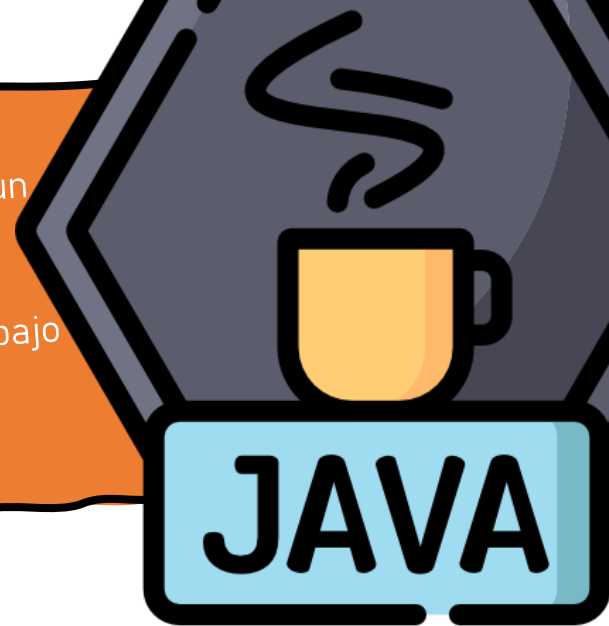


- Creado por James Gosling en 1991 (desarrollador de Sun Microsystems).
- Originalmente llamado Oak y Green.
- Sintaxis basada en C y C++ (con menos utilidades de bajo nivel).
- En 2010 Oracle adquiere Sun Microsystems.



Lenguaje compilado (El resultado se ejecuta en la **máquina virtual de Java**).

Orientado a objetos (Todo código pertenece a una clase).

Fuertemente tipado (Todos los datos/variables tienen un tipo).

Multiplataforma.

Soporte para trabajo en red.

Seguro y de rápida ejecución.

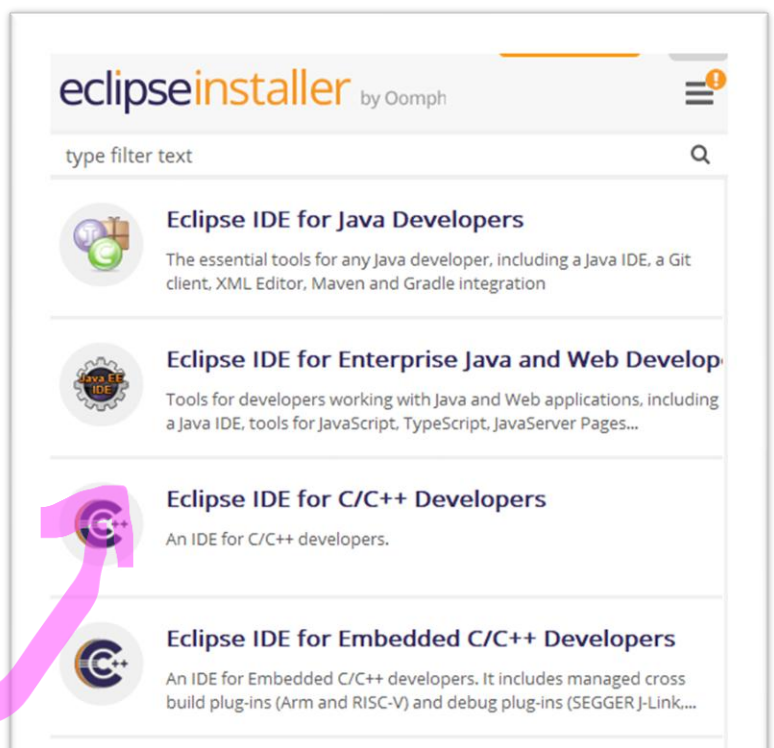
Aplicaciones → Aplicaciones web, software de escritorio (con JavaFX o Swing), aplicaciones móviles aplicaciones que se ejecutan en red (con Java socket).

Entorno de desarrollo.

Necesito JAVA con sus librerías (**JDK**), su máquina virtual (**JVM**) y entorno de ejecución con el compilador (**JRE**).

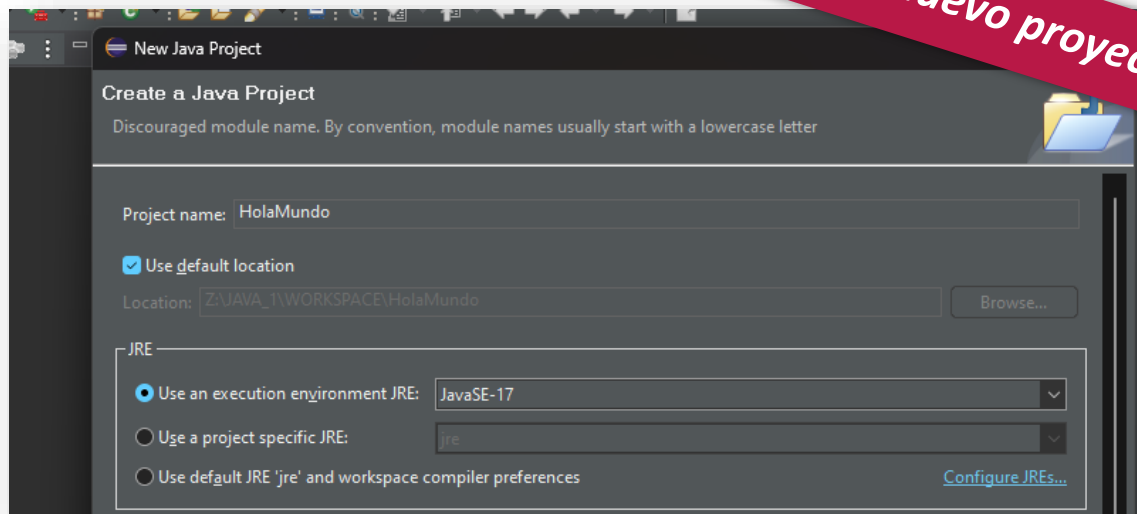
Eclipse (IDE)

Entorno de desarrollo integrado que incluye lo necesario para trabajar con JAVA, además de otros lenguajes.



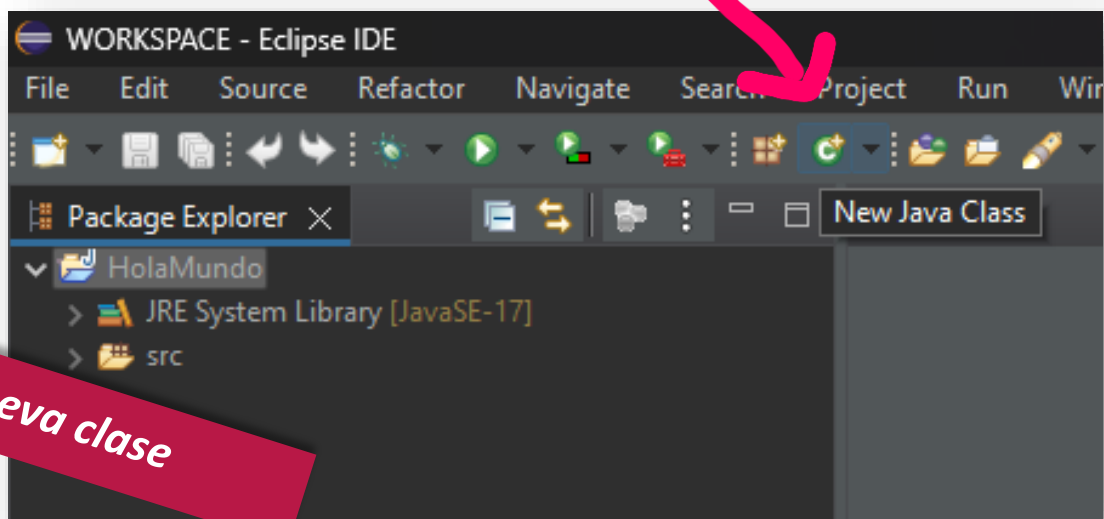
¡Hola mundo! (Ejercicio 0)

Nuevo proyecto



*Todo el código debe estar dentro de alguna clase**

*Todo el código ejecutable debe estar dentro de un método main***



Nueva clase

***** *Una **clase** es un elemento de la programación orientada a objetos que actúa como una plantilla y va a definir las características/comportamientos de un objeto.*

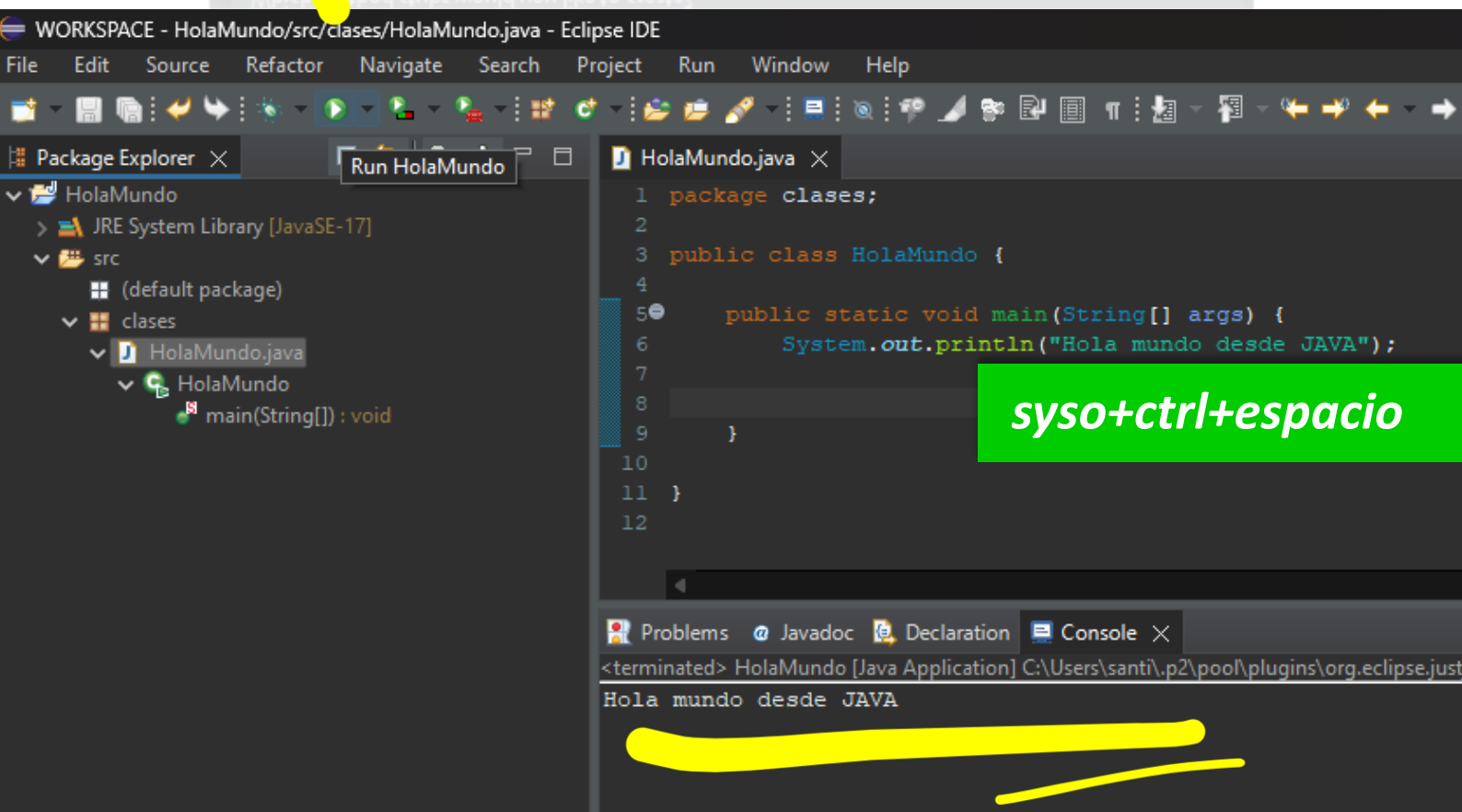
- Una clase es una plantilla para crear objetos.
- Un **objeto** es una instancia de una clase.



Los **paquetes** sirven para facilitar la modularidad del código.

Pueden contener diferentes archivos, correspondientes a definiciones de interfaces o clases. Para utilizar los elementos de un paquete distinto es necesario importarlo.

****método main**



Se pueden agregar **comentarios** al código de la siguiente manera:

```
/*  
Esto es un comentario de múltiples líneas.  
Los comentarios sirven para ordenar el código, explicar el funcionamiento del mismo  
y facilitar el entendimiento posterior a otras personas que tengan que interactuar  
con el programa.  
*/  
public class HolaMundo {  
  
    public static void main(String[] args) {  
        // Instrucción que devuelve un mensaje en pantalla  
        System.out.println("Hola mundo desde JAVA");  
    }  
}
```

Tipos de datos básicos (primitivos):

Tipo de dato	Descripción	Espacio en memoria
byte	Valor entero (-128 ~ 127)	8 bits
short	Valor entero (-32768 ~ 32767)	16 bits
int	Valor entero ($-2^{31} \sim 2^{31}-1$)	32 bits
long	Valor entero ($-2^{63} \sim 2^{63}-1$)	64 bits
float	Valor real	32 bits
double	Valor real doble precisión	64 bits
char	Carácter	8 bits
boolean	Booleano (True,False)	8 bits

Conversión de tipos (JAVA casting)

No hay problema si un tipo de dato de menor rango se convierte a uno de mayor.

byte < short < int < long < float < double

int a = 6;

double b = a;

En caso contrario, se debe realizar un *casting de tipos*.

double b = 5.6;
int a = (int)b;

OPERADORES

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/	División
%	Resto

Aritméticos

Relacionales (resultado booleano)

Operador	Significado
==	Igual
!=	Distinto
>	Mayor
>=	Mayor o igual
<	Menor
<=	Menor o igual

Lógicos (unir comparaciones)

Operador	Significado
	OR: true si alguna condición se cumple.
&&	AND: true si todas se cumplen.
!	NOT: invierte el resultado

De asignación

Operador	Significado
=	Asigna valor a variable.
+=	Suma valor a variable y asigna.
-=	Resta valor a variable y asigna.
*=	Multiplica valor a variable y asigna.
/=	Divide valor a variable y asigna.

Mostrar variables por pantalla

```
package classes;

public class Variables {

    public static void main(String[] args) {

        boolean b = 6>2 && 8==3;
        int num1 = 72;
        String c1="hola";
        System.out.println("El contenido de c1 es: " + c1 + ", el booleano es: "+b+", y el nu"
            + "mero entero es: "+num1);
    }
}
```

Console X Declaration Javadoc Problems
Variables [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18 feb. 2025 13:22:13 - 13:22:13) [pid: 10764]
contenido de c1 es: hola, el booleano es: false, y el numero entero es: 72

Las cadenas en Java son un objeto de la clase string, se delimitan entre comillas dobles y no son un tipo de dato primitivo.

SCANNER

Dentro del paquete java.util, existe la clase Scanner.
Es necesario definir un objeto de tipo Scanner para poder obtener la entrada de datos primitivos.

```
1 package clases;
2
3 import java.util.Scanner; //importo clase scanner del paquete java.util
4
5 public class EntradaSalida {
6
7     public static void main(String[] args) {
8         // Creo un objeto de tipo scanner de nombre miscanner
9         Scanner miscanner = new Scanner (System.in);
10        System.out.println("Introduce un numero entero: ");
11        int numero = miscanner.nextInt();
12        //miscanner.nextLine();
13        //miscanner.nextDouble();
14        //miscanner.nextBoolean();
15        System.out.println("El numero intorducido es: " +numero);
16        miscanner.close();
17    }
18 }
19
```

Console Declaration Javadoc Problems

<terminated> EntradaSalida [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18 feb. 2025 13:24:14 - 13:24:22) [pid: 16900]

Introduce un numero entero:

45

El numero intorducido es: 45

Ejercicio 1

Realizar un programa que pida al usuario su nombre y edad, y muestre con estos datos un **mensaje de bienvenida**.

Ejercicio 2

Realizar un programa conversor de monedas **(UYU a USD)**.

Se pedirá al usuario que introduzca los pesos a convertir y mostrará en pantalla la equivalencia en dólares.

Ejercicio 1

```
package proyectos;
import java.util.Scanner;
public class MensajeBienvenida {

    public static void main(String[] args) {

        Scanner sc = new Scanner (System.in);
        System.out.println("Ingrese su nombre: ");
        String nombre= sc.nextLine();
        System.out.println("Ingrese su edad: ");
        int edad= sc.nextInt();
        System.out.println("Bienvenido " + nombre + " usted tiene " + edad + " años.");
        sc.close();
    }
}
```

```
1 package proyectos;
2 import java.util.InputMismatchException;
3 import java.util.Locale;
4 import java.util.Scanner;
5
6 public class ConvertirMonedas {
7     // Incluye bloque try...catch
8     public static void main(String[] args) {
9         float valor_dolar = 40f; // se agrega f al valor float
10        Scanner sc = new Scanner (System.in);
11        sc.useLocale(Locale.US); //Me aseguro que utilice el punto como separador decimal
12        System.out.println("Ingrese en valor en UYU: ");
13        float pesos;
14
15        try {
16            pesos = sc.nextFloat(); //puede producirse un error en ejecución
17        }
18        catch(InputMismatchException err) { //en caso de producirse determinada excepción, def
19
20            pesos=0f;
21            System.out.println("No ingresó un valor válido");
22        }
23
24        float convertido = pesos/valor_dolar;
25        System.out.println("Equivalen a " +convertido+" dólares");
26        sc.close();
27    }
28 }
29
```

Ejercicio 2

Console Declaration Javadoc Problems
<terminated> ConvertirMonedas [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18 feb. 2025 13:30:08 – 13:30:16)

Ingrese en valor en UYU:
47.64
Equivalen a 1.191 dólares

Ejercicio 3

```
1 package proyectos;
2 //import java.util.Locale;
3 import java.util.Scanner;
4 public class AreaTriangulo {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner (System.in); //System con mayúsculas ya que es una clase
8         int emojicara = 0x1F642; //Valor hexa del emoji
9         //sc.useLocale(Locale.US); //Me aseguro que utilice el punto como separador decimal
10
11         //Utilizo un metodo de la clase Character
12         System.out.println("Ingrese la base del triángulo " + Character.toString(emojicara));
13         float base =sc.nextFloat();
14         System.out.println("Ingrese la altura del triángulo ");
15         float altura =sc.nextFloat();
16         float area = (base*altura)/2;
17         System.out.println("El área ese triángulo es: " + area + " m2");
18         sc.close();
19     }
20 }
21 }
```

```
<terminated> AreaTriangulo [Java Application] C:\Users\santi\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.
Ingrese la base del triángulo ☺
3,3
Ingrese la altura del triángulo
3,3
El área ese triángulo es: 5.4449997 m2
```

Área del triángulo.

try...catch (Excepciones).

Una excepción es un error en tiempo de ejecución que provoca la finalización repentina del programa. Existen varios tipos de excepciones y pueden controlarse mediante un bloque **try/catch**.

```
Ingrese en valor en UYU:
hola
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextFloat(Scanner.java:2496)
    at proyectos.ConvertirMonedas.main(ConvertirMonedas.java:14)
```

```
11 Scanner sc = new Scanner(System.in);
12 sc.useLocale(Locale.US); //Me aseguro que utilice el punto como separador decimal
13
14 System.out.println("Ingrese en valor en UYU: ");
15 float pesos;
16 try {
17     pesos = sc.nextFloat(); //puede producirse un error en ejecución
18 } catch(InputMismatchException err) { //en caso de producirse determinada excepción, defino como actuar
19
20     pesos=0f;
21     System.out.println("No ingresó un valor válido");
22
23 }
24 float convertido = pesos/valor_dolar;
25
26 System.out.println("Equivalen a " +convertido+" dólares");
27
28 sc.close();
29
30
31 }
```

El programa finaliza de forma controlada al detectarse una excepción del tipo InputMismatch (el usuario ingresa un tipo de dato no esperado).

Console X Problems Javadoc Declaration
<terminated> ConvertirMonedas [Java Application] C:\Users\santi\p2\pool\plug
Ingrese en valor en UYU:
hola
No ingresó un valor válido
Equivalen a 0.0 dólares

Pueden utilizarse múltiples bloques catch, uno mismo para varios tipos de excepciones, o utilizar la clase padre Exception para referirse a todas.

```
1 try {
2     // código que puede generar varias excepciones
3 } catch (ArithmeticException | NullPointerException | IOException e) {
4     // código para manejar excepciones de aritmética, puntero nulo y entrada/salida
5 }
```

Ejercicio 4

Programar una calculadora de números enteros, que controle las posibles excepciones.

La calculadora muestra los resultados de sumar, restar, multiplicar y dividir dos enteros (*para elaborar una calculadora más compleja necesitamos utilizar estructuras de control*).

Ejercicio 4

```
package proyectos;

import java.util.Scanner;

public class CalculadoraSimple {

    public static void main(String[] args) {
        int primerNumero, segundoNumero;
        Scanner sc = new Scanner (System.in);
        // ingresa el primer operando
        System.out.println("Ingrese el primer operando: ");

        try {
            // es probable que se produzca una excepcion
            primerNumero = sc.nextInt();
        }
        // ingresa con cualquier tipo de excepcion
        catch (Exception err) {
            System.out.println("Tipo de dato incorrecto. Se
utilizará el valor 1");
            primerNumero=1;
        }

        // se ingresa el segundo entero
        System.out.println("Ingrese el segundo operando: ");

        try {

            segundoNumero = sc.nextInt();
        }
        catch (Exception err) {
            System.out.println("Tipo de dato incorrecto. Se
utilizará el valor 1");
            segundoNumero=1;
        }

        /* se definen y muestran todas las operaciones utilizando
variables, teniendo en cuenta que la división puede
producir una excepción */

        int suma = primerNumero + segundoNumero;
        System.out.println("\n"+a +"+"+b+"= "+suma);
        int resta = a-b;
        System.out.println(primerNumero +"-"+ segundoNumero +"=
"+resta);
        int multip= primerNumero * segundoNumero;
        System.out.println(primerNumero +"*"+ segundoNumero +"=
"+multip);
        int div;

        try {

            div = primerNumero / segundoNumero;
            System.out.println(primerNumero +"/"+segundoNumero+"=
"+div);
```

```
}  
  
    catch (Exception err) {  
  
        System.out.println("No se puede dividir entre 0");  
  
    }  
  
    sc.close();  
  
}
```

Ejecución del programa

The image displays three screenshots of a Java application's console output, illustrating different execution scenarios. Green arrows point from the title 'Ejecución del programa' to each screenshot.

Left Screenshot: Shows the application running with inputs 10 and 5. The output displays the results of four arithmetic operations: 10+5= 15, 10-5= 5, 10*5= 50, and 10/5= 2.

Top Right Screenshot: Shows the application running with inputs 3 and 0. The output displays the results of three arithmetic operations: 3+0= 3, 3-0= 3, and 3*0= 0, followed by the message "No se puede dividir entre 0".

Bottom Right Screenshot: Shows the application running with inputs 7 and 'hola'. The output displays the results of four arithmetic operations: 7+1= 8, 7-1= 6, 7*1= 7, and 7/1= 7, followed by the message "Tipo de dato incorrecto. Se utilizará el valor 1".

Estructuras de control.

Modifican el flujo secuencial del programa.

- De selección.
- Iterativas (con control previo, control posterior o por subrango).

Ejercicio 5

Escribir un programa que solicite un número, e indique si el mismo es negativo y par o no lo es.

Ejercicio 7

Dado un monto de compra se aplica un descuento:

Si el monto es menor a 100 no hay descuento.

15% si el monto es mayor a 200.

10% de descuento en cualquier otro caso.

```
if (condición) {  
    //Instrucciones a ejecutar.  
}  
  
else if (condición) {  
    //Instrucciones a ejecutar.  
}  
  
else {  
    /*Instrucciones a ejecutar si  
    ninguna condición se cumple  
    */  
}
```

if

Ejercicio 6

Escribir un programa que solicite una calificación final y devuelva un juicio..

1<= nota<= 5 insuficiente

5 < nota <= 7 aceptable

8<= nota <= 10 muy buen trabajo

Ejercicio 5

```
import java.util.*;

public class Negativo_Par {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner (System.in);
        int numero;
        System.out.println("Ingrese un numero: ");

        try { // controlo posibles excepciones
            numero=sc.nextInt();
        }
        catch (Exception err) {
            System.out.println("No es un valor válido."
                + "\nSe ingresará 0.\n");
            numero=0;
        }

        if (numero<0 && numero%2==0) { // ambas condiciones deben cumplirse a la vez
            System.out.println("Es negativo y par");
        }
        else {
            System.out.println("No es negativo y par");
        }

        sc.close();
    }
}
```

```
package ejemplos;
import java.util.*;

public class Calificacion {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner (System.in);
        sc.useLocale(Locale.US); //utilizo el punto como separador decimal
        System.out.println("Ingrese la calificación final :");
        // Podría utilizar try/catch
        float calif=sc.nextFloat();

        if (calif <= 5.75 && calif >= 1) {
            System.out.println("Insuficiente.");
        }
        else if (calif > 5.75 && calif <= 7.9) {
            System.out.println("Aceptable.");
        }
        else if (calif >= 8 && calif <= 10) {
            System.out.println("Muy buen trabajo");
        }
        else {
            System.out.println("No es una calificación válida");
        }

        sc.close();
    }
}
```

Ejercicio 6

Ejercicio 7

```
1 package ejemplos;
2 import java.util.*;
3
4 public class Descuento {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         Scanner sc = new Scanner (System.in);
9         float descuento;
10        float total;
11        System.out.println("Ingrese el valor de la compra :");
12        float monto=sc.nextFloat();
13
14        if (monto < 100) {
15            System.out.println("No hay descuento.\nUsted debe abonar $" + monto);
16        }
17        else if (monto > 200) {
18            descuento=(15 * monto) / 100; // monto*0.15 es equivalente
19            System.out.println("Se aplicara un 15% de descuento: (" + descuento + ")");
20            total=monto-descuento;
21            System.out.println("Usted abona: $" + total);
22        }
23        else { // si el monto no es menor que 100 ni mayor que 200 se aplica un 10%
24            descuento=monto*0.10f;
25            System.out.println("Se aplicara un 10% de descuento: (" + descuento + ")");
26            total=monto-descuento;
27            System.out.println("Usted abona: $" + total);
28        }
29        sc.close();
30    }
31 }
32
```

Ejecución del programa.

Ingrese el valor de la compra :
99
No hay descuento.
Usted debe abonar \$99.0

0%

Ingrese el valor de la compra :
125
Se aplicara un 10% de descuento: (\$12.5)
Usted abona: \$112.5

10%

15%

Ingrese el valor de la compra :
201
Se aplicara un 15% de descuento: (\$30.15)
Usted abona: \$170.85

switch

if puede evaluar expresiones booleanas.

switch evalúa un valor contra una serie de valores fijos. Si evalúo contra un gran número de valores, switch es más eficiente.

```
switch (variable) {
```

```
  case valor1:
```

```
    // secuencia de instrucciones.
```

```
    break;
```

```
  case valor2:
```

```
    // secuencia de instrucciones.
```

```
    break;
```

```
  case valorN:
```

```
    // secuencia de instrucciones.
```

```
    break;
```

```
  default:
```

```
    // instrucciones por defecto
```

```
}
```

Ejercicio 8

Escribir un programa que pida al usuario un número y devuelva el día de la semana correspondiente.

Si el número no está en el rango 1-7 devuelve un error.

Ejercicio 8

```
package switch_ejemplo;
import java.util.*;

public class Switch {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner (System.in);

        System.out.println("Ingrese un número(1-7): ");
        int num=sc.nextInt();

        switch (num) {

            case 1: System.out.println("Lunes");
                    break; // si es una sola instrucción no necesito llaves.
            case 2: System.out.println("Martes");
                    break;
            case 3: System.out.println("Miércoles");
                    break;
            case 4: System.out.println("Jueves");
                    break;
            case 5: System.out.println("Viernes");
                    break;
            case 6: System.out.println("Sábado");
                    break;
            case 7: System.out.println("Domingo");
                    break;
            default: System.out.println("No es un número válido");

        }

        sc.close();
    }
}
```

Si no hay break; se continúa ejecutando el siguiente bloque case.

```
case 1: { // Si es una sola instruccion no necesito llaves.
        System.out.println("Lunes");
        //break;
    }
case 2: System.out.println("Martes");
        break;
case 3: System.out.println("Miércoles");
        break;
case 4: System.out.println("Jueves");
        break;
case 5: System.out.println("Viernes");
```

Console Problems Javadoc Declaration

```
<terminated> Switch [Java Application] C:\Users\santi\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre...
Ingrese un número(1-7):
1
Lunes
Martes
```

Ejecuta un bloque de código en bucle, mientras se cumpla una condición.

Se conoce la cantidad de iteraciones de antemano.

for

```
1 package ejercicios;
2
3 public class EjemploFor {
4
5     public static void main(String[] args) {
6
7         for (int i = 0; i < 10 ; i++) { //defino la variable y la utilizo para
8             //controlar las iteraciones.
9             System.out.println(i); //
10        }
11
12        System.out.println("\nYa estoy fuera del bucle");
13    }
14 }
15
```

Console × Declaration Javadoc Problems
<terminated> EjemploFor [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18f 2025 13:55:46 – 13:55:47) [pid: 17524]

0
1
2
3
4
5
6
7
8
9

Ya estoy fuera del bucle

Ejercicio 9

Mostrar por pantalla todos los números **múltiplos de 3**, entre 1 y 100.

Ejercicio 9

```
1 package ejercicios;
2
3 public class Multiplos3 {
4
5     public static void main(String[] args) {
6         //
7         for (int i = 1; i <= 100; i++) {
8
9             if(i % 3 == 0) {
10                 System.out.println(i);
11             }
12         }
13     }
14 }
15
16
17
```

Console X Declaration Javadoc Problems

<terminated> Multiplos3 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18 feb. 2025 14:03:19 – 14:03:20)

3
6
9
12
15
18
21
24
27
30
33
36
39
42

...

Ejercicio 10

Realizar un programa que pida una **cantidad de números** (determinada por el usuario).

Al finalizar muestra el **promedio** de todos los valores y cuantos hay **mayores/menores que 10**.

Ejercicio 10

```
package ejercicios;
import java.util.*;

public class SolicitarValores {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("¿Cuántos valores desea ingresar?");
        int cantidad = sc.nextInt();
        float valor, promedio, total = 0f;
        int mayor = 0, menor = 0;

        for(int i = 0; i < cantidad; i++) {
            System.out.println("Ingrese un valor: ");
            valor = sc.nextFloat();
            total += valor;

            if(valor < 10)
                menor++;
            if(valor > 10)
                mayor++;
        }

        promedio = total / cantidad;
        System.out.println("El promedio es: "+promedio);
        System.out.println("Hay "+mayor+" mayores a 10");
        System.out.println("Hay "+menor+" menores a 10");
        sc.close();
    }
}
```

Ejecución del
programa.

```
¿Cuántos valores desea ingresar?
3
Ingrese un valor:
17
Ingrese un valor:
3
Ingrese un valor:
41
El promedio es: 20.333334
Hay 2 mayores a 10
Hay 1 menores a 10
```

```
Hay 1 mayores a 10
Hay 3 menores a 10
```

while

while(condición de ingreso al bucle)

{

....

....

Bloque de Instrucciones...

....

....

}

do-while

Ejercicio 11

Pedir números por teclado hasta ingresar un 0.

Al terminar mostrar la suma, el mayor y el menor de los números ingresados.

do

//al menos ingresa una vez al bucle.

{

....

....

Bloque de Instrucciones...

....

....

}

while(condición de ingreso al bucle);

```

import java.util.*;

public class IngresarHasta {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Ingrese un valor: ");
        int valor = sc.nextInt();

        // Inicializar los valores con el primer número ingresado
        int mayor = valor, menor = valor, suma = 0;

        while (valor != 0) {
            suma += valor; // Sumar solo si no es 0

            // Verificar si es mayor o menor
            if (valor < menor) {
                menor = valor;
            }
            if (valor > mayor) {
                mayor = valor;
            }

            System.out.println("Ingrese un valor: ");
            valor = sc.nextInt(); // Leer el próximo número antes de volver a iterar
        }

        System.out.println("La suma es: " + suma);
        System.out.println("El mayor es: " + mayor);
        System.out.println("El menor es: " + menor);

        sc.close();
    }
}

```

Ejercicio 11

```

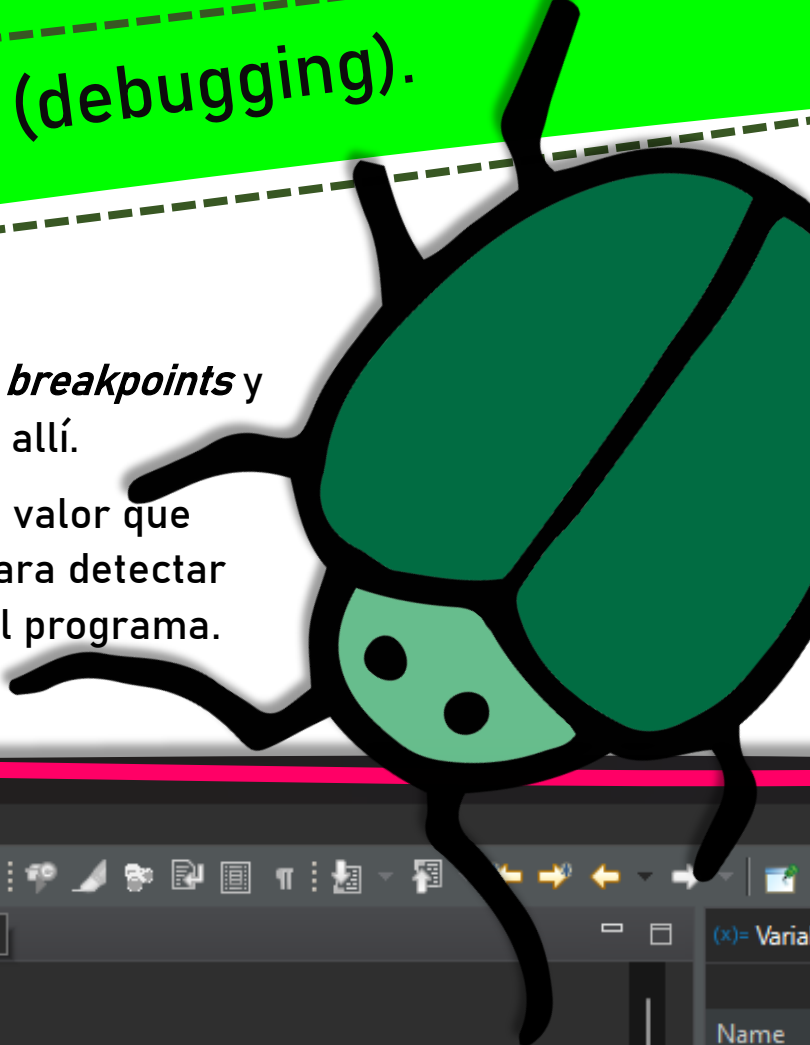
Ingrese un valor:
12
Ingrese un valor:
3
Ingrese un valor:
10
Ingrese un valor:
0
La suma es: 25
El mayor es: 12
El menor es: 3

```


Depuración del código (debugging).

La opción **debug** permite establecer **breakpoints** y ejecutar el código línea a línea desde allí.

Se puede visualizar en tiempo real el valor que adquieren variables y expresiones, para detectar así un funcionamiento no deseado del programa.



Java - Eclipse IDE

File Edit Project Run Window Help

IngresarHasta.java

Debug IngresarHasta (already running)

```
1 import java.util.*;
2
3 public class IngresarHasta {
4
5
6     public static void main(String[] args) {
7
8         Scanner sc = new Scanner (System.in);
9         System.out.println("Ingrese un valor: ");
10        int valor = sc.nextInt();
11        int mayor=valor,menor=valor,suma=valor;
12
13        while (valor!=0) {
14
15
16
17            if (valor<menor) {
18                menor=valor;
19            }
20            if (valor>mayor) {
```

Console Problems Debug Shell

IngresarHasta [Java Application] C:\Users\santi\p2\poc
Ingrese un valor:
4

Variables Breakpoints Expressions

Name	Value
no method return value	
args	String[0] (id=20)
sc	Scanner (id=21)
valor	4
mayor	4
menor	4
suma	4

Cadenas de caracteres//String


La **clase String** se utiliza para manipular cadenas. Tiene varios *métodos* que pueden utilizarse para todos los objetos de este tipo.

```
package cadenas;
import java.util.*;


public class cadenas {

    public static void main(String[] args) {
        // Defino cadenas
        String cadena1 = "Soy una cadena";
        String cadena2 = new String ("Otra cadena de caracteres");
        //Concateno cadenas
        String cadena3=cadena1+" "+cadena2;
        System.out.println(cadena3);
        //Longitud de una cadena
        System.out.println(cadena3.length());
        //Comparar con método equals
        //compareTo compara alfabéticamente
        Scanner sc =new Scanner (System.in);
        System.out.println("Ingrese un texto");
        String texto1=sc.nextLine();
        System.out.println("Ingrese un texto");
        String texto2=sc.nextLine();

        if (texto1.equals(texto2)) {
            System.out.println("Son iguales.");
        }
        else {
            System.out.println("Son diferentes");
        }
        sc.close();
    }
}
```



```
Soy una cadena Otra cadena de caracteres
14
Ingresa un texto
hola
Ingresa un texto
hola
Son iguales.
```



```
Soy una cadena Otra cadena de caracteres
14
Ingresa un texto
hola
Ingresa un texto
chau
Son diferentes
```

```
String cadena1 = "Soy una cadena";
```

```
//Obtener caracter de una posición (comenzando por 0)


char c = cadena1.charAt(1); //caracter en la posición 1
System.out.println(c);

//Primera posición en la que se encuentra un caracter
//lastIndexOf muestra la última coincidencia

System.out.println(cadena1.indexOf("a"));
System.out.println(cadena1.lastIndexOf("a"));

//Convertir a mayúsculas/minúsculas

System.out.println(cadena1.toLowerCase());
System.out.println(cadena1.toUpperCase());
```



```
0
6
13
soy una cadena
SOY UNA CADENA
```

```

1 package cadenas;
2
3
4 public class Recorrer {
5
6     public static void main(String[] args) {
7
8
9
10        String miCadena = new String ("Santiago");
11
12        //recorro cadena con un bucle for
13
14
15        for (int i=0;i < miCadena.length();i++) { //el método length obtiene el largo
16
17            System.out.println(miCadena.charAt(i)); //el método charAt obtiene el caracter en una posición
18            //System.out.print(miCadena.charAt(i));
19            // print no introduce un salto de línea luego de imprimir
20
21        }
22
23
24
25
26
27    }
28
29 }
30

```

Recorrer un string

Problems Javadoc Declaration Console X

<terminated> Recorrer [Java Application] C:\Users\santi\p2\workspace\ejercicios\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (5 feb. 2024 13:22:32 - 13:22:32) [pid: 19028]

S
a
n
t
i
a
g
o

Ejercicio 12

Pedir una cadena de caracteres por teclado y contar cuantos espacios en blanco se ingresaron.

Ejercicio 13

Validar una contraseña ingresada:

- Debe contener entre 8 y 15 caracteres.
- Algunos de los siguientes caracteres: " * " " - " " _ "

Ejercicio 12

```
1 package cadenas;
2 import java.util.*;
3
4 public class PedirCadena {
5
6     public static void main(String[] args) {
7
8         Scanner sc = new Scanner (System.in);
9         System.out.println("Ingrese una cadena de caracteres: ");
10        String cadena=sc.nextLine();
11        int contador =0;
12
13        for (int i=0; i< cadena.length();i++) { //la primer posición es la 0
14            if(cadena.charAt(i)==' ') //comparo caracteres con comillas simples y ==
15                contador ++;
16        }
17
18        System.out.println("Existen "+contador+" espacios en blanco");
19        sc.close();
20
21
22
23
```

Console Problems Debug Shell

<terminated> PedirCadena [Java Application] C:\Users\santi\.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (5 feb. 2023)

Ingrese una cadena de caracteres:
Mi nombre es Santiago
Existen 3 espacios en blanco

```
package cadenas;
import java.util.*;

public class ValidarPasswd {

    public static void main(String[] args) {

        Scanner sc = new Scanner (System.in);
        System.out.println("Ingrese un password válido: ");
        String passwd = sc.nextLine();

        if (passwd.length() >= 8 && passwd.length() <= 15) {
            boolean existe = false;

            for(int i = 0; i < passwd.length(); i++) {
                if (passwd.charAt(i) == '-' || passwd.charAt(i) == '_' || passwd.charAt(i) == '*') {
                    existe = true;
                }
            }

            if (existe) {
                System.out.println("Contraseña válida.");
            }
            else {
                System.out.println("Faltan caracteres especiales.");
            }
        }
        else {
            System.out.println("Longitud inválida.");
        }

        sc.close();
    }
}
```

Ejercicio 13

Ejecución del programa.

```
Ingrese un password válido:  
_passwd  
Longitud inválida.
```

```
Ingrese un password válido:  
_password  
Contraseña válida.
```

```
Ingrese un password válido:  
password  
Faltan caracteres especiales.
```

Estructuras de datos//Array

Permiten almacenar más de un dato del mismo tipo.
Una vez creado **el tamaño es fijo.**

```
public class Arreglos {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int [] arreglo = new int [4]; //Defino un arreglo de 4 enteros  
        String [] palabras = new String [3]; // Arreglo de cadenas  
        /*Los arreglos de enteros por defecto  
        se inician con 0 y los de string con null*/  
    }  
}
```

```
//Defino arreglo e inicilizo sus valores
double [] decimales = {6.4,2.4,-10.7};
//Accedo a una posicion del arreglo (comenzando por 0).
System.out.println(decimales[1]);
//Modifico el valor de una posición
decimales[1]=55.2;
System.out.println(decimales[1]);
System.out.println(decimales.length);
//Recorrer arreglo con bucle for
for(int i=0; i<palabras.length;i++) { //Cargo valores
    System.out.println("Ingrese una palabra :");
    palabras[i]=sc.nextLine();
}
// Muestro valores por pantalla
System.out.println("Las palabras ingresadas son: ");

for(int i=0; i<palabras.length;i++) {
    System.out.print(palabras[i]+" ");
}
sc.close();
```

Ejecución del programa.

```
2.4
55.2
3
Ingrese una palabra :
Santiago
Ingrese una palabra :
curso
Ingrese una palabra :
JAVA
Las palabras ingresadas son:
Santiago curso JAVA
```


Ejercicio 14

Desarrollar un programa que inicialice un **arreglo de 10 enteros aleatorios**. Posteriormente imprime sus elementos y la suma de ellos.

Ejercicio 15

Solicitar la **temperatura de cada día de la semana**.

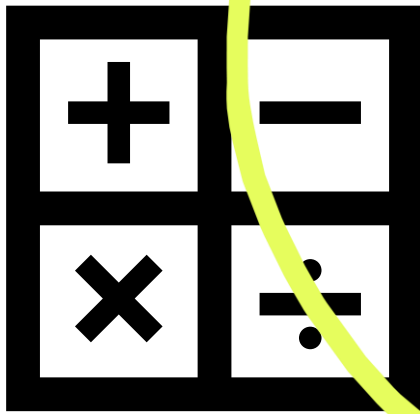
Realizar el promedio e indicar si en algún momento superó los 30°

Ejercicio 14

```
public class ArrayAleatorio {  
  
    public static void main(String[] args) {  
        // valor aleatorio entre 1 y 10  
        //System.out.println((int)(Math.random()*10+1));  
        int [] arreglo_enteros = new int [10];  
        int suma = 0;  
        int valor = 0;  
  
        for(int i = 0; i < arreglo_enteros.length; i++) {  
            valor = (int)(Math.random()*10+1);  
            arreglo_enteros[i] = valor;  
            suma += valor;  
        }  
  
        System.out.println("Los valores son:");  
  
        for(int i = 0; i < arreglo_enteros.length; i++) {  
            System.out.println(arreglo_enteros[i]);  
        }  
  
        System.out.println("La suma es: " + suma);  
    }  
}
```

La Clase **Math** está integrada en Java. Tiene sus propios métodos y forma parte del paquete java. lang

El método **random** genera un número de tipo **double** dentro del intervalo **[0,1)**



```
Los valores son:
8
1
5
2
1
1
7
10
2
8
La suma es: 45
```

```
package ejemplos;
import java.util.*;
public class Temperatura {

    public static void main(String[] args) {

        String [] semana = {"Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"};
        float [] temps = new float [7];
        Scanner sc = new Scanner (System.in);
        float suma = 0f;
        boolean mas30 = false;

        for (int i = 0; i < semana.length; i++) {
            System.out.println("Ingrese la temperatura del "+semana[i]+":");
            temps[i] = sc.nextFloat();
            suma += temps[i];

            if (temps[i] >= 30) {
                mas30=true;
            }
        }
        System.out.println("\nLa temperatura promedio fue: " +suma / 7+"°");

        if (mas30) {
            System.out.println("Se alcanzaron los 30°");
        }

        sc.close();
    }
}
```

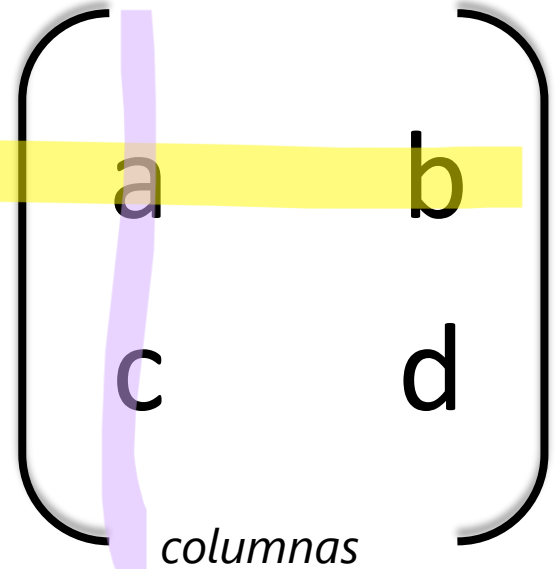
Ejercicio 15

```
Ingrese la temperatura del Lunes:
23,4
Ingrese la temperatura del Martes:
25,6
Ingrese la temperatura del Miércoles:
30,1
Ingrese la temperatura del Jueves:
26
Ingrese la temperatura del Viernes:
24,7
Ingrese la temperatura del Sábado:
26,9
Ingrese la temperatura del Domingo:
24

La temperatura promedio fue: 25.814285°
Se alcanzaron los 30°
```

Arreglos bidimensionales (MATRICES).

filas



Utiliza 2 índices para acceder a los elementos **(se accede al elemento mediante número de fila y columna).**

```
1
2 package ArrayBidimensional;
3
4 public class matrices {
5
6     public static void main(String[] args) {
7
8         String[] arreglo = {"ab","cd","ef"}; //arreglo.
9         //arreglo bidimensional o matriz
10        String[][] matriz = {"abc","def"}, {"ghi","jkl"}, {"mno","pqr"};
11        // Es una matriz de 3 filas x 2 columnas.
12        // Es un arreglo que contiene 3 arreglos de 2 elementos c/u.
13        int[][] matriz2 = new int[4][3]; // Defino matriz de 4 filas x 2 columnas
14        System.out.println(matriz[0][1]); //Elemento de la fila 0 columna 1
15        System.out.println(matriz[2][0]); //Elemento de la fila 2 columna 0
16    }
17 }
18
19
```

Console Declaration Javadoc Problems
<terminated> matrices [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18 feb. 2025 14:38:23 - 14:38:23) [pid: 5732]

def
mno

```

2 package ArrayBidimensional;
3
4 public class Recorrer {
5
6     public static void main(String[] args) {
7
8         String[][] matriz = {"Esto", "es"}, {"una", "matriz"}, {"de", "cadenas."};
9
10        for(int i=0; i<3; i++) { // Itera por filas
11            for (int j=0; j<2; j++) { //Itera por columnas
12                System.out.print(matriz[i][j]+" ");
13            }
14
15            System.out.println();
16        }
17    }
18 }

```

Recorrer una matriz.

Console X Declaration
<terminated> Recorrer [Java Applic...

Esto es
una matriz
de cadenas.

Ejercicio 16

Utilizando matrices, realizar un programa que muestre las **calificaciones asociadas a un grupo de alumnos** y calcule el promedio final.

Datos de la matriz

- Nombre del alumno.
- Nota de clase.
- Nota de proyecto.

Nota final (60% nota de proyecto 40% nota de clase)

```
package ArrayBidimensional;

public class Alumnos {

    public static void main(String[] args) {

        String [][] alumnos = {{ "Pepe", "8", "6"}, {"María", "7", "10"}, {"Santiago", "5", "11"} };
        /* se cargan 4 alumnos con su nombre, nota de clase y nota de proyecto (todo de tipo String)
        */
        System.out.println("Información de los alumnos:\n ");

        for(int i = 0; i < alumnos.length; i++) {
            for(int j = 0; j < alumnos[i].length; j++) {
                System.out.print(alumnos[i][j] + " ");
            }
            //nota final es 60% proyecto y 40% clase
            double notafinal = (Double.parseDouble(alumnos[i][1]) * 0.4) + (Double.parseDouble(alumnos[i][2]) * 0.6);
            System.out.println("\n | Nota final: " + notafinal);
        }
    }
}
```

Ejercicio 16

Información de los alumnos:

Pepe 8 6 | Nota final: 6.8

María 7 10 | Nota final: 8.8

Santiago 5 11 | Nota final: 8.6

Utilizo la **clase Double** (cada tipo numérico tiene una clase en JAVA)

//método **parseDouble** para convertir texto a tipo double y operar.