método sea estático puesto que está en la interfaz de la clase hija.

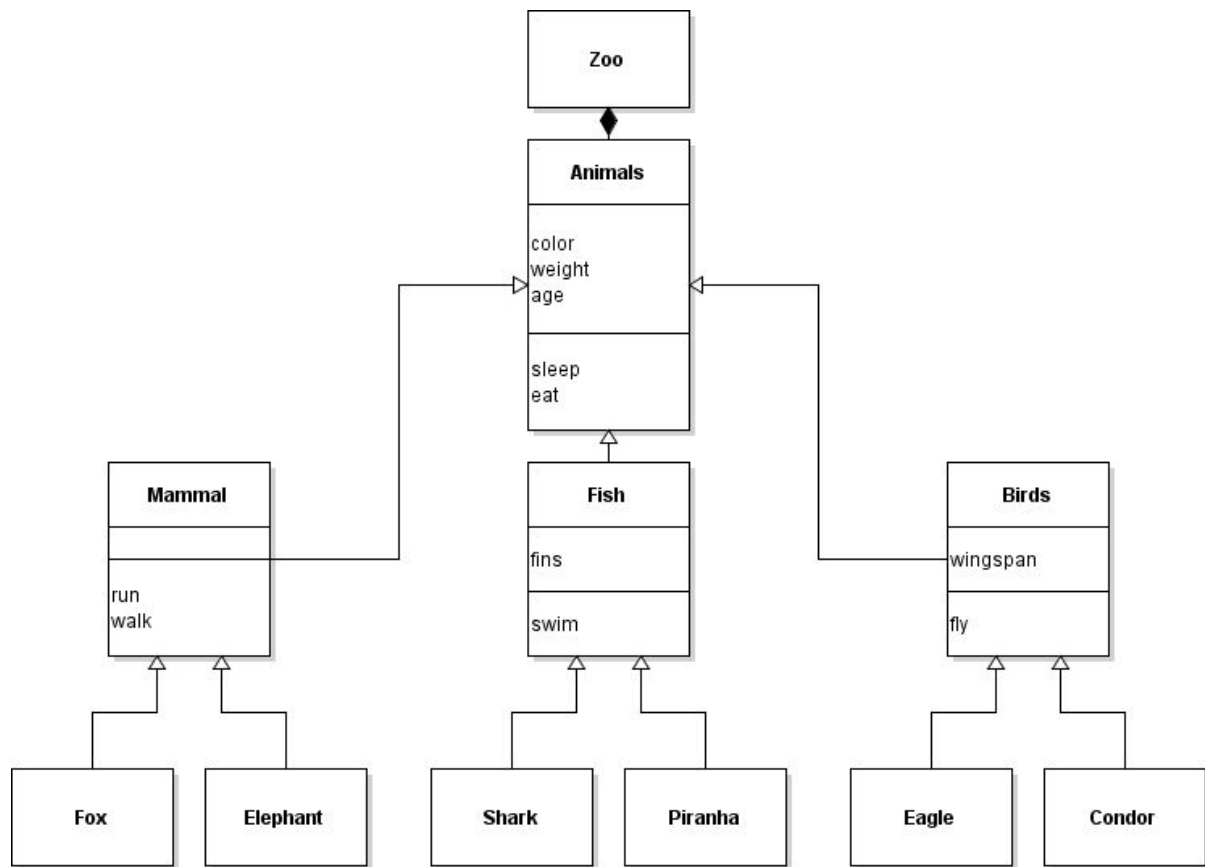
Como ventajas de esta práctica se tiene la ventaja de heredar de diferentes clases sus métodos o atributos, por ejemplo en el caso en que de de la necesidad de crear un área multidisciplinar como por ejemplo Mecatrónica que podría heredar de Mecánica, Sistemas y electrónica sus diferentes clases y atributos.

2. Herencia

Las Clases tanto sus atributos, métodos y otras variables usadas deben estar en inglés

a. Zoo – Diseño Top-Down

Usted ha sido contactado por un zoológico para diseñar un simulador que sirva para estudiar el reino animal. Usted debe crear un diagrama de clases que represente los animales del zoológico. Cada animal (EJ, León, Perro, Zorro, Lobo, Hipopótamo, Pez, Gato, Delfín, Águila) va a ser representado como un objeto. Use todos los atributos y métodos que considere relevantes para el modelado. El enfoque Top Down rompe un concepto general en unos conceptos más específicos. (Especialización: ir de algo general a algo específico)



Los puntos faltantes son prácticos y se encuentran adjuntos.