

Programación orientada a objetos

Nombre: William Villalps

NRC: 1323

1. ¿Qué es paradigma de la programación orientada a objetos

Consiste en simular o modelar los objetos del mundo real. Es diferente al método clásico de programación estructurada. El método de POO usa objetos cuyo comportamiento resuelve problemas, genera programas y aplicaciones informáticas.

Con la POO se aumenta la modularidad de los programas y su reutilización además usa nuevas técnicas como son polimorfismo, encapsulamiento, herencia, etc. Podemos mencionar como ejemplo a un burro, es un objeto con características específicas llamadas variables miembro o atributos de la clase y sus acciones son los métodos miembro de la clase

Burro	Características del burro
	Edad Color Peso Medidas
	Acciones que puede realizar el burro:
	Correr Dormir Comer Rebuznar

2. ¿Qué es una clase, objeto, atributo o método.

CLASE

Es una plantilla para crear objetos, engloban dos componentes: atributos y métodos, se puede decir que es un tipo de dato. Ejemplos:

Clase burro, clase persona, clase cuenta, etc.

Se los nombra como sustantivos.

Las clases se representan gráficamente mediante un programa diagrama de clases UML:

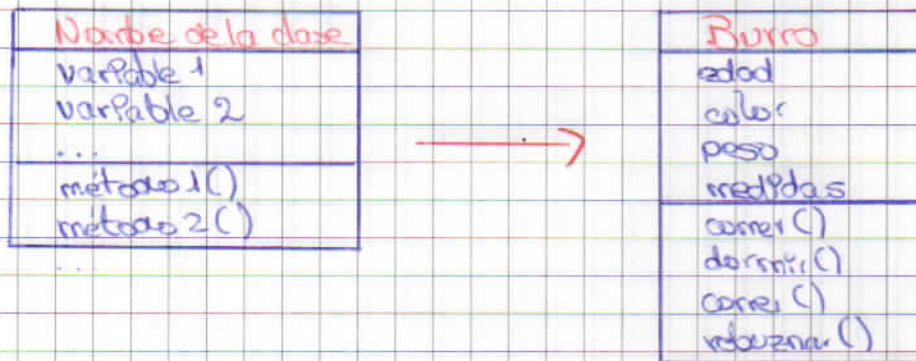


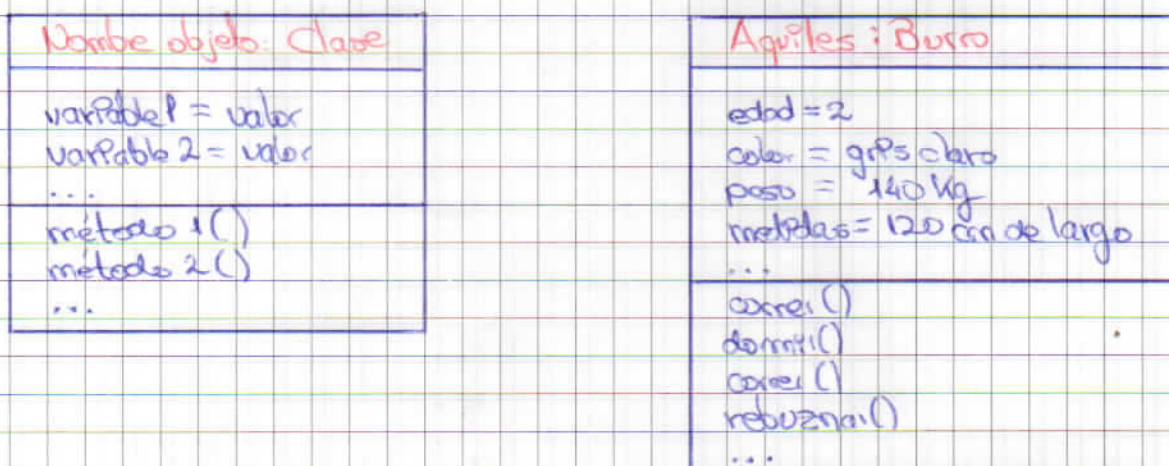
Diagrama de la clase Burro

OBJETO

Es la instanciación de una clase, es decir, la materialización de la clase.

Cuando se instancia un objeto se asignan datos a las variables de la clase y se pueden ejecutar los métodos, se puede decir que es parecido a una variable. Ejemplo: las clases son los modelos de galletas y los objetos son las galletas creadas con los moldes.

Se representan gráficamente mediante un diagrama de objetos UML:



El burro Aquiles es un objeto instanciado de la clase Burro.

ATRIBUTO

Son las propiedades que poseen los objetos de una clase.

MÉTODO

Son las acciones que los objetos de la clase pueden realizar.

3. ¿Qué es un sistema de control de versionamiento y para qué sirve.

Son un tipo de software que ayuda a hacer un seguimiento de los cambios realizados en el código a lo largo del tiempo. A medida que el desarrollador edita el código, el sistema de control de versiones toma una instantánea de los archivos, los guarda de forma permanente y se pueda recuperar más adelante si es necesario.

Sin el control de versiones, los desarrolladores se sienten tentados a mantener varias copias del código en su equipo. Esto es peligroso ya que es fácil cambiar o eliminar un archivo en la copia incorrecta del código y así perder todo el avance realizado.

Ventajas:

- Reconstrucciones de archivos, versiones anteriores cuando se requieren.
- Bitácora de modificaciones y cambios.
- Registro de todo tipo de gestión en el código fuente.

Desventajas:

- Curva de aprendizaje un poco elevada.
- El uso de comandos puede representar un problema si no se documenta su significado.
- Disponibilidad, el código está en un solo repositorio.

4. Realizar tres UML por persona de dos objetos hijos y un objeto padre

Clase Padre: Dispositivo

Marca
Modelo
Tipo
Precio
...
encender()
apagar()
sonido()
...

Clase Hijo 1: Smartphone

Número
Compañía Operadora
...
hacerLlamada()
enviarMensaje()
...

Clase Hijo 2: Tablet

Tamaño Pantalla
batería
...
tomarFoto()
dibujar()
...

Clase Padre: Organismo

Nombre científico
Edad
Habitat
Tipo Alimentación
...
nacer()
crecer()
alimentarse()
respirar()
...

Clase Hijo 1: Planta

Fotosíntesis
Tipo de Iluminación
tipo de suelo
...
podar()
florecer()
reproducción()
...

Clase Hijo 2: Animal

Edad
Habitat
Sexo
...
 locomoción()
dormir()
construir()
...

Clase padre: Producto

Nombre:
Codigo
Descripción
Precio

...
calculaVentaNeto()
stock
descuento()
...

Clase hijo: Producto eléctrico

Marca.
Voltaje
Tipo Uso
...
garantia()

Clase hijo 2: Producto Alimento

Nombre.
Fecha caducidad
tipo alimento
...
caducidad()
origen()
oferta()
...

Programación orientada a objetos

Nombre: WIPam Villaspé

NRC: 1323

Tipo de datos primitivos

Son tipos de datos básicos que están directamente soportados por un lenguaje de programación y almacenan valores simples. No son objetos y no tienen métodos asociados, su tamaño y comportamiento son fijos. Ejemplos:

- Entero (int): representa números enteros, como 5 o -12.
- Decimal: representa números de punto flotante, como 3,14 o -0,001
- Caracter: representa un solo carácter, A o Z
- Booleano: representa un valor lógico, true o false

Estos tipos son "inmutables" y sus valores se almacenan directamente en la memoria.

Tipos de datos referenciados

Se refieren a una dirección en la memoria donde está almacenado un objeto. No almacenan el valor directamente, sino que guardan una "referencia" a un objeto o estructura.

Ejemplos:

- Clases: en POO, los instancias de las clases son tipos de datos referenciados, ejemplo: objeto clase Persona.
- Arreglos (arrays): aunque algunos lenguajes tratan los arreglos como tipos primitivos, en otros, son referenciados.
- Interfaces: representan una referencia a un conjunto de métodos que las clases deben implementar.
- Strings: en Java, las cadenas de texto (string) son tipos referenciados y se manejan como objetos.

Tipo de datos estáticos:

Son aquellos cuyo tipo y tamaño se definen en tiempo de compilación y no pueden cambiar durante la ejecución del programa. Como tipo de dato son los numéricos (INT), carácter (Char), lógico (booleano)

Tipo de datos dinámicos

La memoria se asigna en tiempo de ejecución, pueden cambiar de tamaño o ser liberados durante la ejecución del programa. Dentro de este tipo de datos tenemos a los Lineales y NO Lineales. Dentro de los Lineales tenemos: listas, pilas, colas y dentro de los NO Lineales a los grafos.

Ejemplos de datos primitivos en Java:

```
package curso.java;  
public class TipoDatoPrimitivo {  
    byte n = 137;  
    float f = 5.6f;  
    double d = 58.75;  
    char c = '4';  
    boolean opcion = true;
```

Ejemplo dato de referencia:

```
public class Principal {  
    public static void main(String[] args) {  
        int valor = 10;  
        funcion(valor);  
        System.out.println(valor);  
    }  
    public static void funcion(int valorNumerico) {  
        valorNumerico = 20;  
        System.out.println(valorNumerico);  
    }  
}
```