

— JAVA Y PROGRAMACIÓN ORIENTADA A OBJETOS

JAVA



— DISEÑO DE LENGUAJE

POR QUÉ JAVA ES COMO ES




MÓDULOS C++



ORGANIZACIÓN



SIN ESTÁNDARES



**UN EXTREMO POLARIZA HASTA
LLEGAR AL OTRO**

PACKAGES Y MAIN



CARPETAS COMO ESTRUCTURA



PUNTO DE ENTRADA *OBVIO*



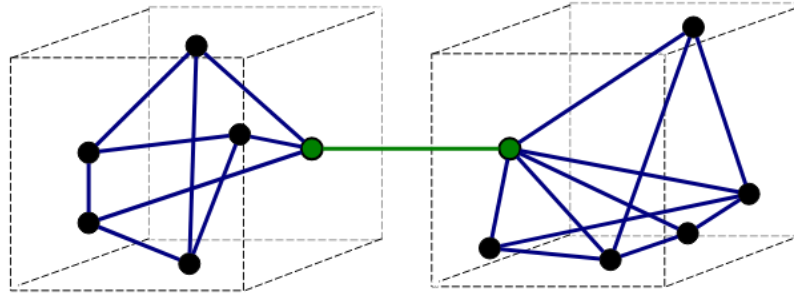
SIGUE USANDO CLI



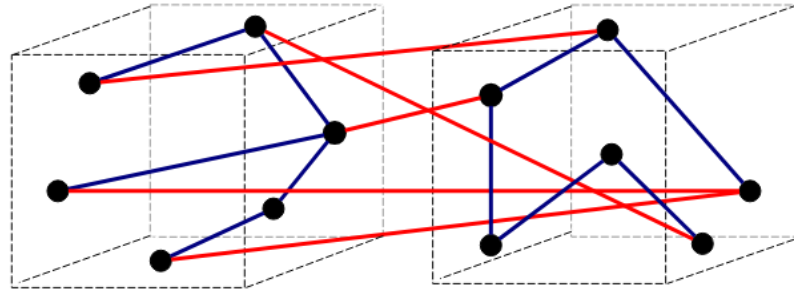
CLASE PRINCIPAL



VISIÓN DE MÓDULOS COHESIONADOS



a) Good (loose coupling, high cohesion)



b) Bad (high coupling, low cohesion)

**TIPADO FUERTE Y
EXPLÍCITO...**
A VECES DEMASIADO



SISTEMAS DE TIPADO



FUERTE VS DÉBIL



EXPLÍCITO



PRIMITIVOS VS USUARIO

GARBAGE COLLECTOR



**ADIÓS MALLOC,
HOLA INCERTIDUMBRE**

— **TRIVIA**



CREACIÓN



1991, JAMES GOSLING







1995, SUN SYSTEM

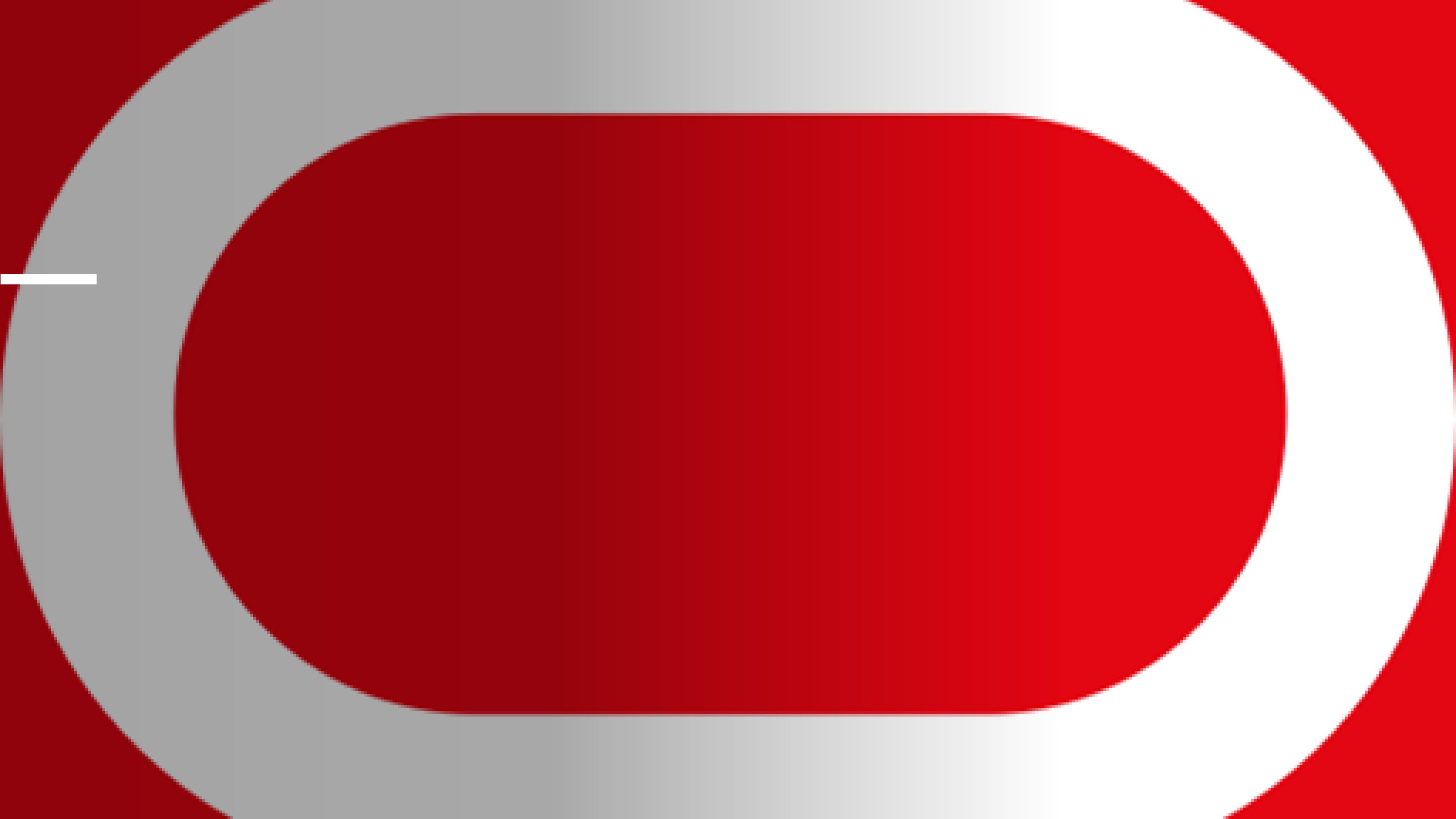




WORA
WRITE ONCE, RUN ANYWHERE



2009, ORACLE





2018, UNGA UNGA DINERO



AMOR, ODIO



ES MUY USADO



GARBAGE COLLECTOR



JVM, JRE



COMUNIDAD

**LAS UNIVERSIDADES LO ENSEÑAN
PORQUE LOS TRABAJOS LO PIDEN**



**LOS TRABAJOS LO PIDEN PORQUE
LAS UNIVERSIDADES LO ENSEÑAN**



**UNIVERSIDADES
LO ENSEÑAN...**



**TOTAL, QUE HAY MUCHA
DOCUMENTACIÓN Y USO**



VERSIONES



JAVA 8

PROGRAMACIÓN FUNCIONAL



JAVA 11

SEGURIDAD, HTTP Y RENDIMIENTO



JAVA 17

SEALED, PATTERN MATCHING



JAVA 21 VIRTUAL THREADS



JAVA 25

NEW MÁS EFICIENTE

PROGRAMACIÓN ORIENTADA A OBJETOS



— INTRODUCCIÓN



POR QUÉ



MAYOR ESTRUCTURA



APLICACIONES MÁS COMPLEJAS



REFLEJAR *INTERACCIONES*

**CLASS IS THE NEW
STRUCT**



**SIGUEN SIENDO STRUCTS
O SE LE PARECEN**



TODO ES UN OBJETO



**ENTRE LOS OBJETOS SURGEN
INTERACCIONES**



**ENTIDAD
OBJETO (INSTANCIA)
INTERACCIONES O RELACIONES**

— **PILARES**



OVERVIEW



**¿PILARES O PRINCIPIOS?
DA IGUAL**



BUEN DISEÑO DE SOFTWARE



**LOS PATRONES,
MÁS ADELANTE**



INTERFACES CONTRATOS ENTRE CLASES

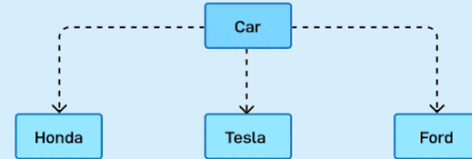


“NOS PONEMOS DE ACUERDO”

Object Oriented Programming

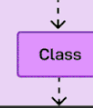
Abstraction

- Process of hiding implementation details and exposing only the functionality to the user.



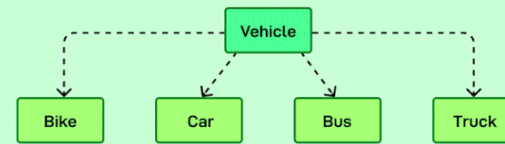
Encapsulation

- Process of wrapping code and data together into a single unit. It means each object in the code should control its own state



Inheritance

- Process of one class inheriting properties and methods from another class. Used to represent the IS-A relationship between objects



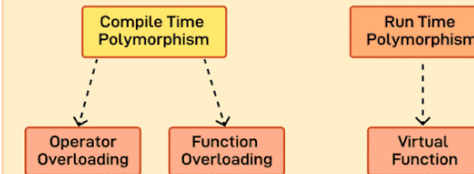
Base Class

Derived Class



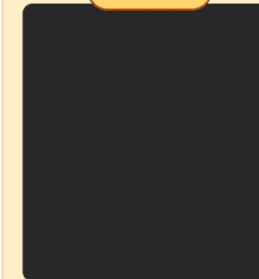
Polymorphism

- Ability to perform many things in many ways. When two types share an inheritance chain, they can be used interchangeably with no errors. This also means using parents like their children



Definition

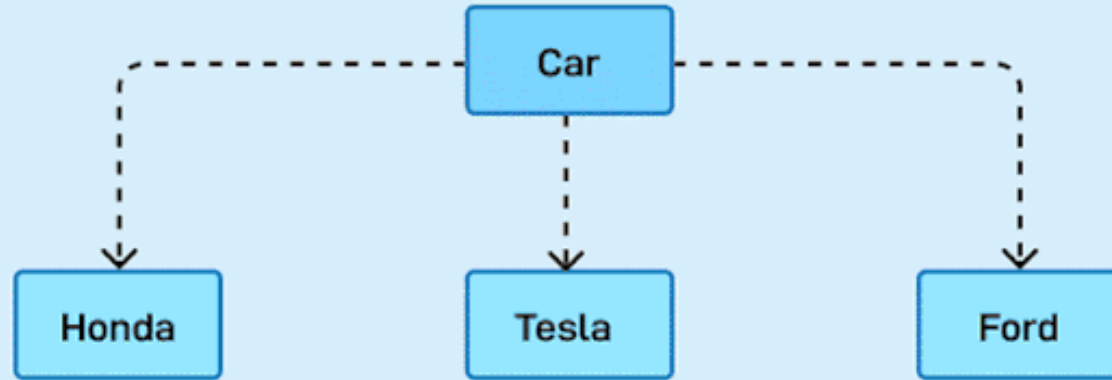
Usage



— ABSTRACCIÓN

Abstraction

- Process of hiding implementation details and exposing only the functionality to the user.



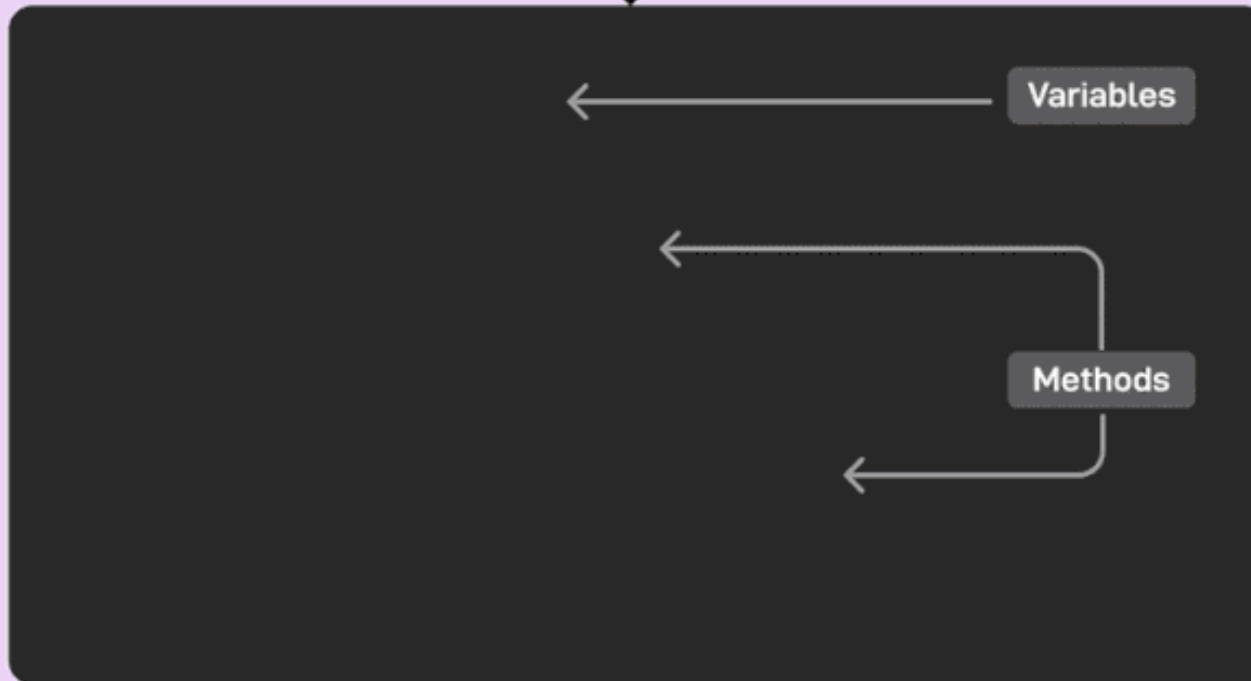
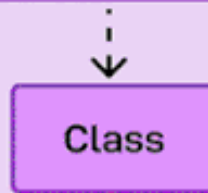
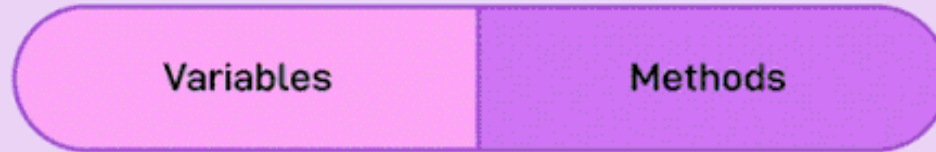
Abstract Class and Method

Providing Implementation

ENCAPSULACIÓN

Encapsulation

- Process of wrapping code and data together into a single unit. It means each object in the code should control its own state

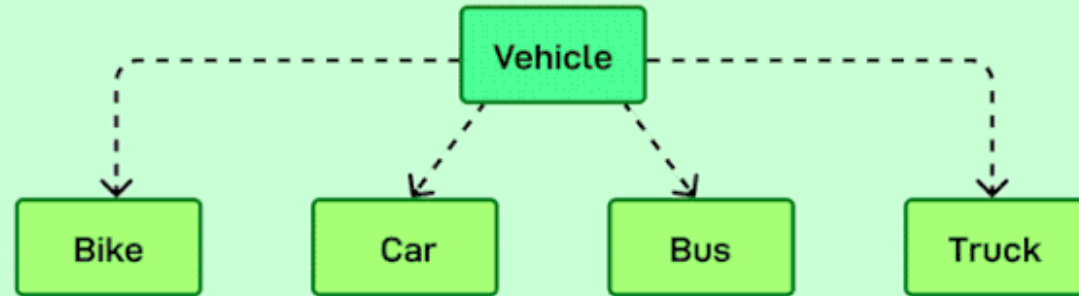




HERENCIA

Inheritance

- Process of one class inheriting properties and methods from another class.
Used to represent the IS-A relationship between objects



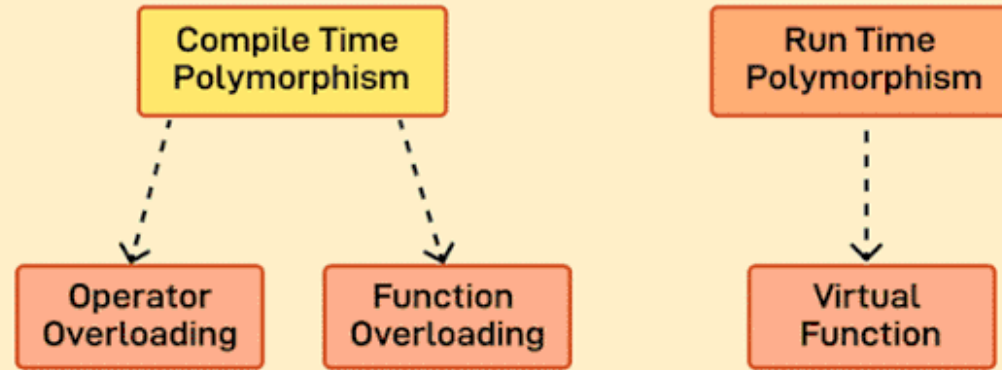
Base Class

Derived Class

POLIMORFISMO

Polymorphism

- Ability to perform many things in many ways. When two types share an inheritance chain, they can be used interchangeably with no errors. This also means using parents like their children



Definition



Usage




— SIGUIENTES PASOS



PLAYGROUND



GOOGLE COLAB NOTEBOOK



**HTTPS://COLAB.RESEARCH.GOOG
LE.COM/DRIVE/1T9U9RXDH6LYM6
GFEUSETIBZ-6ECMIMCN**



RECURSOS

WHAT EVERY PROGRAMMER
SHOULD KNOW ABOUT
OBJECT-
ORIENTED
DESIGN



DH

MEILIR PAGE-JONES

foreword by

Larry L. Constantine



REFACTORING.GURU

CODE SMELLS



REFACTORING.GURU

DESIGN PATTERNS



CONCEPTOS



HASHES



SERIALIZACIÓN



VALUE POOLS

POR QUÉ USAMOS EQUALS Y SIMILARES



SOLID

**SINGLE RESPONSIBILITY
OPEN/CLOSED
LISKOV'S SUBSTITUTION
INTERFACE SEGREGATION
DEPENDENCY INVERSION**



SOLO HACES UNA COSA



SOLO HACES UNA COSA
O EXTIENDES O CREAS NUEVO

SOLO HACES UNA COSA
O EXTIENDES O CREAS NUEVO
LO DE ABAJO POR LO DE ARRIBA

SOLO HACES UNA COSA
O EXTIENDES O CREAS NUEVO
LO DE ABAJO POR LO DE ARRIBA
INTERFACES DESGLOSADAS

SOLO HACES UNA COSA
O EXTIENDES O CREAS NUEVO
LO DE ABAJO POR LO DE ARRIBA
INTERFACES DESGLOSADAS
DEPENDENCIAS DESACOPLADAS

**SOLO HACES UNA COSA
O EXTIENDES O CREAS NUEVO
LO DE ABAJO POR LO DE ARRIBA
INTERFACES DESGLOSADAS
DEPENDENCIAS DESACOPLADAS**

CONCLUSIÓN





QUÉDATE CON

JAVA Y OOP

**ENDPOINTS, APIS, BASES DE
DATOS, ETC.**

**MÁS ADELANTE, DEJA QUE CALEN
LOS CONCEPTOS**

— QUIERO DESTACAR

QUÉ MONA VA ESTA CHICA
SIEMPRE PORQUE... *LA PLANTILLA*
ES CHULÍSIMA



GRACIAS POR TU TIEMPO