

# PROGRAMACIÓN ORIENTADA A OBJETOS - 22951

**Universidad Industrial de Santander**  
**Escuela de Ingeniería de Sistemas e Informática**  
**Programa de Ingeniería de Sistemas**



#LaUISqueQueremos





Universidad  
Industrial de  
Santander

# Clases e instancias



# Generalidades POO



Universidad  
Industrial de  
Santander



Estructura con esquema de objetos – Joaquín Torres García

# Generalidades

- Problemas del software:
  - Tamaño.
  - Costo de mantenimiento.
  - Fiabilidad.
- Soluciones propuestas:
  - Modularidad.
  - Reutilizar el software.
  - Certificación de calidad.



# Tamaño del software

2012

Un DVD Player contiene aproximadamente 50 millones de líneas de código.

Un carro BMW tiene 50 dispositivos conectados en red.

Eclipse IDE contiene 2,5 millones de líneas de código

El kernel de Linux contiene 15 millones de líneas de código

Cada 7 años la cantidad de software se duplica  
<http://users.jyu.fi/~koskinen/smcosts.htm>



Universidad  
Industrial de  
Santander





# Costo de mantenimiento

Año	Porcentaje	Referencia
2000	90%	Erlikh (2000)
1993	75%	Eastwood (1993)
1990	60-70%	Moad (1990)
1980	50%	Lientz & Swanson (1981)

## Otros datos:

- 250000 millones de líneas de código seguían siendo mantenidas en el 2000 (Sommerville, 2000)
- 200000 millones de líneas de código COBOL en 2010 (Gartner Group)
- 70% de las aplicaciones financieras escritas en COBOL (Giga Information Group)

# Fiabilidad

- La agencia espacial europea invirtió 10 años y 7000 millones de USD.
- La nave se mantuvo en vuelo 39 segundos y avanzo 3 Km.
- Problema de software: un dato de 64 bits en una variable de 16 bits.
- La unidad de backup también fallo.

ARIANE 5



# Un Poco de Historia

- Funciones y procedimientos (FORTRAN – 60).
- Tipos de datos (Pascal y Algol - 70).
- Módulos (Datos y funciones - 80).
- Programación orientada a objetos:
  - Simula-67 (Bjarne Stroustrup – 67).
  - Smaltalk 70.
  - C++ (Bjarne Stroustrup – 83).
  - Eiffel (85).
  - Java (95).





# Noción de función



Universidad  
Industrial de  
Santander

```
int potencia(int base, int exp) {  
    int valor = 1;  
    for (int i = 1; i <= exp; i++) {  
        valor = valor * base;  
    }  
    return valor;  
}
```

```
int cuadrado(int base) {  
    int valor = potencia(base, 2);  
    return valor;  
}
```

```
int main() {  
    int ex1 = potencia(2, 3);  
    cout << ex1 << endl;  
    int ex2 = cuadrado(ex1);  
    cout << ex2 << endl;  
    return 0;  
}
```



# Paradigma procedimental

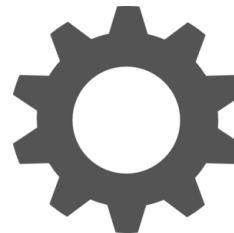
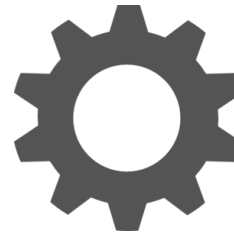
## Datos

radioC1 = 20,5

radioC2 = 18,3

radioC3 = 8,7

ladoC1 = 11,2



## Procedimientos

area\_circulo(float radio)

area\_cuadrado(float lado)

perimetro\_circulo(float radio)



```
float area_circulo(float radio) {  
    return 3.1416 * radio * radio;  
}  
  
float area_cuadrado(float lado) {  
    return lado * lado;  
}  
  
float perimetro_circulo(float radio) {  
    return 2 * 3.1416 * radio;  
}  
  
int main (void) {  
  
    float radioC1 = 20.5, radioC2 = 18.3, radioC3 = 8.7;  
    float ladoC1 = 11.2, ladoC2 = 10.0;  
  
    float areaC1 = area_circulo(radioC1);  
    float areaC2 = area_circulo(radioC2);  
    float areaC1 = area_circulo(radioC3);  
    float perimetroC1 = perimetro_circulo(radioC1);  
    float perimetroC2 = perimetro_circulo(radioC1);  
    float perimetroC3 = perimetro_circulo(radioC1);  
  
    float areaCua1 = area_cuadrado(ladoC1);  
    float areaCua2 = area_cuadrado(ladoC2);  
}
```

# Paradigma orientado a objetos



**:Circulo**

*radio: 20,5*



*area( )  
perimetro()*

**:Rectangulo**

*ancho: 10  
alto: 12,2*



*area( )*

**:Cuadrado**

*lado: 15,2*



*area( )*

**:Rectangulo**

*ancho: 15  
alto: 20*



*area( )*

**:Rectangulo**

*ancho: 5,5  
alto: 20,2*



*area( )*

**:Circulo**

*radio: 18,3*



*area( )  
perimetro()*



Universidad  
Industrial de  
Santander

# POO : Encapsulación y abstracción





# Objetos: 4 conceptos de base

- El objetivo principal de la noción de objeto es organizar programas complejos.
- La organización se basa en 4 conceptos:
  - Encapsulación.
  - Abstracción.
  - Herencia.
  - Polimorfismo.



# Noción de encapsulación

- El principio de encapsulación agrupa en el mismo objeto digital (concepto) los datos y el comportamiento (tratamiento de datos) específicos al objeto.
- Los datos encapsulados en un objeto son conocidos como atributos.
- El comportamiento definido en un objeto corresponde a los métodos (funciones, tratamientos).
- Un objeto está definido por sus atributos y sus métodos.



# Noción de encapsulación



**:Circulo**

*radio: 20,5*



*area( )  
perimetro()*

**:Rectangulo**

*ancho: 10  
alto: 12,2*



*area( )*

**:Cuadrado**

*lado: 15,2*



*area( )*

**:Rectangulo**

*ancho: 15  
alto: 20*



*area( )*

**:Rectangulo**

*ancho: 5,5  
alto: 20,2*



*area( )*

**:Circulo**

*radio: 18,3*



*area( )  
perimetro()*

# Un objeto...

- Tiene un **estado** (datos):
  - Puede decidir si modifica su estado como respuesta a un mensaje.
- Tiene un **comportamiento**. Entiende un conjunto de mensajes y responde a ellos. (Interfaz):
  - Lo que el objeto hace como respuesta a un mensaje puede depender de su estado.
- Tienen **identidad**. Dos objetos pueden tener el mismo valor en sus datos pero son objetos diferentes
  - Un mensaje dirigido a un objeto no modifica ni afecta ningún otro objeto



# Noción de Abstracción

Conjugar **abstraer**.

(Del lat. *abstrahĕre*).

1. tr. Separar por medio de una operación intelectual las cualidades de un objeto para considerarlas aisladamente o para considerar el mismo objeto en su pura esencia o noción.
2. intr. Prescindir, hacer caso omiso. *Abstraer DE examinar la naturaleza de las cosas*. U. t. c. prnl.
3. prnl. Enajenarse de los objetos sensibles, no atender a ellos por entregarse a la consideración de lo que se tiene en el pensamiento.

¶

MORF. **conjug. c. traer**.

*Real Academia Española © Todos los derechos reservados*





# Noción de abstracción

- Los objetos son interesantes cuando permiten un grado de abstracción.
- El proceso de abstracción consiste en identificar para un conjunto de elementos (objetos):
  - Propiedades (características) comunes a todos los elementos.
  - Mecanismos comunes a todos los elementos.
- La idea es realizar una descripción genérica del conjunto considerado:
  - Focalizarse sobre lo esencial y esconder los detalles.



# Noción de abstracción: ejemplo

- La noción de «objeto rectángulo» es interesante si se pueden asociar propiedades y mecanismos generales (válidos para todo el conjunto de rectángulos).
- Las propiedades de largo y alto son generales a todos los rectángulos (**atributos**).
- El mecanismo que permite calcular el área de un rectángulo (*área = alto \* ancho*) es común a todos los rectángulos (**métodos**).



# Noción de Abstracción



**:Circulo**

*radio: 20,5*



*area( )  
perimetro()*

**:Rectangulo**

*ancho: 10  
alto: 12,2*



*area( )*

**:Cuadrado**

*lado: 15,2*



*area( )*

**:Rectangulo**

*ancho: 15  
alto: 20*



*area( )*

**:Rectangulo**

*ancho: 5,5  
alto: 20,2*



*area( )*

**:Circulo**

*radio: 18,3*



*area( )  
perimetro()*

# Noción de Abstracción



- Generalidades del conjunto de círculos:
  - Atributos: radio.
  - Métodos: área, perímetro.
- Generalidades del conjunto de rectángulos:
  - Atributos: alto y ancho.
  - Métodos: área.
- Generalidades del conjunto de cuadrados:
  - Atributos: lado.
  - Método: área.



# Encapsulación y Abstracción

- El resultado del proceso de encapsulación y abstracción en POO es la noción de clase.
- Una clase define un nuevo tipo (categoría de objetos).
- Una realización particular de una clase es llamada instancia (u objeto).
- Una instancia puede ser referenciada por una variable.







Universidad  
Industrial de  
Santander

# Clases e instancias



# Noción de Clase

Una clase es un molde que sirve para crear objetos:

- Especifica la lista de atributos, es decir los datos que definen el estado de un objeto.
- Especifica los métodos que definen el comportamiento de todas las instancias de la clase.
- Una clase es un nuevo tipo de datos definido por el usuario que enriquece el entorno de desarrollo.

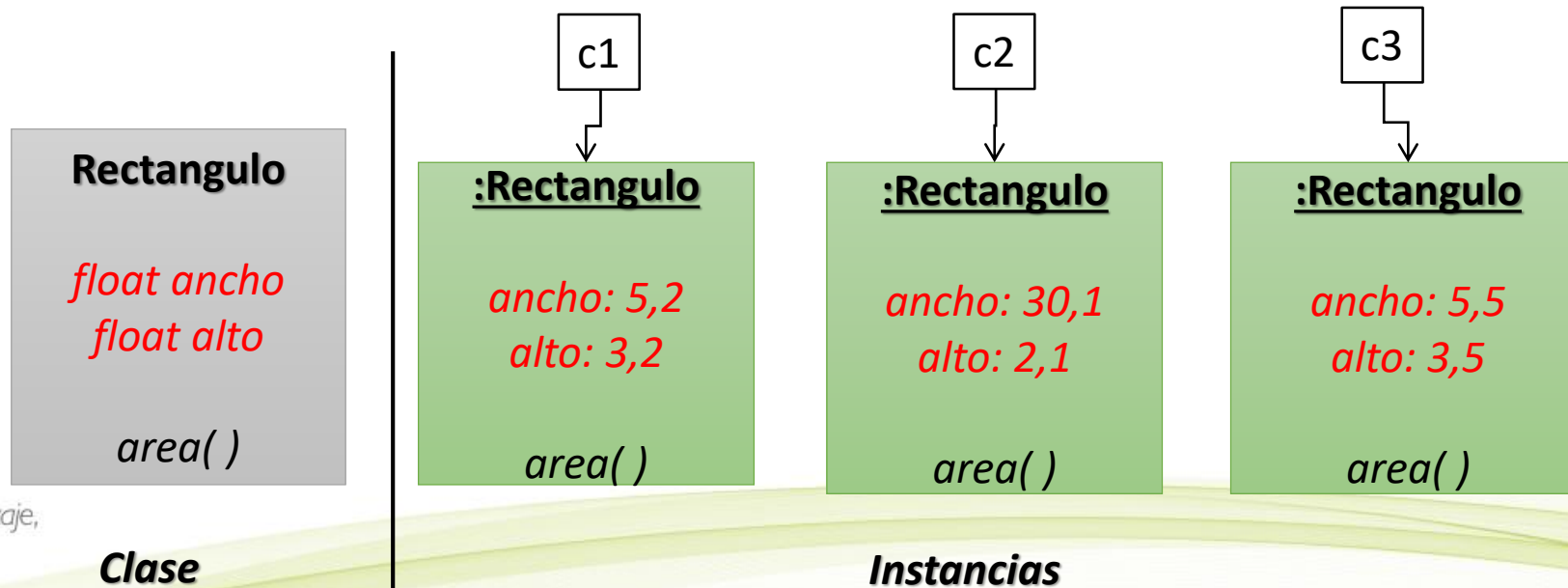


Universidad  
Industrial de  
Santander



# Noción de Instancia

- Una instancia es una realización de una clase.
- En la ejecución del programa una instancia es creada a partir de su clase (molde).
- Una clase puede tener múltiples instancias.
- La noción de clase existe durante la escritura del programa (existencia conceptual). La noción de instancia existe durante la ejecución (existencia concreta).



# Abstracción Clase - Instancia



www.shutterstock.com · 121845664



# Clases en Java

```
public class Rectangulo {
```

```
    public double ancho;  
    public double alto;
```

**Atributos**

```
    public void inicializar(double an, double al){  
        ancho = an;  
        alto = al;  
    }
```

```
    public double area() {  
        return ancho * alto;  
    }
```

```
    public double perimetro() {  
        return (2 * ancho) + (2 * alto);  
    }
```

**Métodos**



Universidad  
Industrial de  
Santander





# Usar la clase rectángulo

```
public class Principal {  
  
    public static void main(String[] args) {  
        Rectangulo r1;  
        Rectangulo r2;  
  
        r1 = new Rectangulo();  
        r1.inicializar(20.2, 10.5);  
  
        r2 = new Rectangulo();  
        r2.inicializar(10.4, 10.3);  
  
        double area1 = r1.area();  
        double area2 = r2.area();  
  
        System.out.println("Area Rectangulo 1: " + area1);  
        System.out.println("Area Rectangulo 2: " + area2);  
    }  
}
```

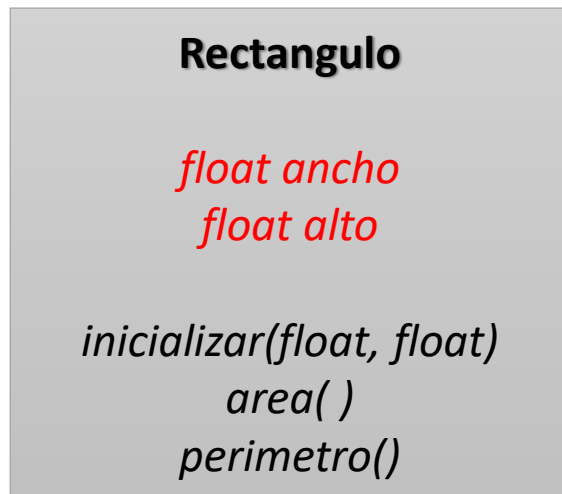


Universidad  
Industrial de  
Santander



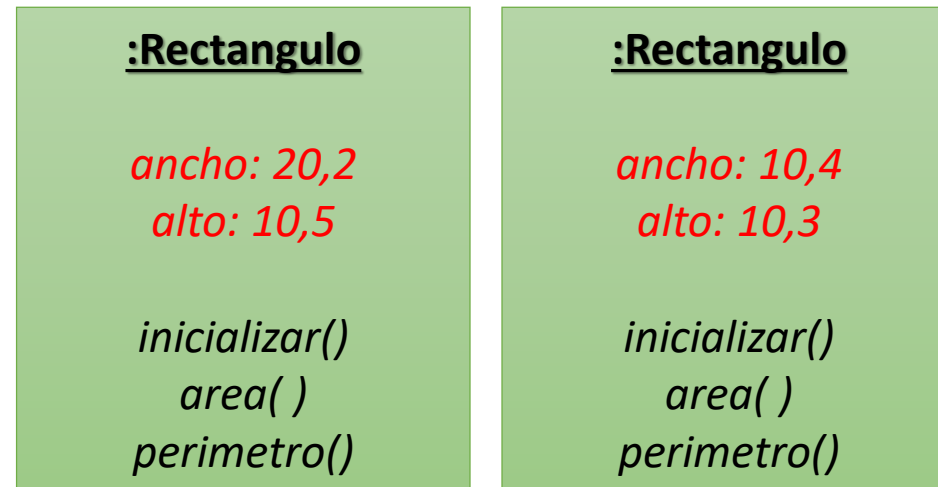
# Gráficamente

## Clase



Existencia conceptual  
(Código del programa)

## Instancias



Existencia concreta  
(Ejecución del programa)



# UML – Unified Modeling Language

- Permite gráficamente especificar clases y sus relaciones.
- Es un lenguaje de diseño no de programación.

Rectangulo
+ancho: Float +alto: Float
+inicializar() +area() +perimetro()



# Por qué usar Objetos?

- Para el desarrollador OO esta bien... pero no necesariamente mejor que el paradigma procedimental.
- Desde el punto de vista de ingeniería de software
  - Usa la noción de objeto para modelar cosas del mundo real y conceptos del dominio.
  - Los conceptos del dominio cambian menos rápido que los requerimientos.
  - La idea de la OO es poner juntas las cosas que evolucionan juntas.
- La complejidad y el cambio continuo es lo que hace a la ingeniería de software difícil e interesante. OO ayuda a gestionar complejidad y cambio.



# Lecturas

## Introduction to Programming Using Java

- Sesión 1 – Capítulo 1, Secciones 3, 4 y 5
- Sesión 2 – Capítulo 4, Secciones 1, 2, 3 y 4



Universidad  
Industrial de  
Santander



# Trabajo (30 minutos, en parejas)

- Aplicando el proceso de encapsulación y abstracción obtenga las clases
  - Triangulo
  - Cuadrado
  - Circulo
- Especifique gráficamente en UML cada clase
- Escriba el código Java de cada una de las clases
- Escriba un programa Java que calcule el área y el perímetro de: un circulo de radio 20, un cuadrado de lado 15 y un triangulo de alto 20 y base 10
- Escriba un programa en C++ que haga lo mismo que el anterior







Universidad  
Industrial de  
Santander



#OrgulloEISI

# iGracias!

#LaUISqueQueremos