

Prog. Avanzada/PB2

Días de Cursada

Martes 19 a 23.

Plataformas

- MleL -> Mensajes y Parciales
- MS Teams -> Clases Virtuales

Condiciones para aprobar la materia

Dos Parciales Teóricos Prácticos

Promoción más de 7 en cada parcial y Correlativas Aprobadas

Nota de 4 a 6 Regular.


Nota 4 = 60% del parcial correcto.

Recuperatorio 1 al final de cuatrimestre.

Sólo se puede recuperar un parcial.

Libre, siguiendo las condiciones de la Universidad.

Cronograma

- 01 - Presentación / Ejercicio práctico - Ejercicio práctico Arrays & Matrices (Sala de Cine)
 - 02 - Junit Básico / TDD
 - 04 - Herencia / Clases Abstractas - Super / Sobreescritura y Ocultamiento (cómo y no el qué)
 - 05 - Práctica
 - 06 - Composición / Colecciones Básico (ArrayList y LinkedList)
 - 07 - Práctica
 - 08 - Parcial 1
- 

Cronograma

09 - Polimorfismo / Interface

10 - Práctica

11 - Colecciones Avanzadas (Interfaces e Implementaciones)

12 - Excepciones / Creación

13 - Práctica con: Patrones de diseño (Template Method)

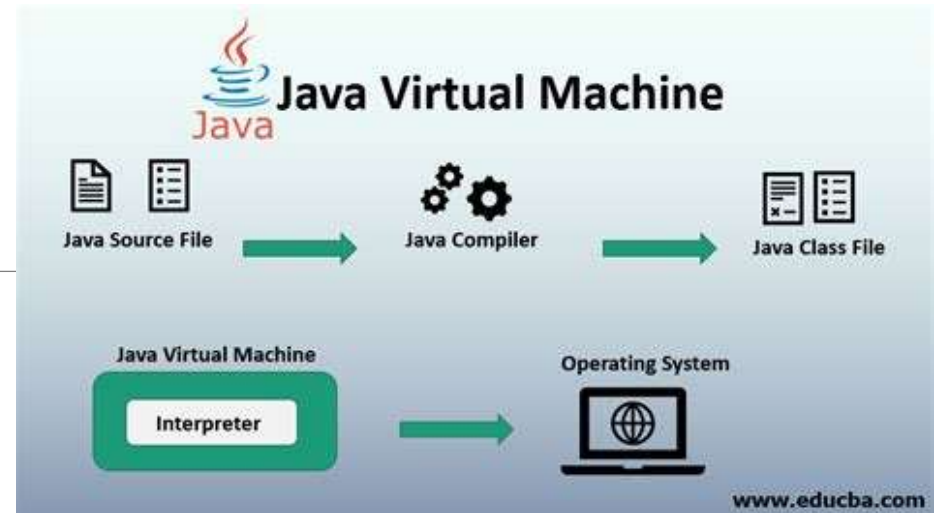
14 - SOLID - presentación (Encapsulamiento - el objeto decide cómo se modifica a si mismo)

15 - Parcial 2


16 - Recuperatorio (Parcial 1 ó 2) - Cierre de notas

Java: composición JDK

- Compilador
- Java Machine Virtual (JMV – Máquina Virtual de Java)
- Interfaz de Programación de Aplicaciones (API: Application Programming Interface)



Java: características

- Uso libre
 - OO
 - Portabilidad
 - Aplicaciones robustas
 - Interpretado y compilado
 - Multithread
- 

Java: ámbitos

- Dispositivos móviles
- Web
- Sistemas de servidores
- Aplicaciones de escritorio

IDEs

- Integrated Development Environment (Entorno de Desarrollo Integrado)
 - Errores de escritura
 - Palabras reservadas
 - Ayudas
 - Depurador o Debugger
- NetBeans
- Eclipse

Hola Mundo!

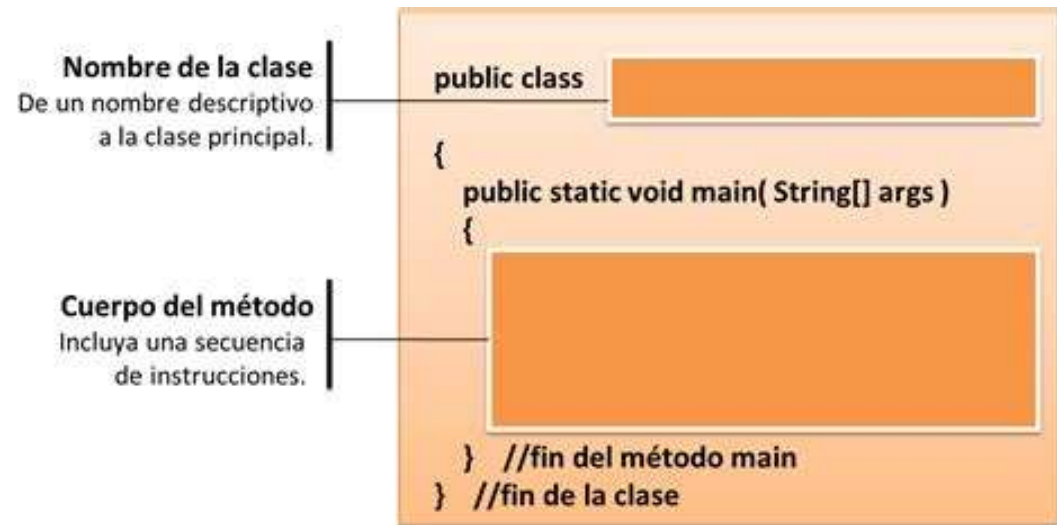
```
public class Saludo {  
  
    public static void main(String[ ] args) {  
        System.out.println("Hola mundo!");  
    }  
}
```

Nombres

- Pueden contener caracteres alfabéticos, dígitos y/o algunos caracteres especiales, como guión bajo, \$, etc, pero no pueden comenzar con un dígito.
 - x1 es válido
 - 1x es inválido
- Nombres compuestos
 - camelCase: sueldoNeto (estilo Java)
 - snake_case: sueldo_netto (estilo Python)
- No pueden usarse palabras reservadas, como for, else, etc.

Clases

- Clases
 - Visibilidad
 - Atributos
 - Constructores
 - Métodos y parámetros
- Objetos



Clases



Objetos



Clases

Una clase en un archivo

Un archivo por clase

```
public class Complejo
{
    double real;
    double imag;
}
```

Clases Atributos

Una clase en un archivo

Un archivo por clase

```
public class Complejo
{
    private Double real = 8.8;
    Double imag;
}
```


Classes Constructor

```
public class Employee {  
    private String firstName;  
    private String lastName;  
  
    public Employee() { //constructor 1  
    }  
  
    public Employee(String firstName) { //constructor 2  
    }  
  
    public Employee(String firstName, String lastName) { //constructor 3  
    }  
}
```

public class Employee {

**public Employee() { //Default constructor
//statements
}**

**public Employee(String name) { //Parameterized constructor
//statements
}**

Employee e1 = new Employee();

Employee e2 = new Employee("Lokesh");

Clases Métodos

<i>visibilidad</i> (opcional)	<i>tipo_de_retorno</i> (obligatorio)	<i>nombre</i> (obligatorio)	<i>parámetros</i> (puede ser vacío)
----------------------------------	---	--------------------------------	--

abstract. Estos métodos no contienen cuerpo, el cual deberá ser definido por una subclase (algún heredero).

- Pasaje por valor
- Pasaje por referencia

Tipos primitivos

Tipo primitivo	Tamaño	Mínimo	Máximo	Tipo envoltura
boolean	-	-	-	Boolean
char	16 bits	Unicode 0	Unicode $+2^{16} - 1$	Character
byte	8 bits	-128	+127	Byte
short	16 bits	-2^{15}	$+2^{15} - 1$	Short
int	32 bits	-2^{31}	$+2^{31} - 1$	Integer
long	64 bits	-2^{63}	$+2^{63} - 1$	Long
float	32 bits	-3.4×10^{38}	$+3.4 \times 10^{38}$	Float
double	64 bits	-3.4×10^{308}	$+3.4 \times 10^{308}$	Double
void	-	-	-	Void