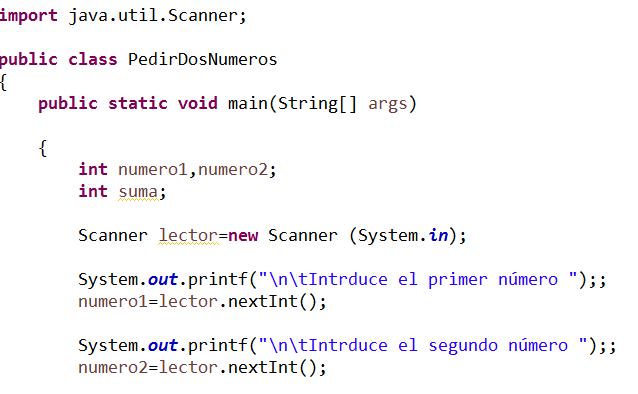
|  |
| --- |
| Solicitar datos por teclado usando la clase Scanner |



1. Hay que importar la librería java.util.Scanner
2. Crear un objeto de tipo Scanner que es el que recoge el dato que se mete por teclado pasándole System.in para indicar que va a recoger datos de entrada
3. Tener en cuenta el tipo de dato que se quiere recuperar si es entero

lector.nextInt() si es Double lector.Double() etc para los demás tipos

Dependiendo del tipo de dato a imprimir usando printf se usará una u otra máscara

%b dato booleano

%d dato numérico

%s dato de cadena

. . . .. . . . . .

System.***out***.printf("Valor de casado %b\n",casado);

System.***out***.printf("Valor de MAXIMO %d\n",MAXIMO);

System.***out***.printf("Valor de nombre %s\n",nombre);

|  |
| --- |
| Dar formato a un dato |

# **Ej B2\_1**

# 

Pedir al usuario que introduzca un número entero.

El sistema calcula la raíz cuadrada de este número y saca en pantalla en una sola línea el resultado redondeándolo con 4 dígitos y sin hacer uso de salto de línea.

Nos dicen que lo saquemos con 4 decimales

Usamos la clase la función **sqrt** de la clase Math está incluida ya en Java por tanto no hace falta importar la librería.

resultado=Math.*sqrt*(valorintroducido);

DecimalFormat df = **new** DecimalFormat("##.0000");

System.***out***.println("\n\t\t\tLA RAIZ CUADRADA DE " + valorintroducido + " ES " +

**df.format(resultado)) ;**

Tarea B3

# **Ej B3\_1**

# 

Tienes que :

1. declarar una variable cad1 con tipo de dato String
2. y atribuirle el valor "EJEMPLO CADENA"
3. declarar una variable cad2  con tipo de dato String
4. y atribuirle el valor "ejemplo cadena" en minúsculas
5. imprimir en pantalla las dos variables usando el método println()
6. usando el método printf():
   * imprimir en pantalla las dos variables.
   * sacar en pantalla la longitud de cad1.
   * concatenar cad1 con cad2.
   * comparar las cadenas usando cada uno de los 2 métodos disponibles.
   * comparar las cadenas usando el método compareTo().
   * sacar en pantalla los 5 primeros caracteres de cad1.
   * pasar cad1 a minúsculas.
7. encontrar la ficha de compareTo(String) en la API de JAVA
8. obligar al sistema a hacer un punto y aparte (pasar de línea) al final del ejercicio

# **Ej B3\_2**

Escriba un programa Java para obtener el primer carácter y el décimo de la cadena siguiente: Estamos practicando!!

|  |
| --- |
| Cadenas |

1. Longitud de la cadena
2. Comparación de cadenas
3. Obtener una subcadena
4. Poner en mayúsculas
5. Obtener un carácter de una cadena

String cadena1="EJEMPLO CADENA";

String cadena2="ejemplo cadena";

// Longitud de una cadena función .length()

longitudcadena1=cadena1.length();

// Comparación de cadenas

System.***out***.println("\n\tComparación: "+(cadena1==cadena2));

System.***out***.println("\tComparación: "+(cadena1.equals(cadena2)));

cadena1.compareTo(cadena2)))

cadena1.substring(0,5));

cadena1.toLowerCase()

cadena3.charAt(0)

Tarea B4

# **Ej B4\_1**

Escriba un programa que pida al usuario que introduzca 3 números enteros (cuadro de diálogo).

A continuación, el programa debe mostrar la suma, el promedio y el producto de estos 3 números en un cuadro de mensaje.

El cálculo de la media debe dar un resultado entero.

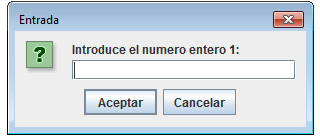
# **Ej B4\_2**

Usando la clase JOption Pane, pedir al usuario que inserte por teclado 3 números enteros. A continuación, resolver la ecuación de primer grado aX+b=c sacando el valor de X y sabiendo que a, b y c son las entradas por teclado.

# **Nota ( Gorka )**

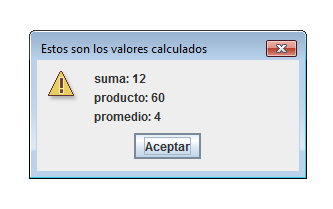
**Es un ejercicio para ver cómo tratar la pérdida de información en el resultado**

Para solicitar datos JOptionPane.*showInputDialog*



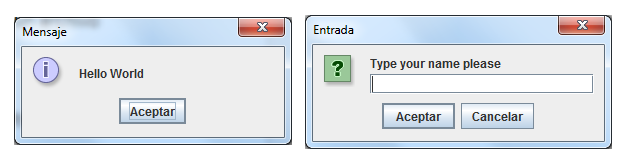
Para sacar mensajes

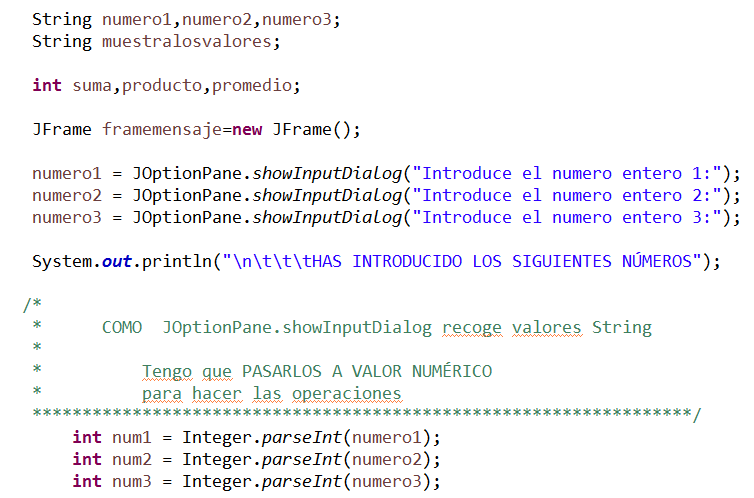
JOptionPane.*showMessageDialog*

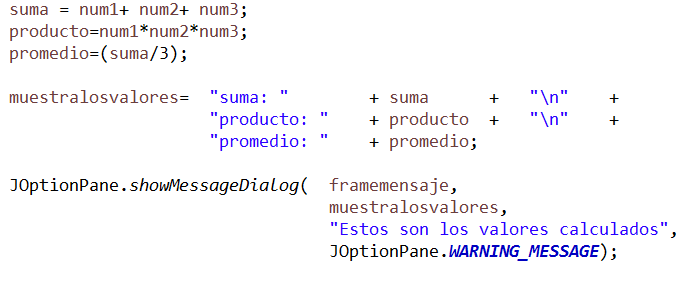


|  |
| --- |
| Cuadros de diálogo |

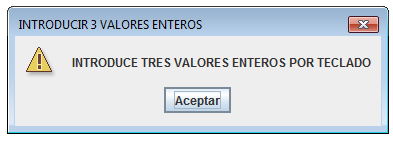
**import** javax.swing.JOptionPane;

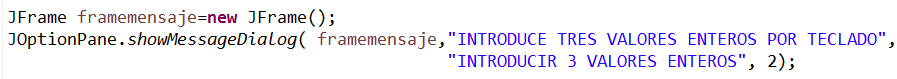


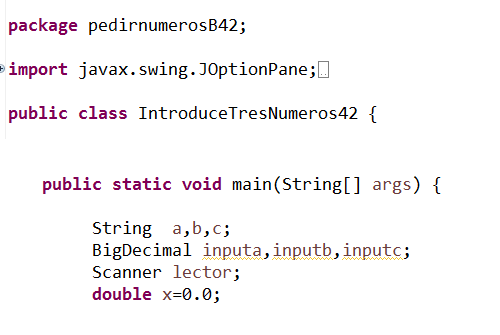


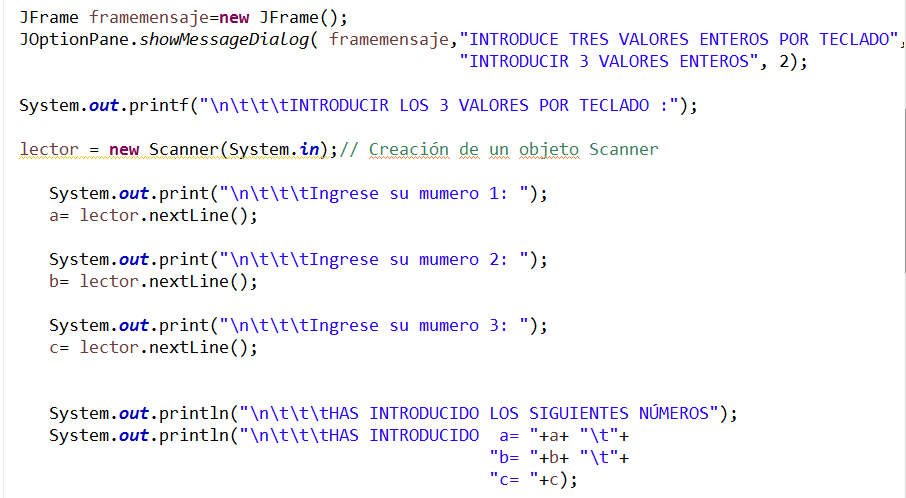


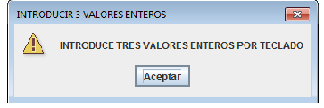
# Ej B4\_2

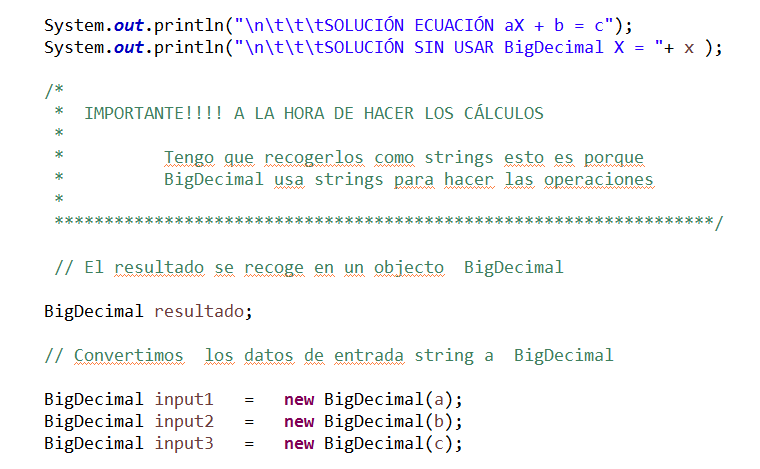


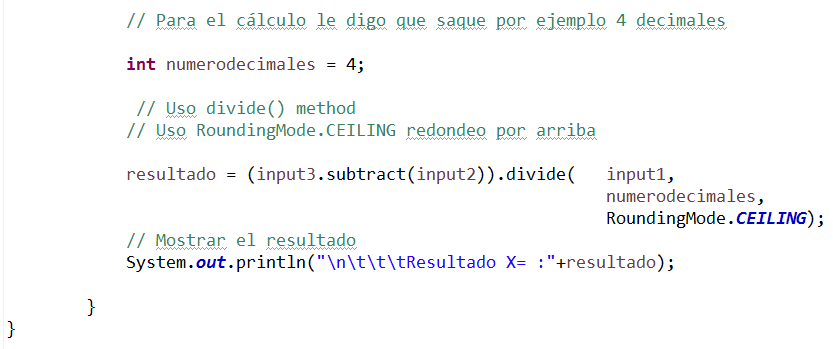


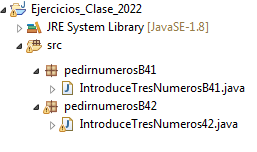








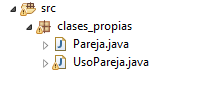




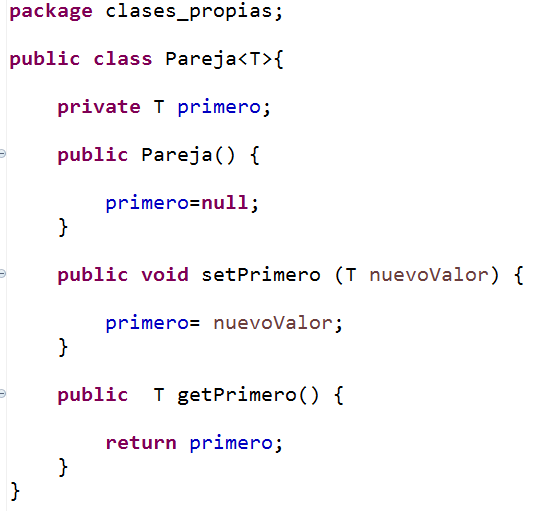
|  |
| --- |
| Java Programación genérica  Creación clases genéricas propias  Píldoras informáticas VIDEO 165 |

En este video explica la utilización de las clases genéricas y para ver su funcionamiento en el ejmplo:

1. usa la clase con un dato tipo String
2. usa la clase con un dato tipo objeto



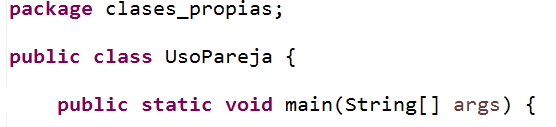
Definición de la clase genérica Pareja.java



Para indicar que es una clase genérica por convenio se escriben en mayúsculas y se suelen usar las letras T U K

Veamos ahora como se usa esta clase genérica

1. Usando dato tipo **String**



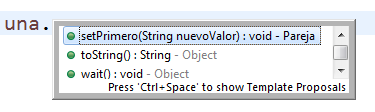
Creamos una nueva instancia de la case Pareja



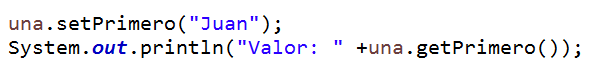
En este momento primero = null

Para asignarle un nuevo valor usaremos su método

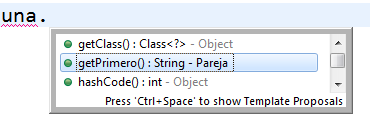
**public** **void** setPrimero (T nuevoValor)



Vemos la "**MAGIA**" de que la clase esté programada como genérica al hacer **una.** se despliegan todos los métodos de los que dispone y Eclipse ya nos está mostrando que el setPrimero requiere que el valor que se le pase sea de tipo String.

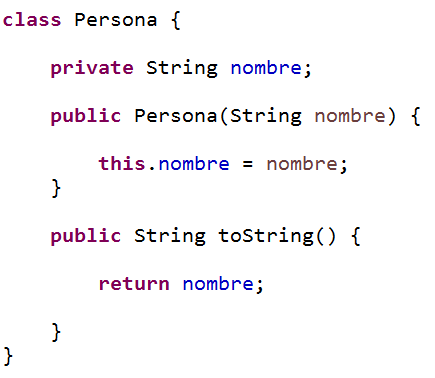


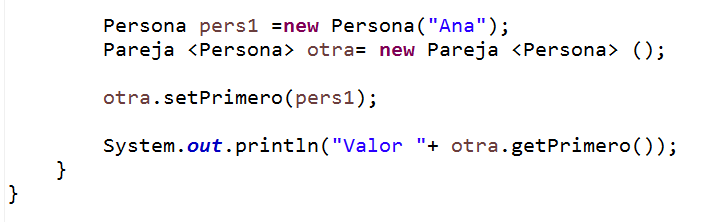
Lo mismo pasa cuando queremos recuperar el valor de la variable usando el método getPrimero(), Eclipse nos dice que devuelve un String



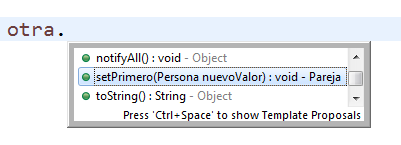
1. Usando dato tipo **Pareja**

**Esta es la clase Persona**

****

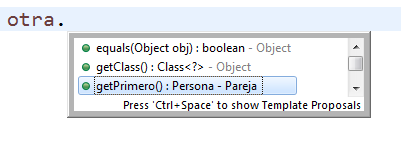
****

Pasa como antes al hacer para el método setPrimero



Vemos que se le tiene que pasar un valor tipo Persona

Para el método getPrimero devuelve un objeto Persona

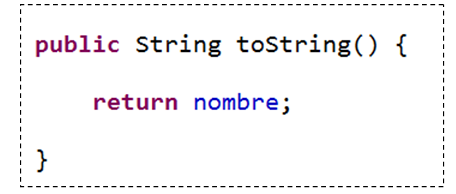


Salida cuando mostramos un objeto Persona



Salida si queremos que nos muestre el valor de primero en este caso

hay que sobreescribir el método toString() en la clase Persona



Salida



package clases\_propias;

public class **Pareja<T>**{

private T primero;

public Pareja() {

primero=null;

}

public void setPrimero (T nuevoValor) {

primero= nuevoValor;

}

public T getPrimero() {

return primero;

}

}

public class **UsoPareja** {

public static void main(String[] args) {

Pareja <String> una = new Pareja <String>();

una.setPrimero("Juan");

System.out.println("Valor una: " +una.getPrimero());

// Ahora vamos a trabajar con valores tipo Pareja

Persona pers1 =new Persona("Ana");

Pareja <Persona> otra= new Pareja <Persona> ();

otra.setPrimero(pers1);

System.out.println("Valor otra: "+ otra.getPrimero());

}

}

class **Persona** {

private String nombre;

public Persona(String nombre) {

this.nombre = nombre;

}

public String toString() {

return nombre;

}

}