Introducción al Paradigma de Objetos

Clase 1

Roberto Andrés Villa
ravilla.educacionit@gmail.com

Introducción al Paradigma de Objetos

Domina el Paradigma Orientado a Objetos. Aprende a pensar, analizar y modelar de una forma diferente, y lograr así mejorar tus desarrollos de programación en distintos lenguajes. Adáptate rápidamente al mercado laboral comprendiendo esta revolucionaria forma de idear sistemas de información.

Clases 1 2 3

Hablá con tus compañeros de clase

Incorporarse a la comunidad FB

Objetivos

En este módulo verás:

- Paradigmas y Paradigmas de programación.
- Concepto de modelo y modelo Orientado a Objetos.
- Concepto de clase y detección.
- Concepto de instancia u objeto.
- Atributos y métodos.

- Programa
- Videos
- Contenido
- Laboratorio
- Descargas
- Etc

Modalidad del curso

Este curso tiene como objetivo **comprender y realizar el Análisis Orientado a Objetos** de un sistema, y no la implementación de los objetos en un Lenguaje de Programación.
La codificación de los objetos se enseña en los cursos de las carreras JAVA, .NET, PHP, todos con uso de PC.

Paradigma

¿Qué es un paradigma?

Un **Paradigma** es un modelo o patrón en cualquier disciplina científica.

Un **Paradigma de Programación** es una propuesta tecnológica que es adoptada por una comunidad de programadores cuyo núcleo central es incuestionable en cuanto a que unívocamente trata de resolver uno o varios problemas claramente delimitados.

El **Paradigma de Programación Orientada a Objetos** es la implementación de un Paradigma de Programación

Un poco de Historia...

Un poco de historia...

Programación Lineal

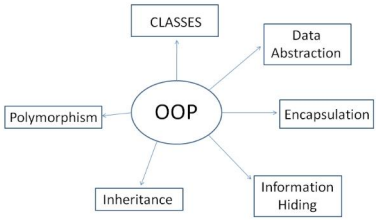
```
1. Hacer una variable igual a 0
2. Sumar 1 a esa variable
3. Mostrar la variable
4. Si la variable es 100 -> terminar, Si_no -> saltar a 1:
```

Programación Estructurada

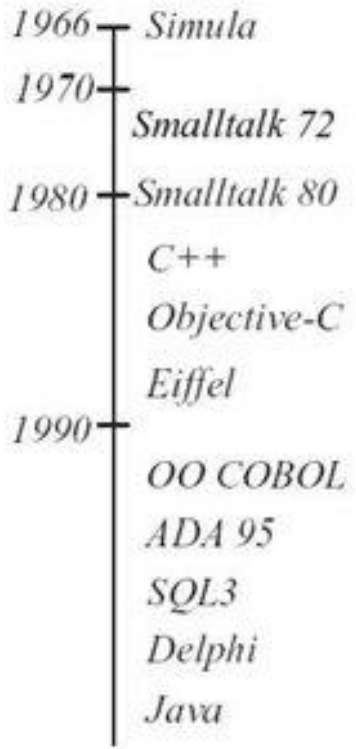
Concepto de función

```
Hacer una variable igual a 0
Mientras que sea menor que 100 -> sumar 1 y mostrarla
```

Programación Orientada a Objetos



Un paso hacia adelante...



Beneficios del Software Orientado a Objetos

- Altamente Escalable
- Fácil de Mantener
- Fácil de Reutilizar
- Muy Simple

¿Qué es un modelo?

Un modelo es una **abstracción** de la realidad

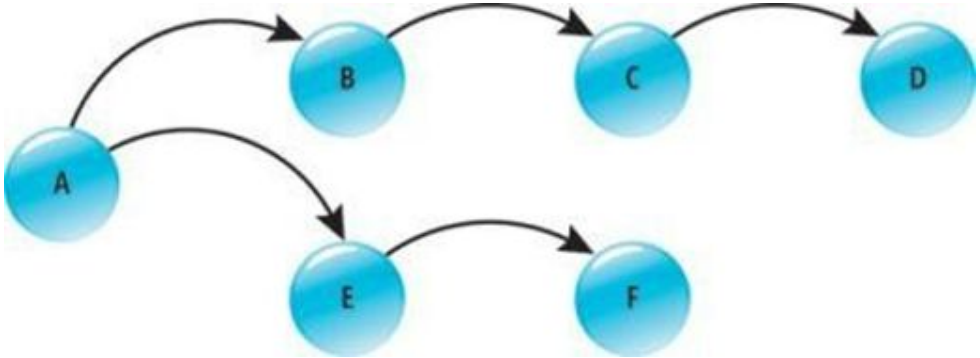
Es una **simplificación de la realidad**, con el objetivo final de pasar del modelo a producto real.

Los Sistemas de Información deben ser modelados previo a ser construidos


OBJECT ORIENTED
PROGRAMMING

El Modelado Orientado a Objetos

Consiste en interpretar un sistema como partes independientes que se comunican entre sí



Las partes independientes se denominan **Objetos**

La comunicación entre los objetos se realiza a través de **Mensajes**

¿Qué es una Clase?

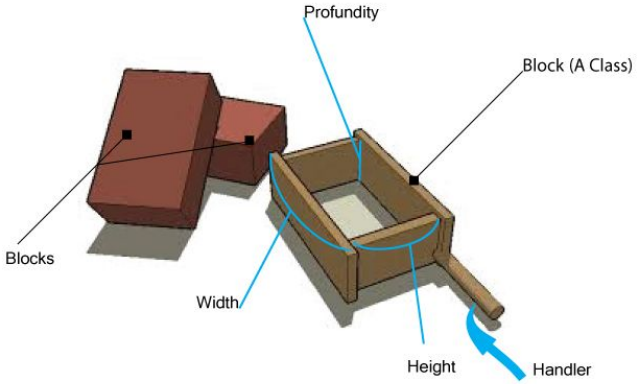
Es una plantilla, es un molde que permite construir objetos.

Representa ideas del mundo real, **en forma genérica**.

Dentro de un sistema, las clases suelen detectarse como **sustantivos en singular**

Poseen atributos y métodos

Ejemplos de clases: Auto, Empleado, CajaDeAhorro, Alumno

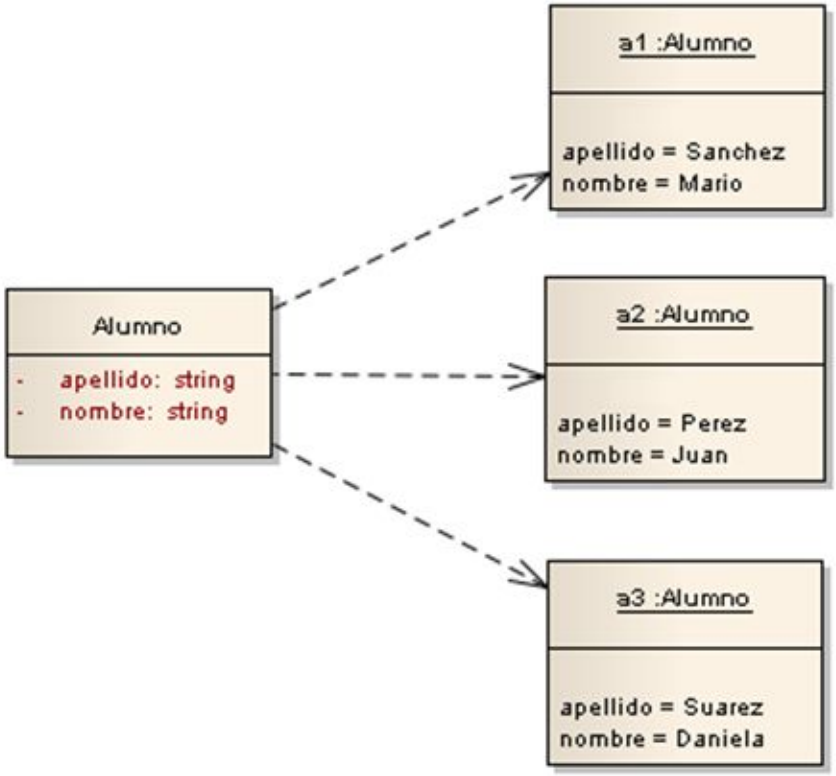


¿Qué es un objeto?

Un objeto es una la **instancia de una clase**, podemos decir que el objeto **representa algo en particular**

Poseen un **estado** (de acuerdo a sus atributos)

Poseen un **comportamiento** (realizan operaciones de acuerdo a sus métodos)



¡Manos a la obra!

De la siguiente lista, detectar cuales representan una **clase**

- Casa
- Auto
- Azul
- Acelerar
- Apagar
- Carpa
- Audi TT RS 2016
- Apagado
- Persona
- Submarino
- Objeto
- Contar
- Color
- Copiar
- Botella
- Vaso con agua
- Computadora
- Silla

TIP: las clases se detectan como sustantivos en singular

¡Manos a la obra!

De la siguiente lista, detectar cuales representan una **clase**

- **Casa**
- **Auto**
- Azul
- Acelerar
- Apagar
- **Carpa**
- Audi TT RS 2016
- Apagado
- **Persona**
- **Submarino**
- **Objeto**
- Contar
- **Color**
- Copiar
- **Botella**
- Vaso con agua
- **Computadora**
- **Silla**

Ejercicio #1 - Detección de Clases

Con los conocimientos que tenemos del modelado orientado a objetos, identificar las clases iniciales y necesarias para construir un sistema de gestión bancaria

TIP: las clases se detectan como sustantivos en singular

Ejercicio #1 - Solución

Banco – Sucursal – GrupoFinanciero – Servicio –

ClientePyme – ClienteCorporacion – ClienteIndividuo

CuentaComercial – CajaDeAhorro – CuentaCorriente

DirectorGeneral – DirectorRegional – DirectorDeSucursal

TIP: las clases se detectan como sustantivos en singular

¿Qué son los Atributos?

Son **características** que posee una clase

Son variables contenidas y establecidas por los objetos, y

normalmente cuentan con un tipo de dato asociado

Las atributos de una clase definen las características de sus

objetos

Las clases definen los atributos, y los objetos “los completan”

¿Qué es un tipo de dato?

Es la forma de describir y/o almacenar un dato

- Los tipos de datos numéricos más conocidos son: **int, long, float, double**
- Los tipos de datos de tipo caracter más conocidos son: **String, char**
- Para valores true/false el tipo de dato utilizado es **boolean** y para fechas se utiliza **Date**

Pueden ser otras clases!

Ejercicio #2 - Detección de Atributos

A partir de las clases detectadas anteriormente, identificar los atributos que considere necesarios en cada clase.
Cada atributo deberá tener establecido un tipo de dato. Deberá identificar al menos tres atributos por clase.

Banco – Sucursal – GrupoFinanciero – Servicio – ClientePyme – ClienteCorporacion – ClienteIndividuo – CuentaComercial – CajaDeAhorro – CuentaCorriente – DirectorGeneral – DirectorRegional – DirectorDeSucursal

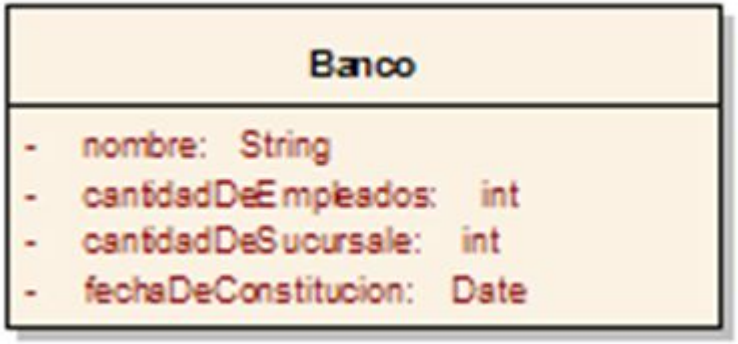
TIPS

- Los atributos son características de una clase
- Los tipos de datos a utilizar para este ejercicio son: **String**, **int**, **long**, **float**, **double**, **Date**

PREGUNTA

¿Donde ubicarías los atributos cantidadDeEmpleados, numeroDeCuenta y edad?

Ejercicio #2 - Solución



Sucursal: nombre, numero, direccion, cantidadDeEmpleados, cantidadDeClientes

ClientePyme: razonSocial, direccion, fechaDeAlta, **cuentaCorriente**

ClienteCorporacion: razonSocial, direccion, fechaDeAlta, cuentaCorriente

ClienteIndividuo: nombre, apellido, direccion, fechaDeAlta, cajaDeAhorro

Ejercicio #2 - Solución

Servicio: nombre, descripcion, fechaDeAlta
GrupoFinanciero: nombre, descripcion, fechaDeAlta

CajaDeAhorro: moneda, numero, cbu, saldo
CuentaCorriente: moneda, numero, cbu, saldo, **giroEnDescubierto**

DirectorGeneral: nombre, apellido, fechaDeNacimiento, fechaDeIngreso
DirectorRegional: nombre, apellido, fechaDeNacimiento, fechaDeIngreso
DirectorDeSucursal: nombre, apellido, fechaDeNacimiento, fechaDeIngreso

¿Qué es una operación?

Las operaciones son acciones contenidas en una clase, y definen su comportamiento

Dentro de un sistema, las operaciones suelen detectarse como **verbos.**

Desde la perspectiva de Diseño y Programación, se denominan **Métodos**

Desde la perspectiva de Análisis, se denominan **Operaciones**

Puede tener opcionalmente valores de entrada (Parámetros) y valores de salida (Valores de Retorno)
Procedimientos (no retornan un valor) vs. **Funciones** (retornan un valor)

¿Qué es un valor de entrada o parámetro?

Los parámetros son valores enviados a una operación
La operación toma los parámetros como **valores de entrada**, y así puede realizar las acciones necesarias
Todos los parámetros deben tener un tipo de dato asociado

-Método encender() - sin parámetros
-Método acelerar(int) - recibe como parámetro la cantidad de “km” a acelerar

-Método frenar(int) - recibe como parámetro la cantidad de “km” que debe bajar de velocidad

¿Qué es un valor de salida o valor de retorno?

El **valor de salida** de una operación es un **valor retornado** por la operación luego de realizar cierto procesamiento

Los valores de entrada son **datos**, y los valores de salida son considerados **información**

Todos los valores de salida deben tener un tipo de dato asociado

Es posible retornar un **único valor de salida**

Ejercicio #3 - Detección de operaciones

A partir de las clases y los atributos detectados en los ejercicios anteriores, identificar al menos dos operaciones que realiza cada clase

TIP: las operaciones son verbos

PREGUNTA: en que clase ubicarías las siguientes operaciones ?

- informarSaldo()
- informarSucursales()
- informarClientes()
- depositarDinero(monto),
- informarDatosDeCuenta(nroDeCuenta)

Ejercicio #3 - Solución

Banco:
informarCantidadDeEmpleados(),
informarCantidadDeSucursales(), **informarSucursales(),**
calcularFacturacionAnual(), calcularFacturacionMensual(),
informarClientes()
Sucursal:
informarDireccion(), informarCantidadDeEmpleados,
calcularFacturacionMensual(), calcularFacturacionAnual(),
informarClientes(), informarDatosDeCuenta(nroDeCuenta)
ClientePyme: infomarMovimientosEnCuentas(), informarDatos()

Ejercicio #3 - Solución

ClienteCorporacion: infomarMovimientosEnCuentas(), informarDatos()

ClienteIndividuo: infomarMovimientosEnCuenta(), informarDatos()

CajaDeAhorro: **infomarSaldo(),** extraerDinero(monto), **depositarDinero(monto)**

CuentaCorriente: **infomarSaldo(),** extraerDinero(monto), **depositarDinero(monto)**

Servicio: informarNombre(), informarDescripcion()

Ejercicio #3 - Solución

GrupoFinanciero: informarFechaDeAlta(), informarDescripcion()

DirectorGeneral: informarDatos()

DirectorRegional: informarDatos()

DirectorDeSucursal: informarDatos()

¿Preguntas?

¡Gracias!

(Por favor apagar computadoras, revisar elementos personales…)

Apéndice - Codificación de una clase

```
class Banco {

    // Atributos aquí

    // Métodos aquí

}
```

Apéndice - Codificación de Atributos

```
class ClientePyme {

    // Atributos aquí
    String razonSocial;
    String direccion;
    Date fechaDeAlta;
    CuentaCorriente cuenta;

    // Métodos aquí

}
```

Apéndice - Codificación de Operaciones

```
class CajaDeAhorro {
        // Atributos aquí
        float saldo;
        // Métodos aquí
        void informarSaldo() {
                // Imprime el atributo saldo
                print(saldo);
        }

        float obtenerSaldo(){
                // Retorna el saldo
                return saldo;
        }

}
```

Apéndice - Codificación de Operaciones

```
class CajaDeAhorro {
        // Atributos aquí
        float saldo;
        // Métodos aquí
        void depositarDinero(float unMonto) {
                // Actualiza el valor del atributo saldo
                saldo = saldo + unMonto;
        }

        void extraerDinero(float unMonto){
                // Actualiza el valor del atributo saldo, NO controla si monto > saldo
                saldo = saldo – unMonto;
        }

}
```

Laboratorio #1

Dado el siguiente texto, detecta las clases existentes atributos y métodos.

Sea una empresa dedicada al alquiler de CD-ROMs de audio. Dicha empresa tiene un local de atención al público donde están expuestas las carátulas de los CDs más demandados y las últimas novedades, aunque también existen listados en papel de todos los títulos que se podrían alquilar. Cuando un cliente solicita en alquiler un título, se comprueba si hay ejemplares disponibles y si el cliente no tiene problemas por ejemplares no devueltos, quedando constancia de la fecha de alquiler y la fecha máxima de entrega; de forma que cuando el cliente devuelva el ejemplar se podrá comprobar si se le tiene que imponer una sanción. Cada cliente puede solicitar una relación de los CDs que ha alquilado previamente.

Laboratorio #2

Dado el siguiente texto, detecta las clases existentes atributos y métodos.

Se tienen CLIENTES de los que se guarda un número de cliente, nombre, apellidos, lista de teléfonos, fax y correo electrónico. Los clientes realizan PEDIDOS. (Un pedido no puede ser realizado por dos clientes simultáneamente). Cada pedido tiene un número de pedido, una fecha asociada y una persona de contacto. Cada pedido aglutina varias LÍNEAS DE DETALLE, cada una con una cantidad y una referencia a un artículo. Los ARTÍCULOS tienen un descriptor, un identificador de familia y un identificador de modelo. Varias líneas de detalle correspondientes a uno o varios pedidos (bien en su totalidad, bien en parte) constituyen un ALBARÁN. Los albaranes contienen una fecha de entrega, una dirección de entrega y el nombre y apellido del receptor. Varias líneas de detalle correspondientes a uno o varios albaranes (bien en su totalidad, bien en parte) constituyen una FACTURA, la cual contiene un número de factura, una fecha de cobro y un modo de pago.

Garantía de Aprendizaje

- Recuperar clases perdidas
- Cambio de clase
- Suspender la capacitación
- Volver a cursar sin costo para, por ejemplo, reforzar conceptos, volver a reforzar las prácticas, etc.