

Programación Orientada a Objetos.

Postulado de programación estructurada:

PROGRAMA = ALGORITMO + ESTRUCTURAS DE DATOS

Paradigma Estructurado 1963. Wirth Nicklaus

Modelar Sistemas de información - Aprender Técnicas.

PARADIGMA: Es la forma de ver las cosas.

Paradigma Estructurado	¿Orientado a?	Types (Tipos) Estructuras Modelos Algoritmos – Procedimientos Funcionamiento
------------------------	---------------	--

¿Por qué Orientado a Objetos?

A semeja la Realidad.

El Paradigma Orientado a Objetos, modela los objetos de la realidad, permite una transición mas natural del análisis al diseño.

Desarrollo Orientado a Objetos: Análisis, diseño e implementación basados en la identificación, y organización de objetos mas que en un lenguaje de programación.

Metodología Orientada a Objetos:

- Uso de modelos para elaborar un diseño independiente del lenguaje de programación.
- Construcción de modelos (Objetos).

Objeto: *“Una abstracción de una cosa o concepto de la realidad que tiene identidad, estado y comportamiento; pertenencia unidad o modulo con una estructura u organización.”*

En UML, una instancia de una clase que encapsula estado y comportamiento. Mas informalmente, un ejemplo de una cosa. (1).

Un objeto consta de atributos (características) y métodos(operaciones).

CLASE : Es una generalización de objetos. Descripción de los conjuntos de objetos que comparten los mismos atributos, operaciones relaciones y semántica.

MOLDE DE OBJETOS(*Tipo de objetos*)

Entonces un objeto es una instancia de una clase.

Ej:

Clase: Persona

Objeto: William Mendoza Ropdríguez

```
public class Persona{  
  
    //atributos  
    String nombre  
  
    //Creador – Constructor  
    Persona(String nombre){  
        this.nombre=nombre;  
    }  
}
```

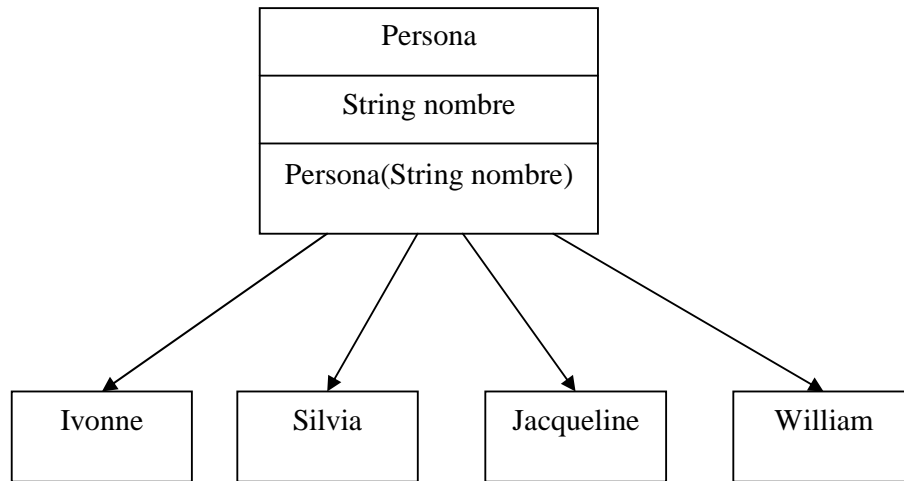
Instanciamiento de nuevos objetos:

- .
- .
- .

Persona P1, P2, P3,P4;

```
P1 = new Persona (Ivonne);  
P2 = new Persona (Silvia);  
P3 = new Persona (Jacqueline);  
P4 = new Persona (William);
```

- .
- .
- .



Métodos: Operación. Descripción (propiedad) del objeto (Atributos).

Constructor: Igual que la clase.

Analizador: get_nombreAtributo(). Devuelve información.

Modificador set_nombreAtributo(Parametros) Cambia información.

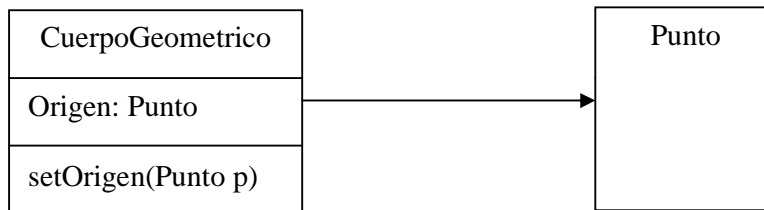
Diagrama de Clases: Modelo gráfico de las relaciones que se están definiendo.

- Objetos y sus características (Atributos) y/o Servicios que presta (Metodos).
- Navegabilidad, Cardinalidad, y nombre de la red.
- Roles.

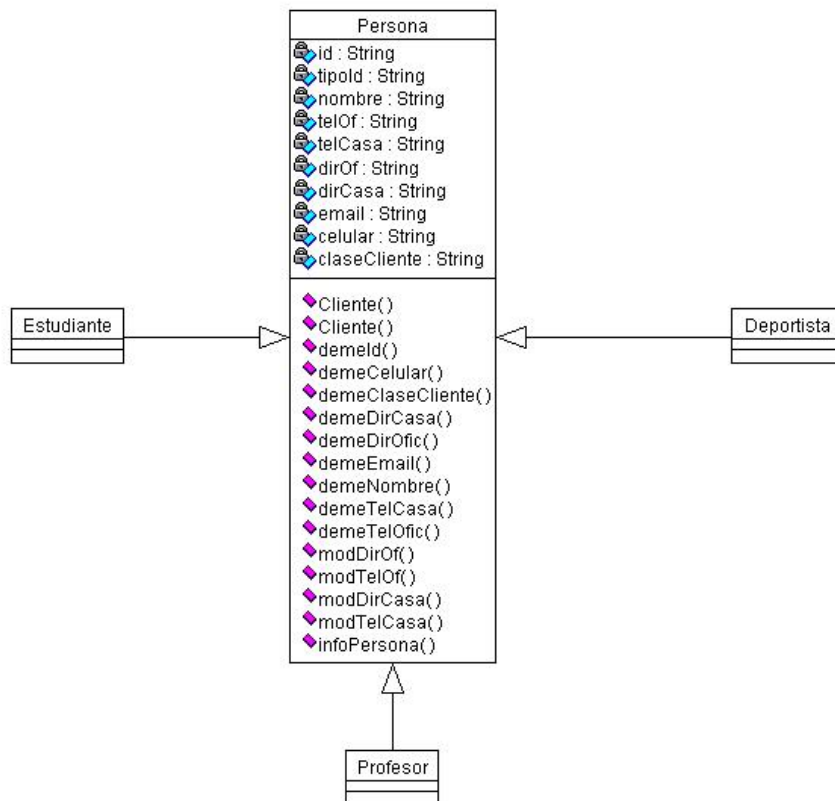
Modelos de Relaciones. Una relación es una conexión entre dos cosas.

- Dependencia.
- Generalización.
- Asociación.

Dependencia: Cuando una clase “USA” a otra en algunas de sus operaciones.



Generalización: Relación entre una clase general y una clase mas particular



Nodo Raiz: Es aquella clase que no tiene superclase.

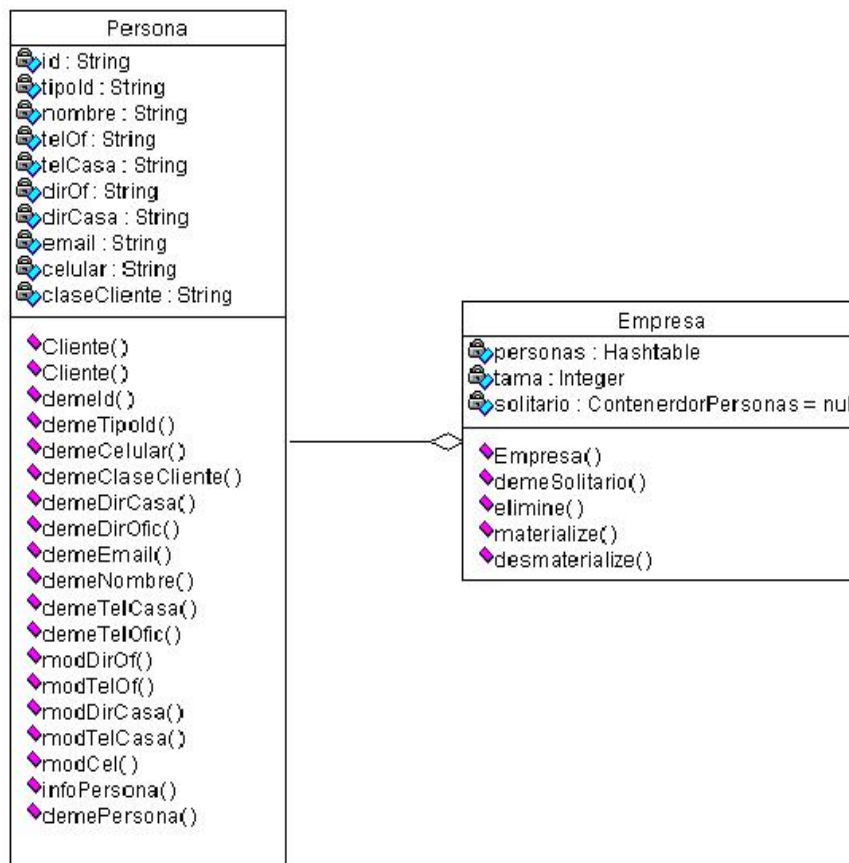
Nodo Hoja: Es aquella clase que no tiene subclases.

“La Generalización es llamada tambien Herencia”.

Herencia Simple: Cuando la correspondencia de generalización es de una sola clase.

Herencia Multiple: Cuando la correspondencia de generalización es de mas de una clase. Java no acepta herencia múltiple.

Asociación: Relación estructural que especifica que objetos de una clase estan conectados con objetos de otra con un proposito comun.



Forma Parte de

¿Qué es una Excepción?

Es una denominación en ejecución de una situación que no es controlada directamente por el programa (En un metodo).

Exception - Runnable - Throwable

CONCEPTOS BASICOS

- Conjunto de Entidades : Una Entidad es una <<cosa>> u <<objeto>> en el mundo real distinguible de todos los demas objetos. Un conjunto de entidades del mismo tipo que comparten las mismas propiedades y atributos. (Clase).
- Conjunto De Relaciones: Una relación es una asociación entre diferentes entidades. Un conjunto de relaciones es un conjunto de relaciones del mismo tipo.

CARACTERISTICAS de la POO.

Encapsulamiento Herencia y Polimorfismo

- Encapsulamiento:** Solo el objeto es aquel que “Se ensucia las manos”, con sus atributos.
- Que solamente el objeto puede acceder sus atributos através de sus métodos.
 - Tambien conocido como “Ocultamiento”.

“Se que hace pero no como lo hace”.

Importante : *No romper el encapsulamiento.*

Herencia: Relación entre una cosa general (Superclase ó Padre) y una cosa más especifica (Subclase o hijo).

La subclase hereda TODASlas características de la superclase (Atributos y métodos).

Para escribir un método, este debe ser igual que la firma (tipo de retorno, nombre y parámetros) de la superclase.

Overriding: Sobre Carga de método; cuando se implementa un método heredado para que se realiza este en lugar del que está en la superclase. Se tiene que llamar igual y con los mismos números y tipos de parámetros. “La misma firma”.

Polimorfismo: Una misma operación se refiere a un comportamiento distinto de las clases. Puede haber más de un método implementando una operación.

POLIMORFISMO = MUCHAS FORMAS

*“Cuando se manda un mensaje al objeto (llamado de método) para llevar a cabo una operación el método que la implementa es seleccionado **Polifórmicamente** en tiempo de ejecución.”*

Pueden tenerse diferentes operaciones con la misma forma en diversos niveles de herencia.

Overloading : Sobrecargado de un método.

Son métodos que se llaman igual pero con diferente número y tipos de parámetro, para hacer lógicamente la misma operación pero en procedimiento particular a los atributos en la misma clase.

setOrigen (int x, int y)

setOrigen (Punto p)

setOrigen (double angulo, double radio)

Abstracción : Se refiere a enfocarse en los aspectos esenciales e inherentes de una entidad ignorando sus propiedades accidentales (específicos).

Enfocarse en lo que un objeto es esencialmente y en las responsabilidades o comportamientos que pueda tener.

Clase Abstracta: Superclase con una definición general dando solo sus subclases. Podrán especificar las características de esta con sus respectivas definiciones.

NO SE PODRA INSTANCIAR UN OBJETO DE TIPO ABSTRACTO

Método Abstracto: Aquel método que define de una manera general un comportamiento pero que debe ser implementado en las subclases heredadas..

Resumen realizado por:

William Mendoza Rodríguez

*Basado en los notas de clase del curso Programación orientada a objetos 1
Escuela Colombiana de Ingeniería "Julio Garavito".
Profesor Ivan Fuentes Quiroz.
II Semestre de 2002*

REFERENCIAS

- (1) Lrman, CRAIG. UML Y PATRONES *"Una introducción al análisis y diseño orientado a objetos y al proceso unificado."*2da Edición. PEARSON-PRENTICE HALL 2003
- (2) Notas de clase del curso PROGRAMACIÓN ORIENTADA A OBJETOS 1 (para optar el título de Ingeniero de Sistemas), dictada en la *ESCUELA COLOMBIANA DE INGENIERIA "Julio Garavito"*. En el primer semestre del año 2002, Dirigido por el profesor IVAN FUENTES Q.