

# Programación Orientada a Objetos

Ignacio Ramírez

Instituto de Ingeniería Eléctrica, Facultad de Ingeniería,  
Universidad de la República. Montevideo, Uruguay.

Montevideo, 2013

# Agenda

## 1 Programación orientada a objetos

- Definición
- Características

## 2 Lenguaje Java

- Características de Java

# Programación orientada a objetos

## Paradigma

programa interacción entre objetos

objeto atributos + comportamiento

clase molde de objeto

Ejemplo : pintar círculo

# Programación orientada a objetos

## Identidad de los datos

tipos de datos objetos

identificador referencia o “handle” (manija)

objeto atributos + comportamiento

copia de los atributos

Ejemplo : varias referencias

# Programación orientada a objetos

## Abstracción

objetivo separar funcionalidad e implementación: que hace? vs como lo hace?

motivación facilitar comprensión

mecanismo información visible e invisible

# Programación orientada a objetos

## Encapsulamiento

### objetivos

- ocultar la implementación (abstracción)
- cohesión: agrupar tareas comunes

### motivación

- facilitar abstracción
- coherencia interna

Ejemplo : cargar nafta

# Programación orientada a objetos

## Herencia

### Objetivos

- organizar jerarquía según especialización
- reutilizar compartir y extender funcionalidad
- flexibilizar generar “variantes”
- abstraer facilitar el encapsulamiento

Ejemplo : formas

# Programación orientada a objetos

## Polimorfismo

idea misma acción realizada de forma diferente  
interfaz “qué hacer” común a todas las variantes  
implementación “cómo hacerlo” (posiblemente) distinto entre variantes

Ejemplo : transformar forma

# El Lenguaje Java

## Características

**fabricante** Sun Microsystems

- tipo**
- orientado a objetos
  - concurrente
  - propósito general

## Consideraciones de diseño

**portable** multiplataforma

**internet** soporte interno para redes

**seguro** ejecución segura en máquinas remotas

**alto nivel** protección de datos, garbage collector, excepciones



# La máquina virtual de java

**objetivo** arquitectura virtual *única* sobre la que corren *todos* los programas Java

**propósito** portabilidad, interoperabilidad, facilitar desarrollo, mantenimiento e implantación

**implementación** particular a cada hardware, opaca al desarrollador

**instrucciones** idénticas en *todo* hardware: *bytecode*

**servicios** protección, excepciones, garbage collector, concurrencia

# La plataforma Java

JVM máquina virtual, ejecución de bytecode (comando `java`)

API bibliotecas estándar con funcionalidad pre-programada (*enorme!*)

SDK compilador (comando `javac`), etc (integrado en `eclipse`)

- Documentación
- javadoc, online
  - sun.com tutorial
  - libro: Eckel

Ejemplo : Compilación, ejecución, API

# Hola Mundo en Java

```
/**  
 * HolaMundo, primer programa en Java;  
 * muestra en pantalla la frase "Hola, mundo."  
 */  
// clase nombre del archivo  
public class HolaMundo  
{  
    // metodo principal  
    public static void main(String args[])  
    {  
        // muestra en pantalla  
        System.out.println("Hola, mundo.");  
    }  
}
```

# Tipos de datos

estricto tipado fuerte, operaciones controladas  
transformación promoción y *casting*  
tipos básicos primitivos (caracteres, números, lógicos) y *referencias*  
tipos complejos clases, interfaces, arreglos.

Ejemplo : JavaScript vs. Java

# Tipos de datos

## Clases

clase Object comportamiento común a todas las clases de Java  
ciclo de vida inicialización (constructor), destrucción (finalize)  
polimorfismo referencia a subclases

Ejemplo : clase object, ciclo de vida

# Tipos de datos

## Interfaces

**propósito** aspectos de comportamiento potencialmente comunes a clases heterogéneas

**implementación** conjunto de métodos abstractos, sin implementación

**polimorfismo** herencia múltiple de interfaces

Ejemplo : improvisar!

# Tipos de datos

## Arreglos

**definición** conjunto de referencias de mismo tipo (primitivos o referencias)

**características** largo fijo, verificación automática de límites

No hay arreglos multidimensionales: arreglos de arreglos.

Ejemplo : Hola a todos

# Excepciones: manejo de errores

**propósito** separar código de manejo de errores del código típicamente ejecutado

**mecanismo** interrumpen la ejecución normal, saltan a zona predefinida

**tipos** verificadas, no verificadas

Ejemplo : OutOfMemoryError, IOException, NullPointerException