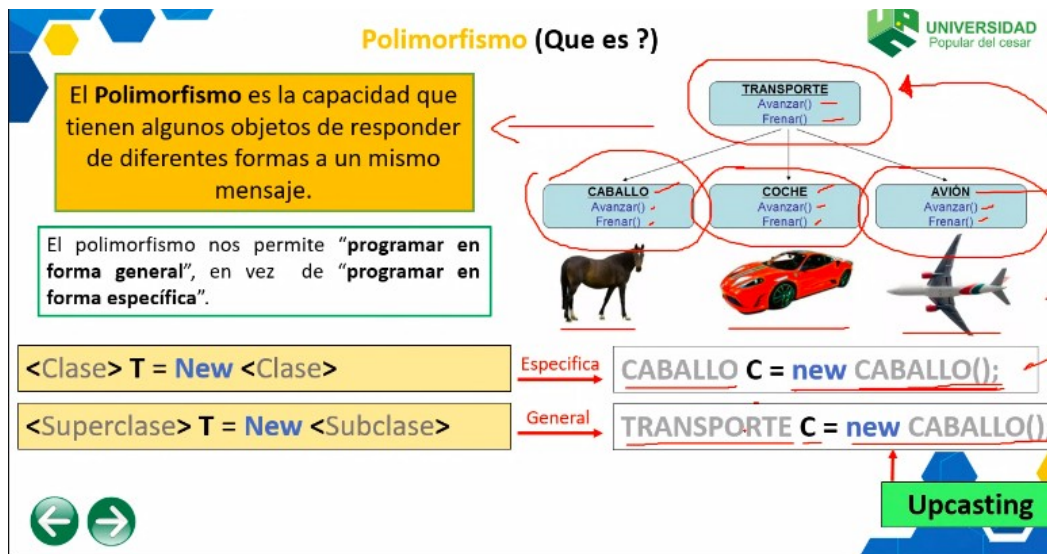


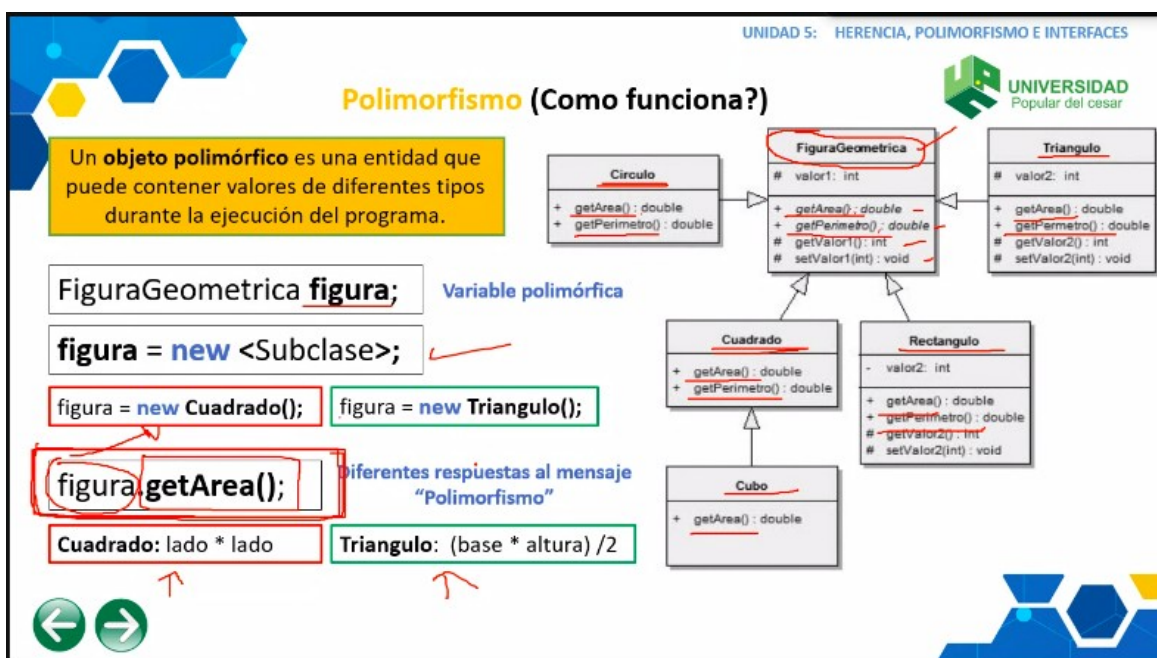
Polimorfismo

Qué es?

Es la capacidad que tienen algunos objetos para responder de diferentes formas a un mismo mensaje, el polimorfismo nos permite programar de forma general en vez de en forma específica, nos permite crear objetos de las clases específicas pero será del tipo genérico. (Es decir, se almacena en una variable de una superclase) Ej:



Se usa para programar de forma general, ¿Como funciona el polimorfismo?, un objeto polimorfo es un objeto que puede contener valores de diferentes tipos durante la ejecución del programa.



Clases y métodos abstractos

Clase abstracta:

Polimorfismo (Implementación?)


Clase abstracta

- ✓ No se pueden declarar objetos de clase abstracta
- ✓ Una clase abstracta contiene uno o más **métodos abstractos**
- ✓ Se debe declarar con la palabra clave **abstract**
- ✓ Una clase que contiene métodos abstractos debe declararse **abstracta**
- ✓ Su propósito es proporcionar una superclase apropiada, a partir de la cual puedan heredar otras clases y, por ende, compartir un diseño común.


Métodos abstractos

- ✓ Son métodos que no proporcionan implementaciones
- ✓ Un método abstracto tiene la palabra clave **abstract** en su declaración
- ✓ Cada subclase concreta de una superclase abstracta también debe proporcionar implementaciones concretas de los métodos abstractos de la superclase
- ✓ Los constructores y los métodos static no pueden declararse como **abstract**

UNIDAD 5: HERENCIA, POLIMORFISMO E INTERFACES



UNIVERSIDAD
Popular del Cesar



```

classDiagram
    class FiguraGeometrica {
        <<abstract>>
        +int dimension1
        +FiguraGeometrica()
        +FiguraGeometrica(int dimension1)
        +abstract double getArea()
        +abstract double getPerimetro()
    }
    class Triangulo {
        +int dimension2
        +Triangulo()
        +Triangulo(int base, int altura)
        +double getArea()
        +double getPerimetro()
    }
    class Cuadrado {
        +Cuadrado()
        +Cuadrado(int lado)
        +double getArea()
        +double getPerimetro()
    }
    FiguraGeometrica <|-- Triangulo
    FiguraGeometrica <|-- Cuadrado
    
```

Handwritten notes on diagram:

- Red circle around `<<Abstract>>` and `FiguraGeometrica` in the class name.
- Red circle around `Fig = C / T` with an arrow pointing to the `FiguraGeometrica` class.
- Red checkmarks and lines indicating inheritance and method calls.

- Si no se quiere heredar los metodos abstractos, la clase hija seria también una clase abstracta, toda clase que herede de una clase abstracta debe sobrescribir los metodos abstractos de la clase madre. Las clases abstractas deben por lo menos definir un método abstracto. Los metodos concretos proporcionan bloque de código, los metodos abstractos solo proporcionan la estructura.

Polimorfismo (Implementación?)

```

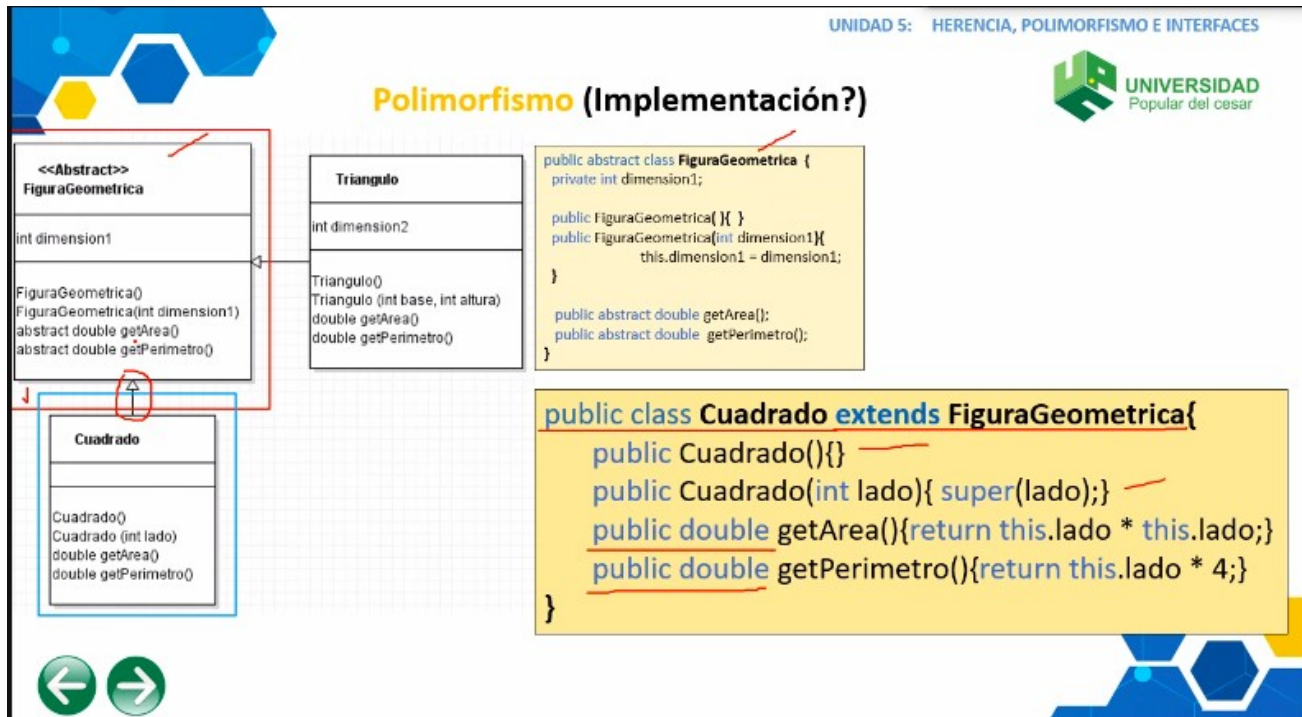
public abstract class FiguraGeometrica {
    private int dimension1;

    public FiguraGeometrica(){} }
    public FiguraGeometrica(int dimension1){
        this.dimension1 = dimension1;
    }
    public int getDim1(){return this.dimension1;};
    public abstract double getArea();
    public abstract double getPerimetro();
}
                
```

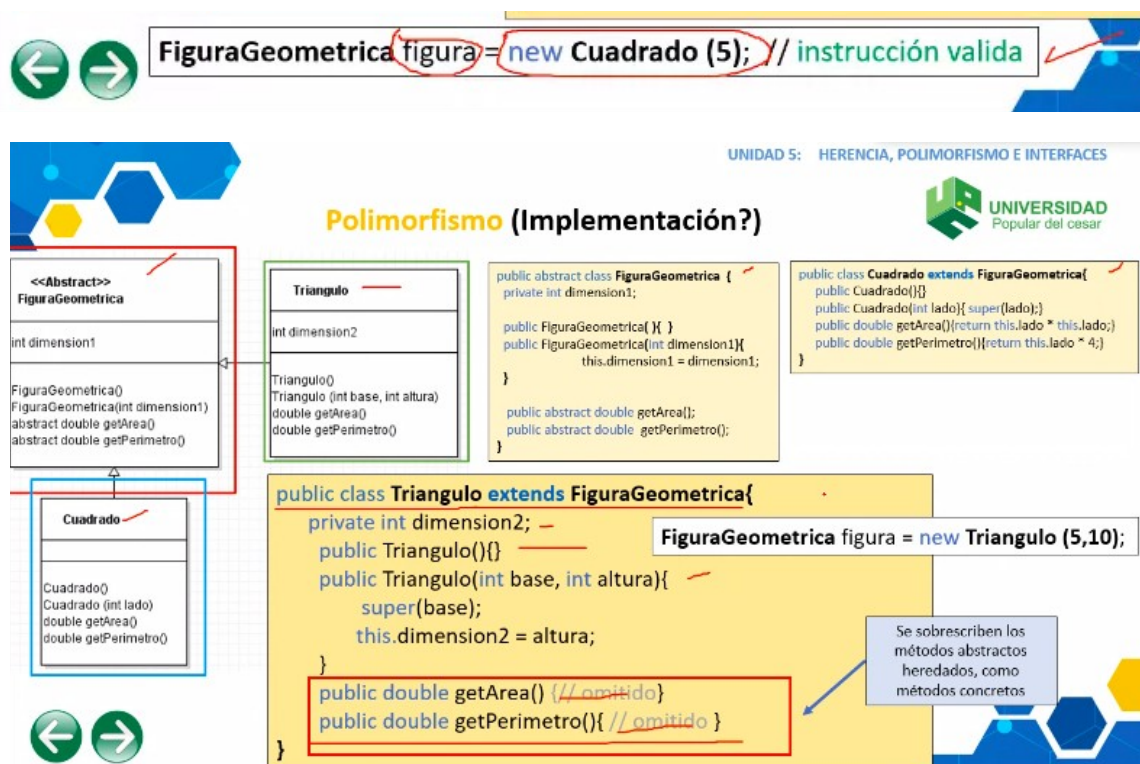
UNIDAD 5: HERENCIA, POLIMORFISMO E INTERFACES

UNIVERSIDAD
Popular del Cesar

Si se pueden definir variables de tipo clase abstracta, pero no instanciarlo con constructores.

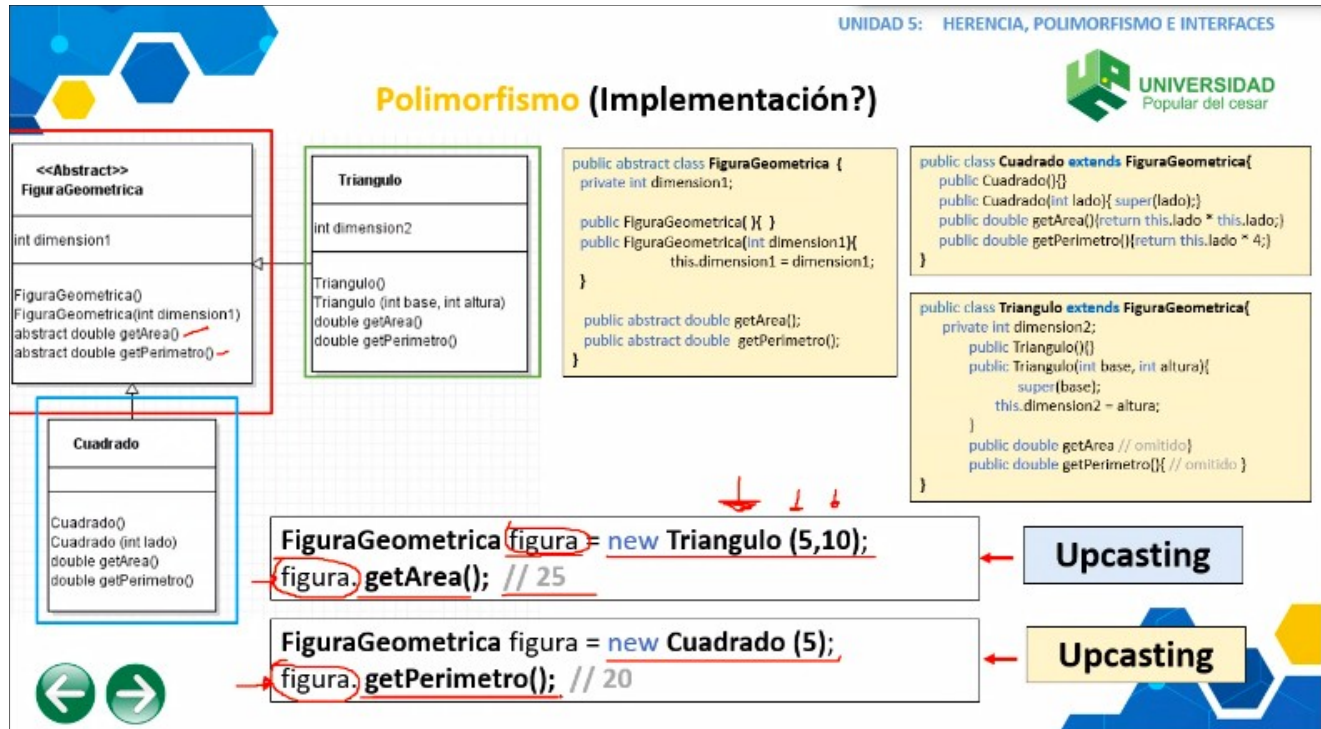


Upcasting: convertir un objeto de la clase hija en un objeto de la clase madre
Lowcasting: convertir un objeto de una superclase en clases específicas

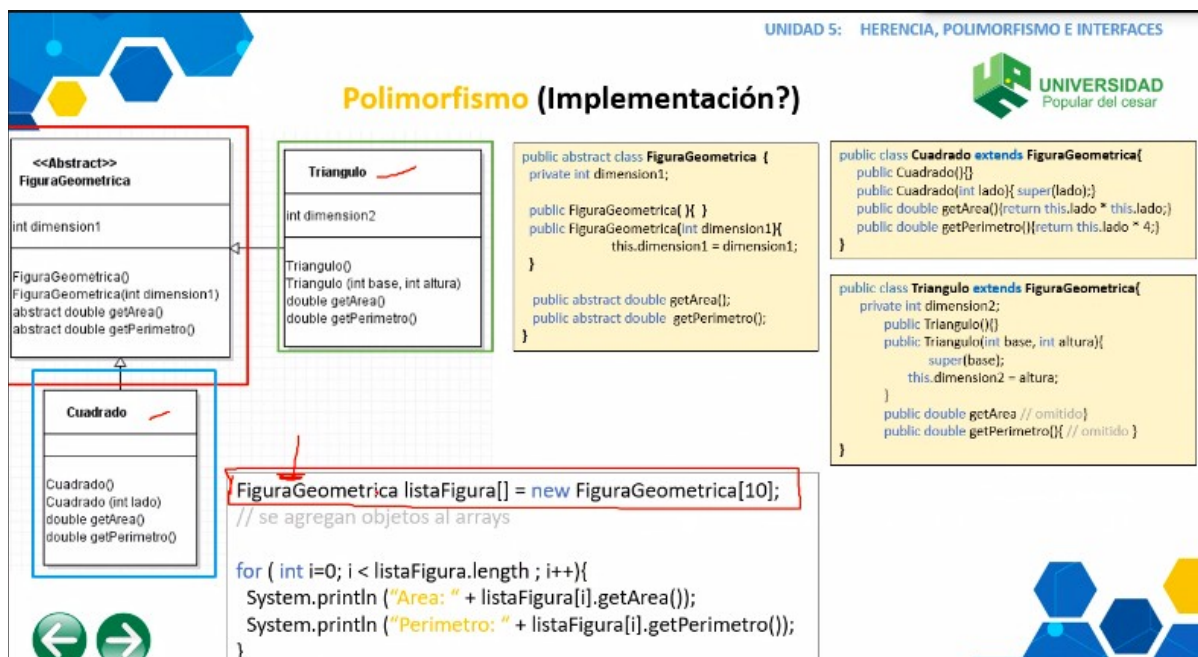


Se definen de la clase madre pero instancian con constructores de las clases específicas para que se sepa que los códigos de la clase madre se sobrescriben.

En tiempo de ejecución el compilador ejecuta los metodos implementados en la versión del constructor definido para un objeto.



Si creamos un arreglo de tipo de la superclase podemos guardar objetos de las clases hijas. Ejemplo: Los metodos get llaman a la versión del método get del objeto que este en ese momento, sea cuadrado o triangulo.



El polimorfismo nos permite extender la funcionalidad de nuestro aplicativo de forma sencilla o menos compleja.