HELEN DANIELA BENITEZ HIPOLITO

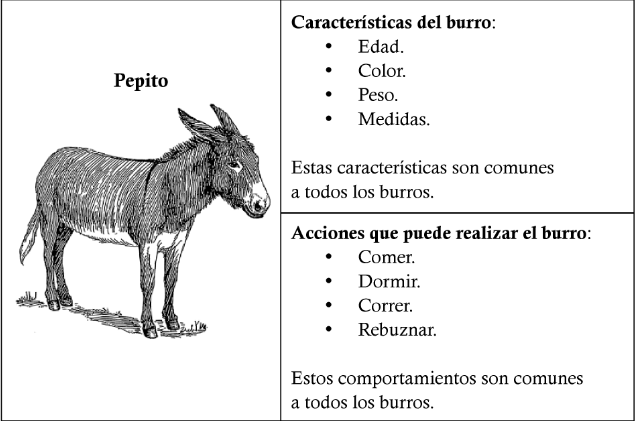
ADSI 2142329

PROGRAMACIÓN ORIENTADA A OBJETOS

DEFINICION DE CLASE Y OBJETO

Programar bajo el paradigma de programación orientada a objetos (POO) consiste en simular o modelar los objetos del mundo real. En un establo, por ejemplo, se pueden identificar diferentes animales como entes con características y acciones propias.

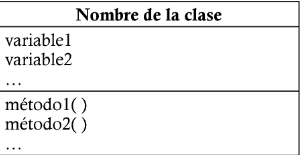
EJEMPLO:



El burro en mención (Pepito) es un objeto con características específicas. Claro está, en el establo pueden existir otros burros con diferente edad, color, peso y medidas. De esta manera se puede definir la CLASE Burro como una abstracción a todos los burros existentes en el establo; las características son las VARIABLES MIEMBRO O ATRIBUTOS DE LA CLASE y las acciones son los MÉTODOS MIEMBRO DE LA CLASE o COMPORTAMIENTO DE LA CLASE.

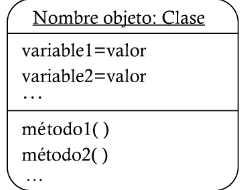
Una clase es una plantilla para crear objetos que constituye una abstracción del mundo real, como la clase Burro, la clase persona, la clase cuenta, etcétera. Las clases se nombran como sustantivos en singular y poseen variables que definen la información que se desea almacenar y métodos que definen las acciones que se desean realizar. Las variables se nombran como sustantivos y los métodos como verbos en infinitivo.

Las clases se pueden representar gráficamente mediante un diagrama cuyas especificaciones son establecidas por el lenguaje unificado de modelado (UML – Unified Modeling Laguage Version 2.4.4), el cual permite diseñar visualmente los sistemas de software. Uno de los diagramas de UML es denominado dia- grama de clases, donde se utiliza la siguiente representación:



Por otro lado, un objeto es una instanciación de una clase, es decir, la mate- rialización de la clase. Cuando se instancia un objeto se asignan datos a las variables de la clase y se pueden ejecutar los métodos. Fácilmente se pueden diferenciar las clases de los objetos con el siguiente paralelo: las clases son moldes de galletas y los objetos son las galletas creadas con los moldes. Todas las galletas creadas con el mismo molde son iguales, así que todos los objetos instanciados de la misma clase tienen las mismas variables y los mismos mé- todos disponibles. Se diferencian en los valores que se asignan a las variables.

Gráficamente, los objetos se representan mediante un diagrama de objetos es- tablecido por UML así:



ASPECTOS PARA TENER EN CUENTA

* Una clase puede no tener variables o no tener métodos.

• Por seguridad, se recomienda declarar las variables como privadas o protegidas.

• Para acceder las variables privadas, se recomienda implementar métodos accesores.

• Se recomienda leer valores por fuera de los métodos de la clase, de manera que puedan ser verificados en el método principal antes de ser enviados a la clase.

• Se recomienda no mostrar valores directamente dentro de los métodos de la clase, sino retornarlos para que sean mostrados por pantalla en el método principal. De esta manera se logra independencia de la interfaz gráfica.

• La sobrecarga de métodos se presenta en la misma clase.

• El polimorfismo se presenta sobre métodos en diferentes clases.

• Las clases heredadas pueden acceder solo los atributos y métodos visibles de la clase base, es decir, los públicos o protegidos.

• Si una clase tiene un método abstracto, la clase debe ser declarada como abstracta también.