+25 ejemplos de código con Java

1. Hello World

Dentro de todos los programas sencillos de Java, este es el primero que debes aprender. Es el primero de los códigos de Java que casi cualquier programador compila. Acá su código:

```java
class HelloWorld
{
   public static void main(String args[])
   {
      System.out.println("Hello World");
   }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta

- **La definición de la clase (class):** se utiliza para definir una nueva clase a través de la palabra clave “class” en los códigos de Java. Debe estar comprendida entre llaves: Apertura “{” y cierre “}” en los códigos de Java.
- **Método principal (main):** cada aplicación de Java debe contener el método, en este caso public static void main (String [ ] args)
- **Public:** se coloca para que pueda ejecutarse en cualquier lugar.
- **Static:** con esto el método principal puede ser llamado sin crear el objeto.
- **Void:** el método principal no devuelve nada.
- **String:** el método principal acepta un único argumento.

+25 ejemplos de código con Java

2. Área de un círculo

El siguiente código de Java te será de muchísima utilidad, por lo que te recomendamos prestarle mucha atención.

```java
import java.util.Scanner;
class AreaOfCircle
{
   public static void main(String args[])
    {

      Scanner s= new Scanner(System.in);

         System.out.println("Enter the radius:");
         double r= s.nextDouble();
         double  area=(22*r*r)/7 ;
         System.out.println("Area of Circle is: " + area);
   }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta

- Para obtener el área de un círculo, debemos definir la fórmula que, como sabemos, consiste en A = r2 π.

- Además, le asignamos el valor de 22/7 al símbolo griego de “pi”.

- El “import” es una palabra clave con la cual podemos obtener ciertas características de paquetes incorporados. Acá usamos el paquete “util” que incluye muchas clases. Una de esas clases es “Scanner”, la cual utilizamos para obtener comandos a través de la consola. Recuerda que la consola es la interfaz entre el usuario y el programa.

- Luego podemos ver la función principal que ya te explicamos detalladamente de “public static void main”.

- La clase scanner es la encargada de escanear los datos de entrada que el usuario dio a través de la consola. Debemos crear un objeto que sea la referencia para almacenar la variable “s”. (Para obtener acceso a la consola debemos crear un objeto “new scanner”).

- Ahora con System.out.println (“Enter the radius:”); pedimos el valor del radio que el usuario introduce por teclado.

+25 ejemplos de código con Java

- Utilizamos “nextdouble”, ya que es el formato que necesitamos para que la consola lea el número que le estamos introduciendo. Además, como el número de pi es un número con decimales, es necesario poner “nextdouble” en vez de “nextint” (formato que no acepta decimales, sino solo números enteros).

3. Área del triángulo

Si quieres resolver el área del triángulo con Java, debemos crear una clase, y dentro de esta clase poner un método main. Veamos como quedarían esas líneas de código en Java:

```
import java.util.Scanner;
class AreaOfTriangle
{
   public static void main(String args[])
    {

      Scanner s= new Scanner(System.in);

         System.out.println("Enter the width of the Triangle:");
         double b= s.nextDouble();

         System.out.println("Enter the height of the Triangle:");
          double h= s.nextDouble();

                  //Area = (width*height)/2
      double area=(b*h)/2;
      System.out.println("Area of Triangle is: " + area);
   }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta

- Para el caso del área de un triángulo, podríamos decir que su código en Java es muy similar al del área de un círculo. Tiene un scanner para leer las entradas por consola y utilizamos “nextdouble”, ya que los valores introducidos pueden ser decimales al igual que en el caso anterior.

- Será suficiente con pedir valores como la altura y la base del triángulo por consola para obtener su área, haciendo uso de la fórmula Atri = (base * altura)/2.

+25 ejemplos de código con Java

4. Área del triángulo isósceles

¿Cómo podríamos obtener el área de triángulos isósceles? ¡Veámoslo a continuación en el próximo código de Java!

```java
import java.util.Scanner;
class AreaOfIsoscelesTriangle
{
   public static void main(String args[])
    {

         Scanner s= new Scanner(System.in);

         System.out.println("Enter the length of same sided");

         double a= s.nextDouble();

         System.out.println("Enter the side2 of the Triangle");

         double b= s.nextDouble();

         double  area=(b/4)*Math.sqrt((4*a*a)-(b*b));

         System.out.println("Area of Isosceles Triangle is: " + area);
   }
}
```

Fuente: Javatutoring

5. Área del triángulo equilátero

¿Y si quisiéramos un triángulo equilátero?

```java
import java.util.Scanner;
class AreaOfEquilateralTriangle
{
   public static void main(String args[])
    {

         Scanner s= new Scanner(System.in);
         System.out.println("Enter the side of the Triangle:");
         double a= s.nextDouble();
         double  area=(Math.sqrt(3)/4)*(a*a);
         System.out.println("Area of Triangle is: " + area);
    }
}
```

+25 ejemplos de código con Java

6. Área del rectángulo

Ahora, veamos cómo quedaría el área del rectángulo en Java:

```java
import java.util.Scanner;
class AreaOfRectangle
{
   public static void main(String args[])
    {

      Scanner s= new Scanner(System.in);

         System.out.println("Enter the length:");
         double l= s.nextDouble();
        System.out.println("Enter the breadth:");
         double b= s.nextDouble();


          double  area=l*b;
      System.out.println("Area of Rectangle is: " + area);
   }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta

- Finalmente, y dentro de las figuras geométricas planas, para el área del rectángulo seguiremos los mismos parámetros: una clase, el bloque principal compuesto por “public static void main”, un escáner y el formato de nextdouble.

- De esta manera, el código de Java imprimirá el área del rectángulo en pantalla siguiendo la fórmula matemática de A = base * altura.

+25 ejemplos de código con Java

7. Perímetro del rectángulo

¿Qué tal si, en vez del área, intentamos calcular el perímetro del rectángulo con códigos de Java?

Básicamente, la variación que sufre el código de Java es en la fórmula. La fórmula viene definida por P = 2(base * altura) como se ve a continuación:

```java
import java.util.Scanner;
class PerimeterOfRectangle
{
   public static void main(String args[])
    {

         Scanner s= new Scanner(System.in);

         System.out.println("Enter the length of the Rectangle:");

     double l= s.nextDouble();

     System.out.println("Enter the width of the Rectangle:");

     double b= s.nextDouble();

         double  perimeter=2*(l+b);

     System.out.println("perimeter of Rectangle is: " + perimeter);
   }
}
```

Fuente: Javatutoring

+25 ejemplos de código con Java

8. Perímetro de un cuadrado

En el caso del rectángulo, necesitamos saber tanto base como altura. En este caso, los lados de un cuadrado son todos de la misma longitud por lo que, con saber la longitud del lado, obtendremos su perímetro. Observa el código en Java:

```java
import java.util.Scanner;
class PerimeterOfSquare
{
   public static void main(String args[])
    {

         Scanner s= new Scanner(System.in);

         System.out.println("Enter the side of the square:");

     double a= s.nextDouble();

         double  perimeter=4*a;

     System.out.println("perimeter of Square is: " + perimeter);
   }
}
```

Fuente: Javatutoring

+25 ejemplos de código con Java

9. Volumen de una esfera

Hasta ahora, en los códigos de Java, hemos trabajado con áreas de superficies planas, pero también podemos hallar volúmenes en superficies sólidas.

```java
import java.util.Scanner;
class VolumeOfSphere
{

   public static void main(String args[])
    {


                Scanner s= new Scanner(System.in);
            System.out.println("Enter the radius of sphere:");
            double r=s.nextDouble();


                 double  volume= (4*22*r*r*r)/(3*7);

             System.out.println("Volume is:" +volume);
      }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta:

- A pesar de que hablamos de una superficie sólida, el procedimiento sigue siendo bastante similar. La primera línea de código sigue conteniendo el “import”, para obtener ciertas características y paquetes incorporados. Igualmente, en este caso, seguimos utilizando el paquete “util” y el formato “nextdouble”.

- En conclusión, lo único que cambia es la fórmula utilizada y los datos requeridos para obtener el resultado que se ve en pantalla.

Y así como obtenemos áreas complejas como una esfera, podemos encontrar valores de muchas superficies sólidas o planas: hexágonos, pentágonos, cilindros, etc.

+25 ejemplos de código con Java

10. Calcular el promedio de “n” números

Siempre que dispongamos de un valor numérico, debemos introducir una fórmula al código. En este caso, decimos que el promedio de “n” números es: la suma de todos los valores / la cantidad de valores.

```
1  class Average
2  {
3   public static void main(String arg[])
4   {
5     int n=5,result=0;
6
7     int a[]=new int[5];
8
9     a[0]=10;
10
11    a[1]=20;
12
13    a[2]=30;
14
15    a[3]=40;
16
17    a[4]=50;
18
19    for(int i=0;i<n;i++)
20     result=result+a[i];
21
22    System.out.println("average of  ("+a[0]+","+a[1]+","+a[2]+","+a[3]+","+a[
23
24   }
25 }
```

Fuente: Javatutoring

Pasos a tener en cuenta

1. Comenzamos asignando un nombre y una clase e iniciamos el algoritmo con “public static void main”.

2. En este caso, definimos la variable “n” bajo el formato de “int”, ya que los números a trabajar son enteros.

3. Posteriormente, en la línea 12 podemos ver cómo asignamos le pedimos al usuario que inserte la cantidad de números a los que le calcularemos el promedio.

4. Nuestro siguiente paso estará basado en el ciclo “for”, para ir agregando los números.

5. Finalmente, y una vez hayamos insertado todos los números, nuestro programa se ejecutará y mostrará en pantalla el promedio de los números insertados.

11. Calcular el promedio de bateo

Modificando un poco el caso del promedio, podemos mirar cómo se calcula el promedio de un bateador a través de un código de Java:

```
class BattingAverage
{
  public static void main(String arg[])
  {
     int Matches=5,totalruns=200,innings=5,notout=1;

     double avg;

     avg=totalruns/(innings-notout);

     System.out.println("batting average="+avg);

  }
}
```

Fuente: Javatutoring

+25 ejemplos de código con Java

12. Suma de dos números

Continuando la tendencia de los códigos de Java más sencillos, podemos continuar con la suma de sólo dos números. Para ello, empleamos el siguiente código:

```java
import java.util.Scanner;
class Add
{
public static void main(String[] arg)
{
int a,b,c;
Scanner sc=new Scanner(System.in);
System.out.println("Enter first number");
a=sc.nextInt();
System.out.println("Enter second number");
b=sc.nextInt();
c=addition(a,b);
System.out.println(" Addition of two numbers is : "+c);
}
static int addition(int x,int y)
{
    return x+y;
}
}
```

Fuente: Javatutoring

Puntos a tener en cuenta

- Al igual que en los casos anteriores, definimos nombre y clase. Iniciamos el código y definimos 3 variables (dos que solicitaremos al usuario y una será el resultado de las anteriores).

- El usuario registra los números y el programa realiza la suma.

Ahora bien, ¿qué pasa si en vez de 2 números quiero hacer la suma de una cantidad “n” de símbolos”?

+25 ejemplos de código con Java

13. Suma de “n” números.

¿Se te ha ocurrido alguna idea de cómo generar el mismo algoritmo pero, en vez de 2 números, sumar la cantidad que tú quieras? Tal vez ya sepas la respuesta (y es que anteriormente hicimos un algoritmo similar). Si tu respuesta es que es un ciclo, estás en lo correcto. Mira cómo se hace:

```
import java.util.Scanner;
class sum
{
    public static void main(String arg[])
    {
                    int n,sum=0;

                    Scanner sc=new Scanner(System.in);

        System.out.println("enter how many numbers you want sum");
                     n=sc.nextInt();
                     int a[]=new int[n];
        System.out.println("enter the "+n+" numbers ");
                     for(int i=0;i<n;i++)
                     {
                   System.out.println("enter  number  "+(i+1)+":");
                              a[i]=sc.nextInt();
                     }
                     for(int i=0;i<n;i++)
                     {
                              sum+=a[i];
                     }
                     System.out.println("sum of "+n+" numbers is ="+sum);
                  }
}
```

Fuente: Javatutoring

+25 ejemplos de código con Java

14. Calcular la factura de la electricidad

Otra herramienta que podemos desarrollar a través de los códigos de Java es la factura de la electricidad. Te mostramos el procedimiento del código en la siguiente imagen:

```java
import java.util.*;
class ComputeElectricityBill
{
        public static void main(String args[])
        {
    int units=280;

    double billpay=0;

            if(units<100)
     {
          billpay=units*1.20;
      }
    else if(units<300)
     {
       billpay=100*1.20+(units-100)*2;
     }
    else if(units>300)
    {
        billpay=100*1.20+200*2+(units-300)*3;
    }

    System.out.println("Bill to pay : " + billpay);
      }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta:

- La única diferencia existente entre este y los otros códigos de Java que ya estudiamos, es que añadimos las condiciones de “if” y de “if else”.

- Estas líneas se ejecutarán, siempre que una condición ejecutada por el usuario se cumpla.

+25 ejemplos de código con Java

15. Calcular el descuento de un producto

En ocasiones, queremos saber el descuento de un producto para obtener su producto final. Para ello podemos usar un simple código de Java que nos facilite este cálculo en cuestión de segundos. A continuación, te mostramos cómo se realiza:

```java
import java.util.*;
class ComputeElectricityBill
{
        public static void main(String args[])
        {
    int units=280;

    double billpay=0;

            if(units<100)
     {
          billpay=units*1.20;
      }
    else if(units<300)
     {
       billpay=100*1.20+(units-100)*2;
     }
    else if(units>300)
    {
        billpay=100*1.20+200*2+(units-300)*3;
    }

    System.out.println("Bill to pay : " + billpay);
      }
}
```

Fuente: Javatutoring

Puntos a tener en cuenta

- Los únicos valores que el código necesita son el precio del artículo y su descuento.

+25 ejemplos de código con Java

16. Año bisiesto

A través del uso de las condiciones que ofrece “if”, se muestra como es el procedimiento para saber si un año es bisiesto o no. Recuerda que el objetivo es obtener agilidad mental, para crear cualquier algoritmo que quieras o debas hacer.

```
import java.util.Scanner;
class Leapyear
{
    public static void main(String arg[])
    {
        long a,y,c;
                    Scanner sc=new Scanner(System.in);
        System.out.print("enter  any  calendar year :");
                   y=sc.nextLong();
        if(y!=0)
        {
        a=(y%400==0)?(c=1):((y%100==0)?(c=0):((y%4==0)?(c=1):(c=0)));
        if(a==1)
            System.out.println(y+" is a leap year");
                else
            System.out.println(y+" is not a leap year");
        }
        else
        System.out.println("year zero does not exist ");
    }
}
```

Fuente: Javatutoring

+25 ejemplos de código con Java

17. Obtener la distancia entre dos puntos

Lo más “complejo” (en realidad no lo es) en este particular es conocer la fórmula necesaria para llevar a cabo este logaritmo. Así que pon especial atención en la fórmula utilizada.

Tip: sqrt = raíz cuadrada de un número.

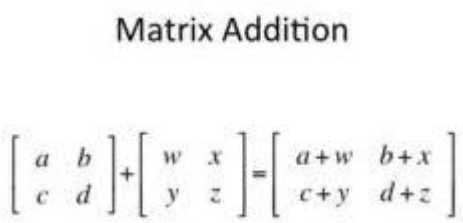
```
import java.lang.Math.*;
class DistanceBwPoint
{
    public static void main(String arg[])
    {
            int x1,x2,y1,y2;
        double dis;
        x1=1;y1=1;x2=4;y2=4;
        dis=Math.sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
            System.out.println("distancebetween"+"("+x1+","+y1+"),"+"("+x2+"
    }
}
```

Fuente: Javatutoring

18. Suma de dos matrices

Otro ejemplo en este top de códigos de Java, tiene que ver con el mundo de las matrices. Estas cumplirán ciertos requisitos para ejecutarse correctamente:

- Deben ser del mismo tamaño e insertados con el mismo orden.
- Aparte de la matriz, cada término de una de estas se agrega al término de la otra, en la misma ubicación. ¿Un poco engorroso, cierto? No te preocupes, te mostramos un ejemplo gráfico de cómo funciona esto sumar con el código en Java:



Fuente: Javatutoring

+25 ejemplos de código con Java

A continuación la suma de matrices en código de Java:

```java
import java.util.Scanner;

class AddMatrix
{
public static void main(String args[])
{
int row, col,i,j;
Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows");
row = in.nextInt();

System.out.println("Enter the number columns");
col = in.nextInt();

int mat1[][] = new int[row][col];
int mat2[][] = new int[row][col];
int res[][] = new int[row][col];

System.out.println("Enter the elements of matrix1");

for ( i= 0 ; i < row ; i++ )
{

for ( j= 0 ; j < col ;j++ )
mat1[i][j] = in.nextInt();

System.out.println();
}
System.out.println("Enter the elements of matrix2");

for ( i= 0 ; i < row ; i++ )
{

for ( j= 0 ; j < col ;j++ )
mat2[i][j] = in.nextInt();

System.out.println();
}

for ( i= 0 ; i < row ; i++ )
for ( j= 0 ; j < col ;j++ )
res[i][j] = mat1[i][j] + mat2[i][j] ;

System.out.println("Sum of matrices:-");

for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
System.out.print(res[i][j]+"\t");

System.out.println();
}
```

Fuente: Javatutoring

+25 ejemplos de código con Java

19. Resta de dos matrices

En este caso, la resta de matrices en Java debe realizarse entre dos matrices del mismo orden. Te lo demostramos de forma gráfica para que sea más comprensible:

$$A = \begin{pmatrix} 5 & 10 & 20 \\ 8 & 6 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 8 & 5 \\ 2 & 9 & 3 \end{pmatrix}$$

Subtraction of two matrixes:

$$A - B = \begin{pmatrix} 5-3 & 10-8 & 20-5 \\ 8-2 & 6-9 & 5-3 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 15 \\ 6 & -3 & 2 \end{pmatrix}$$

Fuente: Javatutoring

+25 ejemplos de código con Java

Por otro lado, la resta de matrices se plantea de la siguiente manera:

```
import java.util.Scanner;

class SUBMatrix
{
   public static void main(String args[])
   {
      int row, col,i,j;
      Scanner in = new Scanner(System.in);

      System.out.println("Enter the number of rows");
      row = in.nextInt();

      System.out.println("Enter the number  columns");
      col  = in.nextInt();

      int mat1[][] = new int[row][col];
      int mat2[][] = new int[row][col];
      int res[][] = new int[row][col];

      System.out.println("Enter the elements of matrix1");
    i=0;
     do
       {
       j=0;
            do
           {
                mat1[i][j] = in.nextInt();
                 j++;
           }while(j < col);
         i++;

       } while ( i < row);
      System.out.println("Enter the elements of  matrix2");


     i=0;
     do
       {
       j=0;
            do
           {
                mat2[i][j] = in.nextInt();
                 j++;
           }while(j < col);
            i++;

       } while ( i < row);

   i=0;
      do
       {
       j=0;
            do
```

Fuente: Javatutoring

+25 ejemplos de código con Java

20. Multiplicación de matrices

En este otro ejemplo de los códigos de java, podemos destacar la multiplicación de matrices bajo el siguiente código:

```
import java.util.Scanner;
class MUlMatrix
{
public static void main(String args[])
{
int r1, r2,c1,c2,i,j,k,sum;
Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows of matrix1");
r1 = in.nextInt();

System.out.println("Enter the number columns of matrix 1");
c1 = in.nextInt();
System.out.println("Enter the number of rows of matrix2");
r2 = in.nextInt();

System.out.println("Enter the number of columns of matrix 2");
c2 = in.nextInt();

if(c1==r2)
{

int mat1[][] = new int[r1][c1];
int mat2[][] = new int[r2][c2];
int res[][] = new int[r1][c2];

System.out.println("Enter the elements of matrix1");

for ( i= 0 ; i < r1 ; i++ )
{

for ( j= 0 ; j < c1 ;j++ )
mat1[i][j] = in.nextInt();

}
System.out.println("Enter the elements of matrix2");

for ( i= 0 ; i < r2 ; i++ )
{

for ( j= 0 ; j < c2 ;j++ )
mat2[i][j] = in.nextInt();

}

System.out.println("\n\noutput matrix:-");
for ( i= 0 ; i < r1 ; i++ )

for ( j= 0 ; j <c2;j++)
{
sum=0;
for ( k= 0 ; k <r2;k++ )
{
```

Fuente: Javatutoring

+25 ejemplos de código con Java

21. TableModel en JTable

Vamos a ver cómo utilizar un JTable de java. Se trata de un componente visual de Java que permite dibujar una tabla y agregar datos en cada fila y columna.
Además de usar otros constructores que hay en JTable, una de las formas más sencillas de utilizar un JTable consiste en instanciar como modelo de datos un DefaultTableModel y luego un JTable, pasándole el modelo al constructor. El código quedaría así:

```
DefaultTableModel modelo = new DefaultTableModel();
JTable tabla = new JTable (modelo);
```

Imagen: chuidiang

Ahora puedes añadir columnas a nuestros datos con el código de Java: datos.addColumn ("Nombre columna");
Y luego puedes añadir, borrar y modificar datos así:

- datos.addRow (arrayConLosDatosParaUnaFila);
- datos.removeRow (fila);
- datos.setValueAt (dato, fila, columna);

Cuando hagas modificaciones, el JTable actualizará el resultado automáticamente.

22. Arrancar un programa externo desde Java

Puede pasar que alguna vez necesites llamar a ejecutables externos, por ejemplo: lanzar el Acrobat Reader para ver un pdf desde el navegador. Para esas situaciones, este ejemplo de código de Java te servirá un montón.

Es importante considerar que esta práctica no es muy recomendable, ya que se pierde el objetivo de Java que es poder ejecutar el programa en cualquier plataforma. Sin embargo, a veces es necesaria.

El código quedaría así:

```
try
{
   /* directorio/ejecutable es el path del ejecutable y un nombre */
   Process p = Runtime.getRuntime().exec ("directorio/ejecutable");
}
catch (Exception e)
{
   /* Se lanza una excepción si no se encuentra en ejecutable o el fi
}
```

Imagen: chuidiang

+25 ejemplos de código con Java

23. Verificar número compuesto

Este código de Java, propuesto en el blog **Java desde Cero**, permite verificar si un número es compuesto. Solo tienes que cambiar las declaraciones de devolución (return). El return true se cambia a devolver false y viceversa. ¡Es muy sencillo!

```
// Un método optimizado basado en Java
// para comprobar si un número es compuesto o no.

class NumeroCompuesto
{
    static boolean esCompuesto(int n)
    {
        // Casos especiales
        if (n == 1)
            System.out.println("Falso");

        if (n == 3)
            System.out.println("Falso");

        // Esto se verifica para que podamos omitir
        // los cinco números intermedios en el bucle inferior
        if (n % 2 == 0 || n % 3 == 0) return true;

        for (int i = 5; i * i <= n; i = i + 6)
            if (n % i == 0 || n % (i + 2) == 0)
                return true;

        return false;
    }

    public static void main(String args[])
    {
        System.out.println(esCompuesto(28) ?
                "Sí, ¡El número es un número compuesto!" : "No es compuesto, es primo");

        System.out.println(esCompuesto(19) ?
                "Sí, ¡El número es un número compuesto!" : "No es compuesto, es primo");
    }
}
```

Imagen: Java desde cero

+25 ejemplos de código con Java

El resultado de este código de Java arroja dos resultados:

- Sí, ¡el número es un número compuesto!
- No es compuesto, es primo

24. Verificar si una URL es válida

Estos códigos de Java también fueron publicados en el blog Java desde Cero, y se trata de tres métodos para verificar si una dirección web es válida o no.

El resultado esperado en los tres métodos es: “La url dada (url) es válida” o “La url dada (url) no es válida”

A continuación, te contamos cuáles son estos códigos para Java.

Validar URL con Apache Commons Validator

Con el paquete Apache Commons Validator, se puede usar la clase URLValidator para validar la URL, comprobando el esquema, la autoridad, la ruta, la consulta y el fragmento. Así quedaría el código:

```
// Programa Java para verificar si una URL es válida usando Apache
// common validator.
import org.apache.commons.validator.routines.UrlValidator;

class URLValidator {
    public static boolean urlValidator(String url)
    {
        /* Obteniendo UrlValidator */
        UrlValidator defaultValidator = new UrlValidator();
        return defaultValidator.isValid(url);
    }

    public static void main(String[] args)
    {
        String url = "https://www.javadesdecero.es/";

        /* validar una url */
        if (urlValidator(url))
            System.out.print("La url dada " + url + " es válida");
        else
            System.out.print("La url dada " + url + " no es válida");
    }
}
```

Imagen: Java desde cero

+25 ejemplos de código con Java

Validar URL con java.net.url

Otra opción para validar una URL con Java es usar java.net.url. El fin es crear un objeto URL a partir de la representación String especificada.

Así, se lanzará un error MalformedURLException si no se especifica ningún protocolo, si se encuentra un protocolo desconocido o si la especificación es nula. Luego, llamaremos al método toURI() que arrojará una URISyntaxException si la URL no se puede convertir a URI.

```
// Programa Java para verificar si una URL es válida usando
// java.net.url
import java.net.MalformedURLException;
import java.net.URISyntaxException;
import java.net.URL;

class URLValidator {

    public static boolean urlValidator(String url)
    {
        /*validación de url*/
        try {
            new URL(url).toURI();
            return true;
        }
        catch (URISyntaxException exception) {
            return false;
        }

        catch (MalformedURLException exception) {
            return false;
        }
    }

    public static void main(String[] args)
    {
        String url = "https://www.javadesdecero.es/";

        /* validar la url */
        if (urlValidator(url))
            System.out.print("La url dada " + url + " es válida");
        else
            System.out.print("La url dada " + url + " no es válida");
    }
}
```

Imagen: Java desde cero

+25 ejemplos de código con Java

Validar URL con expresión regular por owasp

La idea es simple usando la expresión regular por owasp. El código de Java quedaría así:

```
// Programa Java para verificar si una URL es válida usando
// regular expression por owasp
import java.util.regex.Matcher;
import java.util.regex.Pattern;

class URLValidator {
    /*expresión regular*/
    private static final String URL_REGEX =
            "^((((https?|ftps?|gopher|telnet|nntp)://)|(mailto:|news:))" +
                    "(%{2}|[-()_.!~*';/?:@&=+$, A-Za-z0-9])+)" + "([).!';/?

    private static final Pattern URL_PATTERN = Pattern.compile(URL_REGEX);

    public static boolean urlValidator(String url)
    {

        if (url == null) {
            return false;
        }

        Matcher matcher = URL_PATTERN.matcher(url);
        return matcher.matches();
    }

    public static void main(String[] args)
    {
        String url = "https://www.javadesdecero.es/";

        /* validar la url */
        if (urlValidator(url))
            System.out.print("La url dada " + url + " es válida");
        else
            System.out.print("La url dada " + url + " no es válida");
    }
}
```

Imagen: Java desde cero

+25 ejemplos de código con Java

25. Contar elementos repetidos

Este ejemplo de código en Java seguro te será de utilidad ya que sirve para contar elementos repetidos. Te lo dejamos a continuación:

```
//Codificado por sAfOrAs
public class SyGContarElementosRepetidos{
    static int A=10;
    static int B=20;
    static int vectorA[]=new int[A];
    static int vectorB[]=new int[B];
    static int elemA=0;
    static int elemB=0;
    static int z=0;
    static void llenaArreglo(){
        for(int i=0;i<vectorA.length;i++){
            vectorA[i]=(int)(Math.random()*100+1);
        }

        for(int i=0;i<vectorB.length;i++){
            vectorB[i]=(int)(Math.random()*100+1);
        }

    }

    static void compara(){
        for(int i=0;i<vectorA.length;i++){
            for(int j=0;j<vectorB.length;j++){
                if(vectorA[i]==vectorB[j])
                    elemA++;
            }
            System.out.println("El elemento "+vectorA[i]+
            elemA=0;
        }
    }

    static void imprimir(){
        for(int i=0;i<vectorA.length;i++){
            System.out.print("A"+"["+i+"]= "+vectorA[i]+"
        }
        for(int i=0;i<vectorB.length;i++){
            System.out.print("B"+"["+i+"]="+vectorB[i]+"\
        }
    }

    public static void main(String[] args){
        llenaArreglo();
        imprimir();
        compara();
    }

}
```

Imagen: beastieux

+25 ejemplos de código con Java

26. Eliminar elementos repetidos

Siguiendo en línea con el código de Java anterior, este ejemplo te permitirá eliminar elementos repetidos. Toma nota:

```java
//Codificado por sAfOrAs
public class SyGEliminaElementosRepetidos{
    public static void main(String[] args)
    {
        SyGELiminaCaracteresNulos cadena= new SyGELiminaC
        char[] arraycar={'p','a','j','a','r','r','a','c',
        String sCadenaSinBlancos="";
        System.out.println(arraycar);

        for(int i=0;i<arraycar.length;i++){
            for(int j=0;j<arraycar.length-1;j++){
                if(i!=j){
                    if(arraycar[i]==arraycar[j]){
                        arraycar[j]=' ';
                    }
                }
            }
        }

        String s = new String(arraycar);
        cadena.eliminaEspacio(s,sCadenaSinBlancos);
    }
}
```

Imagen: beastieux