Introducción al Lenguaje Java

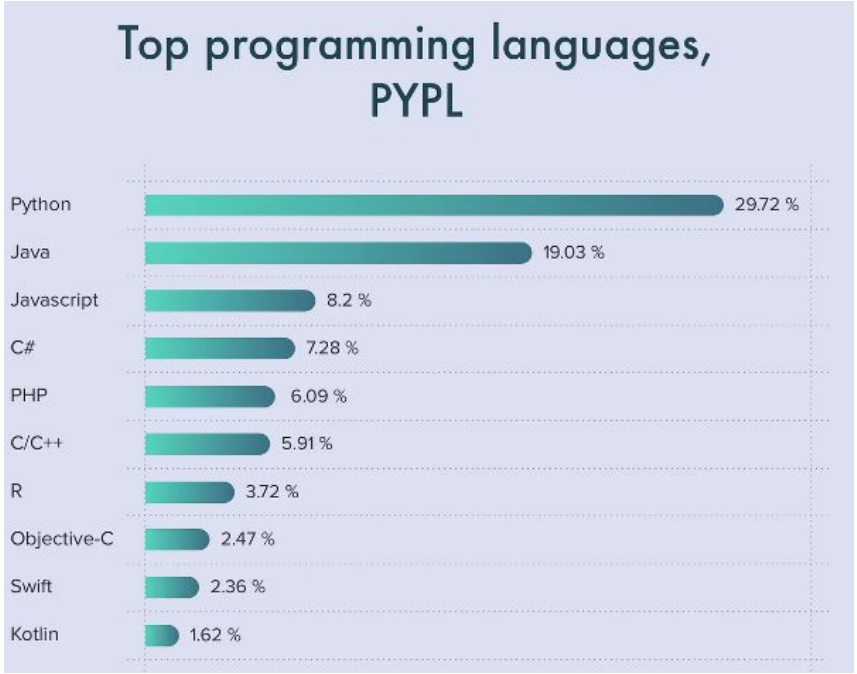
Programación Orientada a Objetos

San Salvador de Jujuy

UNJu - Facultad de Ingeniería
Ing. José Zapana

¿Qué es Java?

- Java es un lenguaje de programación de propósito general, orientado a objetos y de alto nivel, desarrollado por Sun Microsystems en 1995.
- Es ampliamente utilizado para el desarrollo de aplicaciones empresariales, móviles, web y sistemas embebidos.



Introducción a Java

- Java, conocido por su portabilidad y escalabilidad, sigue siendo una opción popular en el desarrollo empresarial. Su robusta infraestructura y la compatibilidad multiplataforma le aseguran vigencia a lo largo del tiempo.

- Aunque su sintaxis puede parecer un tanto extensa, su comunidad activa y la confiabilidad en grandes proyectos empresariales son sus puntos fuertes.

Ventajas y Desventajas

Ventajas

- 1. Portabilidad: 'Write Once, Run Anywhere' (WORA).
- 2. Seguridad: Fuerte gestión de memoria y verificación de bytecode.
- 3. Orientación a Objetos: Facilita la reutilización de código.
- 4. Amplia comunidad y soporte: Gran cantidad de bibliotecas y frameworks.
- 5. Multithreading: Manejo de múltiples tareas simultáneamente.

Desventajas

- 1. Rendimiento: Más lento que lenguajes compilados como C++.
- 2. Consumo de memoria: Requiere más memoria que otros lenguajes.
- 3. Verbosidad: Código más largo en comparación con otros lenguajes modernos.
- 4. Gestión de recursos: Requiere atención para evitar fugas de memoria.

Mejoras en las últimas versiones

- **Java 8:** Introducción de lambdas, Streams API, y la API de fechas.
- **Java 9:** Sistema de módulos (Project Jigsaw) y JShell.
- **Java 10:** Inferencia de tipos con 'var'.
- **Java 11:** Nuevas mejoras en la API HTTP y soporte para ZGC (Garbage Collector).
- **Java 17:** Características de previsualización como patrones de coincidencia y sellado de clases.
- **Java 21 y 22:** Mejoras varias en lenguaje, sus API y rendimiento.

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;
```

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 keyboard 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 {
  private String nombre;
   private int balance;
   private int Id;
   private static int proxIdDisponible = 1;

   /** Constructor, establece nombre dueño y balance
   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 accesar 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 LocalDate getNow(){
          return LocalDate.now();
       }
}
```

…

Date fechaIngreso = FechaUtil.getNow();