

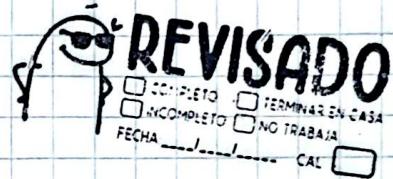
Nombre del algoritmo "Hola mundo"

1 { int

2 Entrada; "Hola mundo";

3 Fin del algoritmo

4 }



Programación orientada a objetos.

Un modelo o un estilo de programación un poco similar de hacerse basa en tratar a el y que esté basado en el concepto de clases y objetos. Es tipo de tema de base para estructurar un programa de software.

$$\begin{array}{r} \text{3 = } \\ \times \quad \times \quad \times \\ - \quad 0 \quad 1 \\ \hline 4 \quad 2 \quad 1 \\ \hline 1 \quad 0 \quad 1 \end{array}$$

La POO se usa el código reutilizable organizado para manejar por el desarrollo de software ampliado por varios programadores. Sirve para facilitar el acceso a datos personales, a la ejecución de código propietario.

Ingeniero

Nombre: Alberto
Título: Jefe
Empresa: Fort
Edad: 28
Número: 1234

Loco

Nombre: GTR 3
Número: 1234
Cilindrada: 320 k
Velocidad: 150 km/h
Peso: 100 t
Número de serie: 123456789



Teléfono	base
Modo 01347	olypto
Pulgadas 8.74	
Peso 32.4g	
Software Si	
Nombre Motobala	
Barco	close
Tiene Modo Si	olypto
Tiene llantas No	
Flota Si	
Balente. Si	

Clase

Una clase es una plantilla que define de modo genérico ~~sobre~~ los objetos de un determinado tipo

Objeto

Un objeto es la utilización de esa misma plantilla para obtener características de ese mismo clase

Instancia

Se aplicará cada que se crea plantillas a la de necesidad llamadas

D

M

A

18/02/2024

Scribe®

Capsulamiento: La encapsulación se presenta todo el información importante de un objeto y solo expone la información elegida al mundo exterior. Esta propiedad permite asegurar que la información de un objeto esté oculta, agrupada en una clase con atributos o métodos que tiene un acceso privado, y los comportamientos o métodos que cuentan con un acceso público. La encapsulación de cada objeto es responsable tanto de su información como de su estado. El único modo en el que se puede modificar es a través de sus propios métodos. De esta manera, los atributos internos de un objeto son inaccesibles desde fuera, pudiendo solo modificarlos llamando a los funciones correspondientes. Así se consigue mantener el todo dentro de usos intencionalmente predefinidos y evitar resultados inesperados.

- Público
- Protegido
- Privado

Abstracción: Es un tipo de programación orientada a objetos que se practica cuando el usuario interactúa solo con atributos y métodos, usando herramientas simplificadoras de alto nivel para acceder a un objeto complejo en la PDS. Los programas suelen ser muy grandes y los objetos suelen comunicarse mucho entre sí. De este modo la abstracción facilita el mantenimiento de un código bastante tarde, donde pueden surgir drásticos cambios con el paso del tiempo. Así la abstracción es la base para representar cosas simples que tengan complejidad. Los objetos y las clases representan código subyacente quitando la complejidad a los usuarios.

Tarea: Investigar términos herencia y polimorfismo

JDK = "Complejo"

Java = "Lenguaje"

Metabeanos

D

M

A

18/02/2024

Scribe®

Como instalar java paso a paso

1: Buscar Java Jdk

2: Entrar al link "Java Downloads"

3: elegir donde instalarlo "linux, MacOs, Windows"

4: Descargar documento y dale click derecho

5: Te abre una ventana y debes de aceptar los términos donde "Next"

6: Elegir la carpeta donde instalar

7: Verificar la descarga

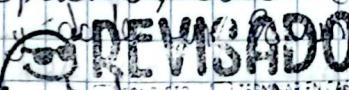
8: Abrir una simulación y abrir Java

9: Quitar las variables de Java como %JAVA_HOME%

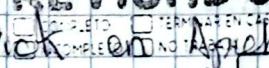
Como instalar Netbeans

1: Abrir Navegador y buscar Netbeans

2: Darle click a Apache Netbeans Releases

3: Abrir una ventana con varias opciones, darle click a donde dice Apache NetBeans 12.6 

4: Te abren varios sitios, darle click en Apache NetBeans 12.6 Win windows



5: Te abre otro link y da click al primero

6: Se descarga, da click derecho a la descarga

7: Te abre una pantalla y da click en Exit instalar

8: Te abre una pantalla para guardarlo en mi carpeta y ya lo puedes abrir

D M A
18/02/2024

Scribe®

Herencia: La herencia que una clase (llamada clase hija o clase hijo) adquiere propiedades y métodos de otra clase (llamada super clase y clase padre).

La clase hijo puede usar métodos y atributos de la clase padre sin necesidad de escribirlos de nuevo.

La clase hija puede agregar o modificar comportamientos sin afectar la clase padre.

"Tipos de Herencia"

Herencia Simple

Herencia Multinivel

Herencia Multiple

Herencia jerárquica

Polimorfismo: El polimorfismo permite que diferentes clases usen el mismo método de distintas formas. Es decir, un mismo método pueden comportarse diferentes según la clase lo implementa.

Se diferencia más de un método con el mismo nombre pero con diferente parámetro.

"Tipos de polimorfismo"

- Polimorfismo de compilación
- Polimorfismo de ejecución

Para ver si Java tiene la versión deseada de Poxer (Windows + R) aparece un buscador y debemos poner (cmd), Aperte una ventana negra y debemos de poner (java -version) y aparece la versión de java

* se escriben los nombre de la clase con mayuscula **public class**

Práctica &

se inici

{ Se abre Java, se debe de cerrar

2: Cualesquier método público puede entrar cualquier persona, proteger solo lo puedes ver quien quieras.

* Siempre se pone público cuando es main

* El método main siempre tiene static

* El método main inicia la clase

* String es una clase [] donde se guarda o donde se arregla

* Asignar tipo de cadena de caracteres.

* Tipos de datos en Java

Tipos Primitivos

- byte = Números enteros pequeños
- short = Números Enteros medianos
- int = Números Enteros Grandes -128 a 128
- long = Números Enteros muy grandes (requiere LEI al final)
- float = Números decimales de precisión simple (requiere LEI al final)
- double = Número decimal de precisión doble (más preciso que float)
- char = Almacena un solo carácter (ej. 'A')
- boolean = Representa True o False

Clases Wrapper (Envoltorios)

- Byte = Versión objeto de byte
- Short = Versión objeto de short
- Integer = Versión objeto de int
- Long = Versión objeto de long
- Float = Versión Objeto de float
- Double = Versión objeto de double
- Character = Versión objeto de char
- Boolean = Versión objeto de boolean

Tipos no Primitivos

- String = Almacena texto inmutable
- StringBuilder = Cadenes modificables, más rápidos que String
- String Buffer = Similar a String Builder, pero sincronizados para hilos
- Array = Arreglo de tamaño fijo [int[]] numeros = {1, 2, 3};
- List (ArrayList, LinkedList) = Lista dinámica
- Set (HashSet, TreeSet) = Conjunto sin elementos duplicados
- Map (HashMap, TreeMap) = Diccionario clave-valor
- Queue (LinkedList, PriorityQueue) = Cola FIFO
- Stack = Pila LIFO

Tipos Numéricos avanzados

→ Números enteros extensamente grandes

- BigInteger → Almacena long → Similar a BigInteger pero con long
- BigDecimal → Alta precisión con decimales (mucha más que double)
- AtomicInteger → Entero seguro en programas con múltiples hilos

D

M

A

23/02/2023

Scribe®

Tipos de Fecha y Tiempo (java.time)

- LocalDate = Fecha sin hora (2023-02-22)
- LocalTime = Hora sin fecha (14:30:00)
- LocalDateTime = Fecha y hora combinados (2023-02-22T14:30)
- ZonedDateTime = Fecha y hora con zona horaria
- Instant = Momento en tiempo en nanosegundos
- Period = Diferencia entre dos fechas (años, meses, días)

Tipos Especiales

- Optional<T> = Evita NullPointerException con valores opcionales
- AtomicInteger = Pattern y AtomicInteger = Expresión regular para buscar fechas
- AtomicLong
- UUID = Identificadores únicos globales

Tipos Personalizados

- enum = Lísta de valores constantes (enum class {Lunes, Martes, ...})
- record = Clase inmutable con valores (record Punto (int x, int y))
- Clases definida por el usuario (class) = Define un objeto con atributos y métodos

Cadenas Y Texto

String
StringBuffer
StringBuilder

Tipos de Entrada/Salida

- File = Representa archivos y directorios
- Path = Manipulación moderna de rutas de archivo
- Files = Método (Path) para trabajar con archivos
- InputStream (FileInputStream) = Flujo de entrada
- OutputStream (FileOutputStream) = Flujo de salida
- Reader = Lectura de caracteres
- Writer = Escritura de caracteres

Tipos de Base de Datos

- Connection = conexión a una base de datos
- Statement = Ejecuta una o más SQL Completas
- PreparedStatement = Consultas SQL preparados y parametrizados
- ResultSet = Resultado de una consulta SQL
- Blob, Clob, Array, Struct, SQLXML = Almacena grandes datos binarios o de texto, Manipula estructura de datos, Manipulación de datos XML en SQL.

Aritméticos

% módulo resta lo que sobra

long	int
char	string

+ suma

- resta

/ división

* Multiplicación

++ suma en uno en uno

-- resta uno en uno

Asignación

=

Comparación

+=

= = true - false

-=

> = true - false

/=

< = true - false

*=

< = true - false

%=

> = true - false

!= Diferencia true - false

Logicos

|| Y

! O

Negación o not

public class Operaciones {

public static void main (String [] args)

Introducción Variables

Tipo Nombre;

```
public class Suma {  
    public static void main (String [] args)  
    { int v1=5, v2=2;  
        int r;  
        r = v1+v2;  
        } System.out.println ("El resultado es": r);  
    }
```

```
public class Resta {  
    public static void main (String [] args)  
    { int v1=5, v2=2;  
        int r;  
        r = v1 - v2;  
        } System.out.println ("El resultado es": -r);  
    }
```

```
public class Multiplicación {  
    public static void main (String [] args)  
    { int v1=5, v2=2  
        int r;  
        r = v1 * v2;  
        System.out.println ("El resultado es": *);  
    }
```

```
public class Division {  
    public static void main (String [] args) {  
        int v1=5, v2=2;  
        int r;  
        r = v1 / v2;  
        System.out.println ("El resultado es: " + r);  
    }  
}
```

```
public class Modulo {  
    public static void main (String [] args) {  
        int v1=5, v2=2;  
        int r;  
        r = v1 % v2;  
        System.out.println ("El resultado es: " + r);  
    }  
}
```

Una persona quiere calcular el área de un terreno el terreno tiene una forma rectangular que tiene de largo 25.5 metros, de ancho 8.4 metros además la persona quiere calcular los metros que cubre el contorno del terreno, además el terreno lo va a dividir entre 8 hijos que tiene, calcular el área de cada terreno que le dará al hijo y su perimetro.

```
public class Terreno {
    public static void main (String [] args)
```

$$\begin{aligned} \text{float } L = 25.5, A = 8.4; \\ \text{float } r; \end{aligned}$$

$$\begin{aligned} r = L * A; \\ r = 214.2; \end{aligned}$$

System.out.println ("El resultado es " + r);

$$\begin{aligned} \text{float } L = 25.5, A = 8.4; \\ \text{float } r; \end{aligned}$$

$$\begin{aligned} r = L + A + L + A; \\ r = 67.8; \end{aligned}$$

System.out.println ("El resultado es " + r);

$$\begin{aligned} \text{float } A = 254.2, H = 8; \\ \text{float } r; \end{aligned}$$

$$\begin{aligned} r = A / H; \\ r = 26.775; \end{aligned}$$

System.out.println ("El resultado es " + r);

Condición en Java

IF (condición) {
 instrucción
}

$x > 50$

 x = PAR

else {

 instrucción

}

IF ($y > 5$) {

Hacer un programa en Java que me solicite el mes de año y
de acuerdo al numero del mes

me me capture el mes que quiero encontrar. 5 (Mayo)

Import. java.util.Scanner;
Public class Mes;

Int Mes, Num;

Scanner ssc = new Scanner();

System.out.println("Ingresa un numero del 1-12
que corresponda a un mes");

Num = ssc.nextInt();

If (Mes == 1)

System.out.println("Enero");

else if (Mes == 2)

System.out.println("Febrero");

else if (Mes == 3)

System.out.println("Marzo");

else if (Mes == 4)

System.out.println("Abril");

else if (Mes == 5)

Algoritmo Juego de la Piedra, Papel o Tijera

Definir computadora (como Entero);

Definir jugador (como Entero);

Computadora ← Aleatorio (1, 3);

Escribir "Selección, una opción";

Lectura "1 = Piedra";

Lectura "2 = Papel";

Lectura "3 = Tijera";

Lectura jugador;

Si (jugador = computadora) entonces

Escribir "Empate";

Fin Si

Si (jugador = 1 Y computadora = 2) entonces

Escribir "Pierdes";

Fin Si

Si (jugador = 2 Y computadora = 3) entonces

Escribir "Ganas";

Fin Si

Si (jugador = 2 Y computadora = 1) entonces

Escribir "Ganas";

Fin Si

Si (jugador = 2 Y computadora = 3) entonces

Escribir "Pierdes";

Fin Si

Si (jugador = 3 Y computadora = 1) entonces

Escribir "Pierdes";

Fin Si

Si (`jugador = 3`) y (`competicion = 2`) entonces

"Ganó";

Fin 2.

Fin algoritmo

```
Package com.example.practicas.practicas4;
import java.util.Scanner;
public class Piedra_Papel_Tijera {
```

```
Public static void main (String [] args) {
```

```
System.out.println ("Juego P-P-T");
```

```
Scanner teclado = new Scanner (System.in);
```

```
int comp = (int) (Math.random () * 3) + 1;
```

```
int jug = teclado.nextInt ();
```

```
if (comp == jug) {
```

```
System.out.println ("Empate");
```

```
} else if (comp == 3 && jug == 2) {
```

```
System.out.println ("Pierdes");
```

```
} else if (comp == 3 && jug == 1) {
```

```
System.out.println ("Ganás");
```

```
} else if (comp == 2 && jug == 1) {
```

```
System.out.println ("Pierdes");
```

```
} else if (comp == 2 && jug == 3) {
```

```
System.out.println ("Ganás");
```

```
} else if (comp == 1 && jug == 2) {
```

```
System.out.println ("Ganás");
```

```
} else if (comp == 1 && jug == 3) {
```

```
System.out.println ("Pierdes");
```

Ciclo for

Sintaxis

```
for (variable; condición; incremento s) {  
    decremento  
}
```

```
public class Tabla-Pr-Dos {
```

```
    public static void main (String [] args) {
```

```
        int base=2;
```

```
        for (int i=1; i<=10; i++) {
```

```
            System.out.println ("base+"x"+i+" = "+base*i);
```

```
} }
```

```
public class Tabla-f.
```

```
    public static void main (String [] args) {
```

```
        int = base;
```

```
        System.out.println ("Ingresa el numero de la  
        tabla que deseas imprimir");
```

```
        Scanner tecleado = new Scanner (System.in);
```

```
        int base = tecleado.nextInt();
```

```
        for (int i=1; i<=10; i++) {
```

```
            System.out.println ("base+"x"+i+" = "+base*i);
```

Muy buena explicación, es de gran ayuda para entender mejor el ciclo for.

Que es IDE y que significan los siglos

IDE significa "Entorno de desarrollo integrado" (Integrated Development Environment en inglés.)

Es un software que proporcionan herramientas para programar de manera eficiente.

- Editor de código (para escribir el código fuente)
- Depurador (para encontrar y corregir errores.)
- Compilador o intérprete (para convertir el código en programa ejecutable)
- Auto completado y recordado de sintaxis (para facilitar la escritura del código)

Ejemplos:

Visual Studio, Eclipse, PyCharm, etc.

Diferencia entre Primitivos y compuestos

Datos primitivos Es el tipo de dato más básico, no se pueden descomponer en elementos más pequeños.

Ejemplos:

- Entero (int) → 5, 10, -3
- Decimal (float) → 3.14, 7.5
- Caracteres (char) → 'A', 'z'
- Booleanos (bool) → True, False.

Datos compuestos Se forman a partir de varios datos primitivos combinados.

Ejemplos:

- Cadena de texto (String) → "Hola Mundo" (compuesto de caracteres)
- Lista o arreglos (array). [1, 2, 3, 4] → [1, 2, 3, 4]
- Estructura y objetos (Struct, class) → Agrupan múltiples datos en una entidad más compleja

Los datos primitivos son simples y los compuestos agrupan varios valores.

¿Qué es un acumulador?

Es una variable que se usa para sumar o acumular valores dentro de un proceso, generalmente en un ciclo o bucle. Su función principal es ir almacenando el resultado de una operación repetitiva.

Ejemplo:

Acumulador ← 0 // se inicia en 0

para i ← 1 hasta 5 hacer

acumulador ← Acumulador + i // se suma el valor en i

Fin para

mostrar acumulador // Resultado: 15 ($1+2+3+4+5$)

Características:

✓ Se inicia con un valor (usualmente 0) (se aplica a todos)

✓ Se actualiza en cada iteración del ciclo

✓ Se usa para sumar, restar o acumular resultados

¿Qué es un contador?

Es una variable que se usa para llevar la cuenta de cuantas veces ocurre algo dentro de un programa, generalmente dentro de un bucle. Se implementa en bucle fijo (usualmente for) en cada iteración.

Ejemplo:

Contador ← 0 // se inicia en 0

para i ← 1 hasta 5 hacer

contador ← contador + 1 // se incrementa

Fin para

Mostrar contador // Resultado: 5 (Porque el bucle se repitió 5 veces)

Usos comunes:

✓ Contar ciertas veces se repite un evento

✓ Contar elecciones en una linea

✓ Llevar un registro de interacciones en un bucle

Q es un bucle y cuales son los mas usados en java?

Es una estructura en programación que permite repetir un bloque de código varias veces hasta que cumple una condición.

Ejemplos:

Bucle "Para" (For) → Usarla cuando se sabe cuánto se repetirá

Para i = 1; i <= 5;
 mostrar i
Fin para

Salida: 1,2,3,4,5.

Bucle "Mientras" (while) → Mientras una condición sea verdadera

i <= 5
Mientras i <= 5 hacer
 mostrar i
 i = i + 1
Fin Mientras

Bucle "Pensar Hasta" (do-while) → Se ejecuta 1 vez, luego depende de la condición

repetir
 Mostrar i
 i = i + 1
Hasta i > 5

Usos

- ✓ Recorrer un lista
- ✓ Contar elementos
- ✓ Hacer cálculos repetitivos.

Tarea: Fracciones de la clase
Math como se implementa en Java

Scribble

- Clasificación

Math. pow (a, 2);
ventana Math. sqrt (a); - Raíz
Panel

import javax.swing.JOptionPane;
String texto = JOptionPane.showInputDialog("Ingrésa valor");
String texto = JOptionPane.showInputDialog("Número");
String double = JOptionPane.showInputDialog("Número entero")

Input javax.swing.JOptionPane; int valor = Integer.parseInt("");
public class DibujarNumeros {
 public static void main (String [] args) {
 String resultado = "Número ingresado por el usuario convertido en entero";
 }
}

Funciones Básicas.

Funciones de Math

- Math.abs(x); Devuelve el valor absoluto de x.
- Math.max(a, b); Devuelve el valor máximo entre a y b.
- Math.min(a, b); Devuelve el valor mínimo entre a y b.

Potencias y Raíces.

- Math.pow (base, exponente); Calcula $\text{Base}^{\text{exponente}}$
- Math.sqrt(x); Raíz cuadrada de x.
- Math.cbrt (x); Raíz cúbica de x.

Logaritmos y Exponentiales.

- Math.exp (x); Calcula e^x (donde e es el número de Euler)
- Math.log (x); Logaritmo natural (base e)

Punto de venta - Comisión
Publico total vtd mun (3) vnd (3) args) {

$$\text{Total} \cdot C = 100, Vg = 1500, \text{IVA} = 10.000 \text{ } \{ r_1 = 0.25, r_2, r_3 \\ T = 0.14, v_1 = 0.25, v_2 = 0.25, r_4, r_5, r_6, r_7, r_8 \}$$

$$r_2 = Vg * IVA; \quad r_8 = r_2 + r_2;$$

$$r_2 = Vg * \text{IVA} / 100;$$

$$r_3 = r_2 * C_1 / 100;$$

$$r_4 = r_3 + r_2;$$

$$r_5 = IVA * B_1;$$

$$r_6 = IVA * T;$$

$$r_7 = IVA * T;$$

System.out.println ("El IVA es " + r2);

System.out.println ("La ganancia es " + r3);

System.out.println ("El precio de los artículos es " + r5);

System.out.println ("El precio de la tele es " + r7);

System.out.println ("El precio de la silla es " + r6);

System.out.println ("El precio de los 1500 comisiones es " + r1);

System.out.println ("El precio total con el IVA es " + r8);

3

3



Caja

- Recuerda
actuador



Pantalla

Huecogranadas



Triángulo

- Recuerda
almacenamiento

Indicar
el filo

- Separar colisiones reconocidas
- Pasos lógicos
- Clasidad sobre belleza
- Elección de lo que tiene los módulos.

Introducción
• Objetos
• Fase
• Persona

Atributos
• Círculos, rectángulos
• Figuras

Uso
• Tipo de acción
• corrección

IGU {
↓ Uso de el
 ↑ mundo exterior mundo
 ↓ recuerda todo, much
 ↑ cosas, todo
Personas { recordar todo

~~Uso y uso solo~~

La leyida se muga
en el paquete
de servicios

JDBC Por sus siglas en inglés Java Data Base Connection

Es la librería que me permite conectarme a una base de datos mediante un Driver de conexión.

Connection

Clase que me permite operar con métodos de conexión a base de datos.

Driver Manager

Clase que me permite configurar la conexión a clase de datos.

SQLException

Clase que me permite identificar errores en ejecución de base de datos "SQL"

Statement

Me permite crear un objeto que me permite la realización de consulta, comando, actualización, y creación de datos dentro de una base de datos.

ResultSet

Contiene el resultado de la ejecución del comando ejecutado dentro de la clase anterior simplemente.

Action Listener

Permite utilizar eventos relacionados con un dispositivo o control de la pantalla.

JFrame

Es tipo clase que me permite construir el frame que me permite escribir texto en Java

Tenemos 6 ecuaciones que no contamos con la de solo 1 variable cada una solo contiene una variable, dame el valor de esa variable.

Public class Resultado extends JFrame {

```
public JPanel panel = new JPanel ();
public JLabel variable = new JLabel ("");

```

```
Panel V, r;
```

variable.setEditable (false);

variable.addActionListener (new ActionListener () {

public void actionPerformed (ActionEvent e) {

```
if (e.getSource () == V) {
```

```
r = (2 * V) + 5;
```

System.out.println ("El resultado es " + r);

```
} else if (e.getSource () == VI) {
```

```
r = (3 * VI) + 5;
```

System.out.println ("El resultado es " + r);

D

M

A

Scribe®

Public class Operation extends JFrame {

 public JPanel panel = new JPanel();

 public JCheckBox ch1 = new JCheckBox();

 public JCheckBox ch2 = new JCheckBox();

 public JButton Resolver = new JButton("Enter");

 public Operation

 this.setSize(500, 300);

 this.setLayout(new BorderLayout());

 this.setResizable(false);

 this.getLayout().add(panel, "Center");

 panel.setLayout(null);

 panel.add(ch1);

 panel.add(ch2);

 panel.add(Resolver);

 panel.add(Resolver);

}

};

 this.getContentPane().add(panel);

3)

D M A
8/04/23

Scribe®

MenuItem

- Opción dentro de un menú
- text, accelerator, actionListener.

JDialog

- Ventana secundaria usada para diálogos
- modal, title, visible, size

JPasswordField

- Campo de texto que oculta lo escrito (como contraseña)
- echoChar, text, columns

JOptionPane

- Muestra cuadros de diálogo estandar (mensaje, confirmación, input)
- Método estático como showMessageDialog, showConfirmDialog

JFileChooser

- Dialogo para seleccionar archivos o directorios
- currentDirectory, selectFile, fileFilter

CardLayout

LAYOUT PARA MOSTRAR UN PUEBLA LA VEZ (COMO CARTAS VISIBLES)

Métodos show, next, previous.

JTable

Muestra datos en forma de tabla

model, rowHeight, columnModel, selectionModel.

JPanel

- Contenedor genérico para organizar otros componentes
- layout, background, border, size.

JFrame

- Ventana principal de una aplicación Swing
- title, size, defaultCloseOperation, visible.

JButton

- Botón que ejecuta una acción al hacer click
- text, icon, enabled, ActionListener.

JComboBox

- lista desplegable para seleccionar un ítem
- selectedItem, model, editable, ActionListener

JLabel

Mostrar texto o imagen sin interacción

text, icon, foreground, Font.

JTextField

Campo para ingresar datos

font, columns, editable, ActionListener.

JList

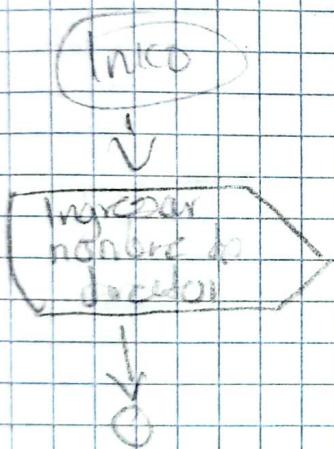
Lista de elementos visibles con posibilidad de selección

selectedIndex, model, selectionModel.

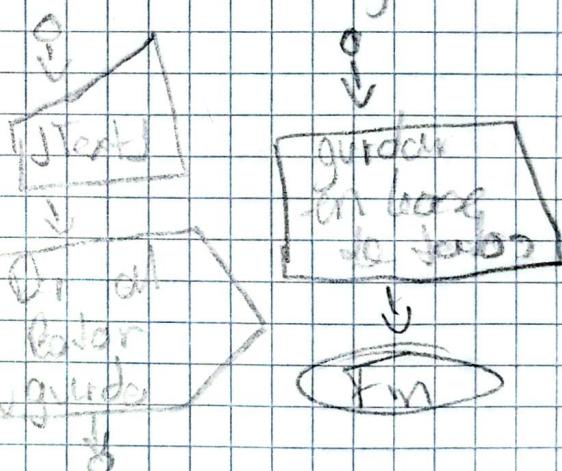
Alumnos: 10) Víctor
11) Giovanni

Registro Director:

Nombre:	<input type="text"/>
<u>Giovanni</u>	



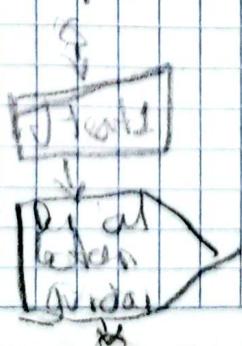
- 1) Ingrasa a la pantalla de registro
- 2) Ingrasa el nombre del director
- 3) Dar al botón guardar



- 1) Ingrasa a la pantalla de registro
- 2) Ingrasa el nombre del género
- 3) Dar al botón guardar

Registro Género:

Nombre:	<input type="text"/>
<u>Guardado</u>	



Director:

10) Víctor
11) Giovanni
Director y
genio
y su
director
en la
base
de datos.

Forma de mostrar un mensaje.

La primera consonante, btn
spn, chk, rbd

Scribe®

Consola

```
Public class Mensaje {
```

```
    public static void main (String [] args) {  
        System.out.println ("holo");  
    }  
}
```

JDialog

JOptionPane

```
Import javax.swing.JButton;  
Import javax.swing.JDialog;  
Import javax.swing.JFrame;  
Import javax.swing.JLabel;
```

```
Public class Mensaje extends JFrame {
```

```
    private JLabel , bl  
    private  
    private
```

Un constructor es el llamado a una clase siempre se llama de la misma forma de la clase, no tiene un valor de retorno. Public mensaje()

D

M

A

Scribe®

Recibirar una clase que me permita manejar 4 tipos de mensajes

1 = Artista Favento

2 = Deportes fav

3 = Nombre completo

4 = Nombre Precio

Public class 4Messages {

private String ArtistaFav;

private String DeportesFav;

private String NombreCompleto;

private String NombrePrecio;

Public 4Messages (String ArtistaFav, String DeportesFav,

String NombreCompleto, String NombrePrecio) {

this. ArtistaFav = Artista Favento;

this. DeportesFav = Deportes fav;

this. NombreCompleto = Nombre Completo;

this. NombrePrecio = Nombre precio;

Public void MostrarMensaje () {

JOptionPane.showMessageDialog (null, "Artista Favento es " +
 JOptionPane.showMessageDialog (null, "Deporte fav es " +
 JOptionPane.showMessageDialog (null, "Nombre completo es " +
 JOptionPane.showMessageDialog (null, "El nombre de precio es " +

ArtistaFav);

DeportesFav);

NombreCompleto);

NombrePrecio);

*.jar

java api resources

* war

ejecutable

windows

* bat

Mac / linux

* sh
* sc

Para poder ejecutar el archivo y luego el 100% de debe instalar

1 - java

2 - My sql

3 - Ejecutar el script BB

Se utiliza para compartir un proyecto y poder usarlo en otro servidor como si instaláramos una aplicación.