

---

# Introducción al Lenguaje Java

## Programación Orientada a Objetos

San Salvador de Jujuy

UNJu – Facultad de Ingeniería

Ing. José Zapana



# Introducción a Java

---

- Lenguaje de Programación Orientado a Objetos
- **Características**
  - Independiente de la plataforma
    - Compilador no genera código ejecutable nativo a la máquina donde se ejecuta
    - Compilador produce un formato especial llamado *byte codes*
    - Para que bytescodes puedan ejecutarse en una máquina específica se necesita un intérprete en tiempo de ejecución. El intérprete es el que genera código de lenguaje de máquina entendido por la máquina
    - Ambiente de ejecución normalmente se conoce como Java Virtual Machine (JVM)
  - Recolector de basura
    - Memoria es liberada automáticamente cuando JMV lo determina



# Programando en Java

---

- Primer programa

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hola, amigo!!");  
    }  
}
```

- Compilador: javac : Traduce archivos **.java** en **.class** (bytecodes)
- Intérprete: java : dinámicamente ejecuta programa previa interpretación de bytecode a código de lenguaje de máquina
- Existen ambientes para la creación, edición, compilación y ejecución: Eclipse, Spring Tools, Visual Studio Code, etc



# Estructura de programas en Java

---

- Cada programa ejecutable en java consiste en una clase
- Aplicación
  - Al menos una de las clases debe contener el método main
  - Declaración de método main debe ser método estático (método asociado a clase no objeto)
  - Declaración public, cualquiera puede invocarlo



# Palabras clave en Java

---

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	<b>public</b>	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	<b>static</b>	<b>void</b>
char	finally	long	strictfp	volatile
<b>class</b>	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	

- Las palabras claves tienen significado especial y no pueden usarse como identificadores de variables ni clases ni métodos



# Tipos de datos

---

- Tipos primitivos
  - Tipos de datos simples definidos por el lenguaje de programación
    - En Java son 8
      - boolean (true o false), char (16 bits set caracteres unicode), byte (8 bits con signo), short (16 bits), int (32 bits), long (64 bits), float (32 bits punto flotante), double (64 bits punto flotante)
  - Tipos no primitivos se conocen como Tipos Objetos



# Constantes

---

- Declaración

– `public static final <type> <name> = <value>;`

- Ejemplos

```
public static final int DAYS_IN_WEEK = 7;
public static final double INTEREST_RATE = 3.5;
public static final int SSN = 658234569;
```



# Control de flujo

---

- Ciclos
  - for(), while(), do-while()
- Condicionales
  - If – else
  - switch (expresion){
    - case n: sentencias;
    - case m: sentencias;
    - ...
    - default: sentencias;}



# Tipos de datos Objetos

---

- Instancias de tipos de datos complejos llamados clases
- Entidad que contiene datos y comportamientos
- Existen variables, que almacenan datos dentro de un objeto
- Existen métodos dentro de objeto que representan su comportamiento
- Creación de objetos en Java mediante uso de `new`



# Variables de referencia

---

- Variables de tipos de objetos son llamadas **referencias**
- Las referencias no almacenan objeto, sino que almacenan la dirección de una ubicación en memoria del objeto
- Si se asigna una referencia a otra para referirse al mismo objeto, el objeto no es copiado sino que las dos referencias comparten el mismo objeto.
- Ejemplo
  - `Alumno a1 = new Alumno();`
  - ...
  - `Alumno a2 = a1;`



# Variables estáticas *static*

---

- Variables estáticas
  - También se les conoce como *class variables*
  - Variables que se asocian a clase no a objeto
  - Variable común a todos los objetos (variables compartidas entre todos los objetos de la clase)
    - Se definen como variable de clase con la palabra clave `static`
  - Ejemplo : Identificador de cuenta de CuentaBanco. Un número que identifique únicamente a dueño de cuenta.



# Ejemplo: variables estáticas

---

```
public class CuantaBanco {
    private String nombre;
    private int balance;
    private int Id;
    private static int proxIdDisponible = 1;

    /** Constructor, establece nombre dueño y balance inicial
    de la cuenta */

    public CuentaBanco(String nombre, int balance){
        this.nombre = nombre;
        this.balance = balance;
        this.Id = proxIdDisponible;
        proxIdDisponible++;
    }
}
```



# Métodos estáticos

---

- Algunos métodos no están asociados, en forma natural, con objetos particulares
  - Ejemplo, métodos en clase Math, sqrt, sin, cos, tan
- También podría darse el caso que nos gustaría llamar un métodos antes de crear un objeto
  - Tales métodos pueden ser declarados estáticos: el método no es parte de una instancia sino que de la clase
    - Se invocan enviando mensaje a la clase
    - No puede acceder referencia “this” o cualquier variable o método dentro de un método estático dado que no está asociado a un objeto



# Métodos estáticos - Ejemplo

---

```
public class FechaUtil() {  
    public static Date getNow() {  
        return new Date();  
    }  
}
```

...

```
Date fechaIngreso = FechaUtil.getNow();
```



# Nomenclatura java

---

- <http://www.oracle.com/technetwork/java/codeconventions-135099.html>