# PROGRAMACION ORIENTADA A OBJETOS

En Java tenemos que declarar variables al principio de nuestro programa



Las operaciones aritméticas solo se pueden hacer con el mismo tipo de variable, para que el resultado sea del mismo tipo de las variables.

Pero podemos cambiar el tipo de variable, en caso de necesitar transformar de int a double.

Table

Description automatically generated

# Tipos de datos

* Numéricos
* Strings
* Booleanos

También hay conjunto de datos, colecciones o arrays. Pero para eso están las variables. En Java, tenemos que definir que tipo de dato vamos a guardar.

string nombre= ‘Walter’;

TIPO DE DATO

NOMBRE VARIABLES

VALOR DE LA VARIABLE

Table

Description automatically generated



# Fecha del parcial 17 de noviembre

# Tema

En las clases las manejamos singular, manejamos la creación de variables con VariablesConMayusculas.

El método main, es donde ejecutamos o mejor dicho es lo primero que se ejecuta.

PSVM = public static void main

Seamos descriptivos, no vamos a crear variables con una letra.

Sout= system.out.println()

# Vamos a tener dos tipos de clases

La clase integer permite usar métodos integer que ya están usados.

**Métodos = funciones**

Cuando vamos a usar lo primitivo y cuando no lo vamos a usar. Si no necesito métodos de la clase integer, lo almaceno en métodos primitivos ( int, boolean, double).

**Cuando nos piden una fecha, vamos a manejar la fecha como un STRING, OJITO, STRING**. En caso de que en el trabajo nos toque hacer, usamos localdate y otros métodos para nombrar las variables.

Hay otras formas de concatenar, seria = . Concat(variable);

Recordamos que en **el *sout = system.out.println (“Acá va el mensaje”+ “\n Hola, esto iría en el renglón de abajo o lina de abajo del código, para que no nos muestre todo en una misma linea”);***

Viendo como la interacción con la librería de , ***import javax.swing.JOptionPane***, lo que hacemos en esto es poder mostrar diálogos en nuestro programa de JAVA, los mensajes como en ventanas emergentes.

Anexar ejemplo:

***Entero = Integer.parseInt(JOptionPane.showInputDialog(“Digite un numero:”));***

Lo que esta haciendo anteriormente es convertir ese dialogo en un dato tipo entero, utilizando el tipo de variables que no son primitivas.

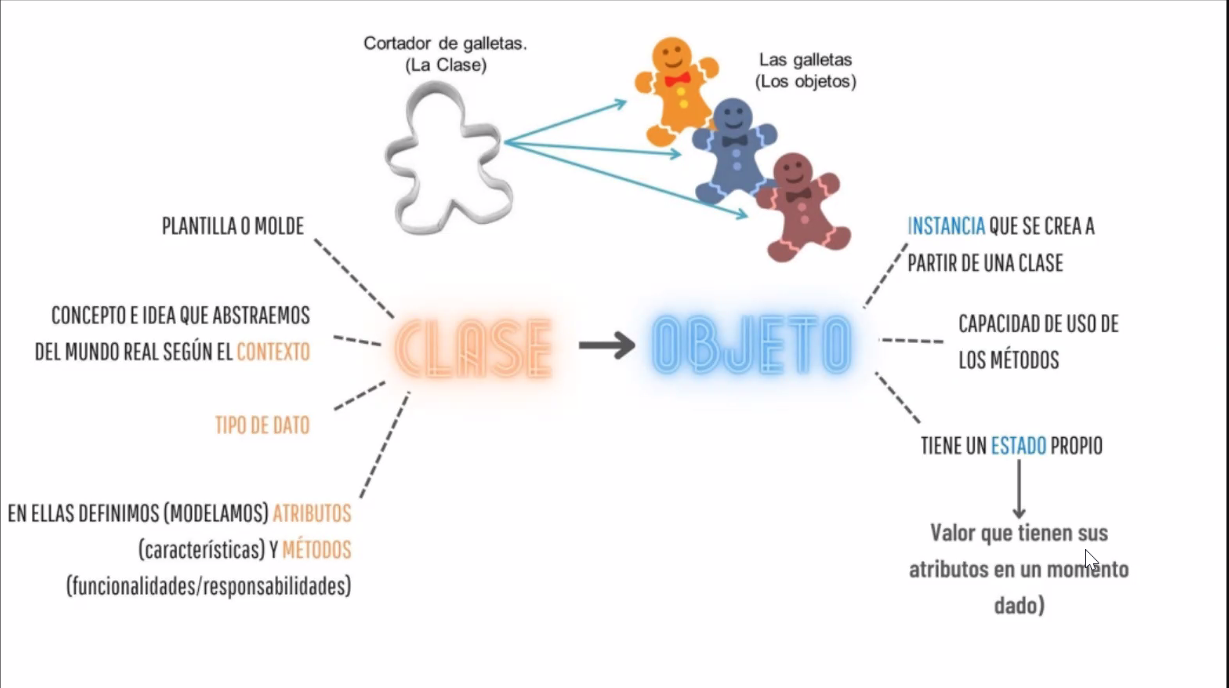
# METODOS : parámetros y argumentos

Los parámetros son declaraciones que se hace en el método los argumentos son los que lo invocamos.

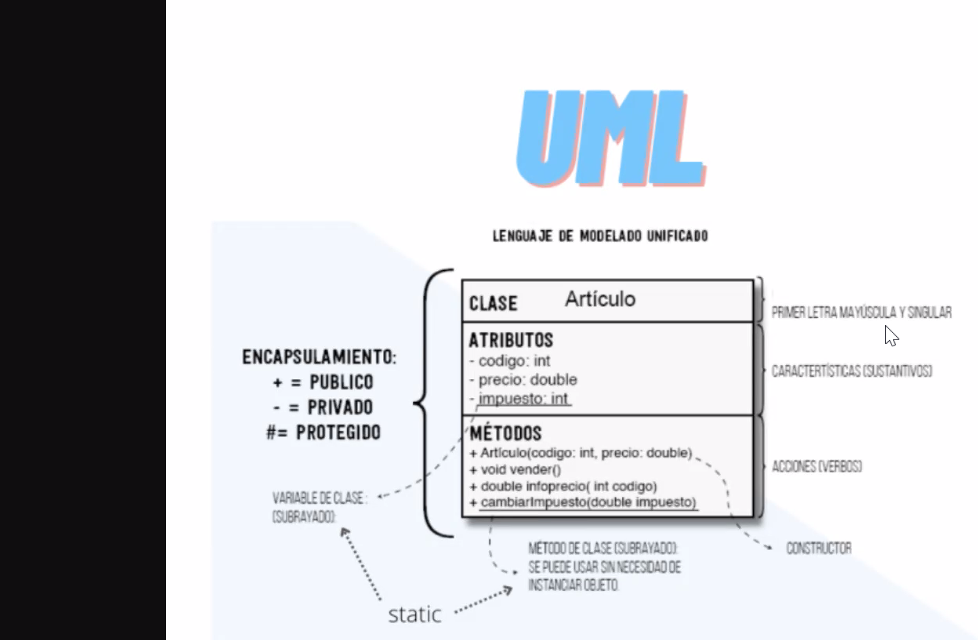
NOTA: a la hora de trabajar en el mismo archivo con métodos, creamos con static el método o función, en caso de que sea vacio static void, en caso de que sea en otra clase ponemos public y void si esta vacia, que no nos retorne nada de nada.

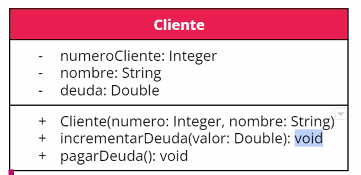
# CLASES Y OBJETOS

La clase es el croquis de los objetos, es el plano. Los atributos tratemos de que sean mas específicos con mas contexto, el modelado tiene que ser enfocado



Vamos a hacer todo código con UML,uppercomer case, la clase.





ESCRIBIMOS el integer como variable primitiva.

Text

Description automatically generated

Los getters y los setters, no recuerdo por que lo copie, pero los getters

# LA POO

Las famosas relaciones entre los objetos, vamos a encontrar dos relaciones importantes, **tiene un**

## Relación usa un

Cuando una clase usa a otra clase, una clase es parámetros de un método de otra clase.

Diagram

Description automatically generated

## Asociación Simple (asociación)

Nota, no importa que tipo de asociación, ene l código se ven exactamente iguales, no vamos a poder identificar en el código, solo lo vamos a ver en el UML.

Es cuando cuenta con un tiene un, cuenta con un atributo, establece relación es parte de eso, pero no es de

, va de quien la tiene hacia donde vaGraphical user interface, application

Description automatically generated

## Agregación (Es parte de )

Además de que una clase tiene un objeto o muchos objetos de la otra clase, esos objetos son parte de la clase que la contienen, ósea que si una clase deja de existir, la otra clase puede seguir existiendo.

Diagram

Description automatically generated

## Composición dueño ( es dueño de)

Una clase depende de otra para existir, una cuenta pertenece al banco, entonces la cuenta no puede existir si no hay un banco.

## Método constructor

El método constructor Es para cuando tenemos dos datos definidos anteriormente.

Setters, es para modificar los datos privados de mi objetos.

El método constructor es un método especial de una **de una clase**  que se invoca siempre que se crea un objeto de esa **clase.**

### Ocurren 3 cosas cuando se crea un objeto

1. Se asigna memoria para el objeto.-
2. Se inicializan los atributos de ese objeto
3. Se invoca al constructor de la clase que puede ser uno entre varios.

### Características de los constructores

1. Tienen el mismo nombre de la clase.
2. No devuelve ningún valor.
3. Debe declararse público.

## Herencias

En la clase Padre lo identificamos como general (atributos o métodos), cosasque se pueden poner en las hijas, que son más específicas.

Se necesitan una relación de clase herencia.

### Que es la sobre carga de métodos

La firma de nuestro método va a cambiar, lo que va a cambiar para que podamos sobrecargar el método seria la cantidad de parámetros, orden o tipo de parámetros.

En la clase cursar tenemos un método igual en ambos, pero son diferentes atributos o parámetros que se le piden a ese método

Lo que tiene que cambiar es lo que sucede dentro de los paréntesis.

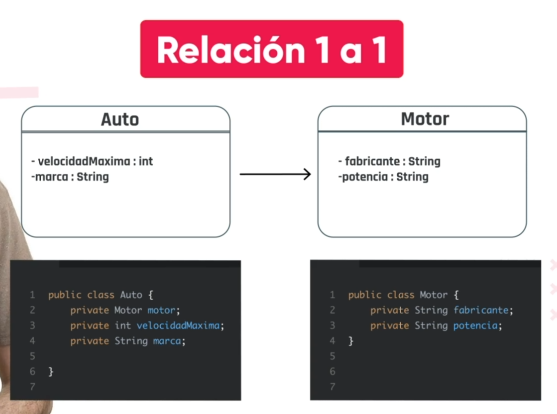
Diferencia entre sobre escritura y sobrecarga

### Sobre escribir

Hace lo mismo del Padre, pero de una forma diferente, o lo mismo mas otra actividad.

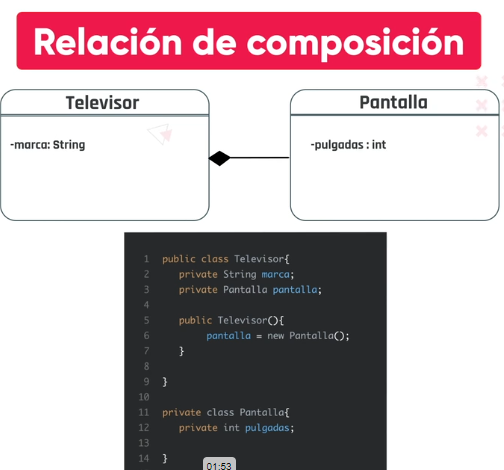
## Aplicando en java las relaciones

Las relaciones se representan como atributos



Hay varias relaciones en en java que las tenemos en UML y en java. Entendiendo la explicación anterior lo que hacemos es, crear un atributo en la clase, pero ese atributo tendrá el nombre de la clase como en UML con upper case, después en miniscula diciendo que es un atributo de mi clase principal.





Tener siempre cuidado en como puedo poder la cardinalidad. Definir un atributo tipo clase.

**Legajo:** el legajo es la identificación de la persona pero en ese establecimiento.

## Ejercicio integrador clase 9

El invierno ha llegado y un comercio de electrodomésticos desa realizar un relevamiento para detectar las necesidades de calefaccionb de las viviendas de una ciudad la que identifica con su nombre.

La necesidad de calefaccionde una vivienda se mide en calorías y dependerá de su tama;o (medido en m2), la cantidad de personas que viven en ella y los artefactos de calefacción instalados. Tener en cuenta que en una vivivienda pueden estar instalados ams de un artefacto.

Vamos a considerar solamente casas( viviendas comunes) y departamentos (viviendas en edificios)

Para los departamentos sabemos que:

Se necesitan 500 calorias por m2 mas 100 calorias por cada habitante.

Para las casas sabemos que

Se necesitan 700 calorias por m2 mas 150 calorias por cada habitante

Super clase o clase Padre: vivienda

Hijas : Casa, edificio y artefactoCalefaccion

Tengo que crear otro constructor vacio en mi super clase, para que la clase hija se conecte y sepa que es la hija, que se conecte a ese constructor vacio.

# Ternario

En un escenario similar a este, también podemos usar el operador ternario que funciona con el mismo concepto que if y else, sin embargo, la única diferencia es que necesitamos devolver un valor después de la prueba estrictamente en una sola línea.

A picture containing logo

Description automatically generated

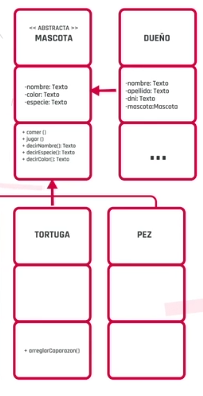
Tener claridad cuando yo llamo un método en otras clases.

Equals es para comprar las asignaciones de memoria.

1:23:22 clase

# Clase Abstracta

Cuan vemos algo en general, que necesitamos más especificidad. Acá tenemos la clase abstracta **Mascota\*,** pero como podemos definir a mascota como tal.



La clase que sea abstracta obliga a las hijas a heredar el método abstracto en la otra clase.

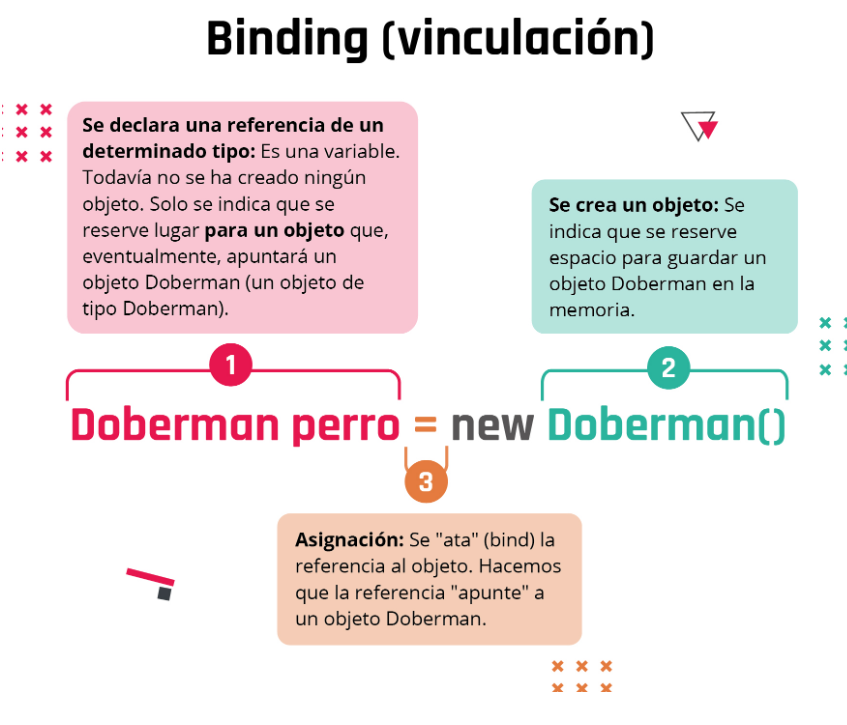
Text

Description automatically generated with medium confidence

Text

Description automatically generated

Así creamos un objeto de la clase dóberman en nuestra clase main de nuestro archivo principal.



Puedo tener un constructor por default que es prácticamente implícito en nuestro código y tener otro explicito. Entonces toca tener ya ambos CONSTRUCTORES explícitos en nuestro programa.

22:57 min clase 7 u 8