ELEMENTOS DEL LENGUAJE

Contenido

[PAQUETE BASE 3](#_Toc415389755)

[PAQUETES 3](#_Toc415389756)

[IDENTIFICADOR 4](#_Toc415389757)

[COMENTARIOS 5](#_Toc415389758)

[Línea 5](#_Toc415389759)

[Varias líneas 5](#_Toc415389760)

[Documentación 5](#_Toc415389761)

[TIPOS DE DATOS 6](#_Toc415389762)

[Tipos de Datos Primitivos 6](#_Toc415389763)

[Tipos Referenciados 6](#_Toc415389764)

[OPERADORES 6](#_Toc415389765)

[Operadores de Asignación 6](#_Toc415389766)

[Operadores de Comparación 7](#_Toc415389767)

[Operadores Lógicos 8](#_Toc415389768)

[Operador Ternario (?:) 8](#_Toc415389769)

[ESTRUCTURAS DE CONTROL 9](#_Toc415389770)

[IF 9](#_Toc415389771)

[IF-ELSE 9](#_Toc415389772)

[Ejercicio 01 10](#_Toc415389773)

# PAQUETE BASE

Todo proyecto de tener un paquete base:

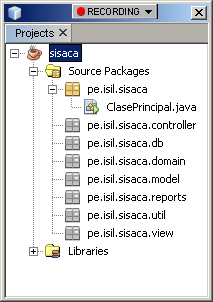
dominio\_a\_la\_inversa.aplicacion

**Ejemplo 1:**

Empresa ISIL, aplicación Sistema Academico, paquete base podría ser:

* **pe.isil.sisaca**
* pe.isil.academico
* pe.isil.sa

La estructura de paquetes sería:

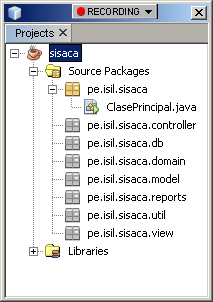


# PAQUETES

Se utilizan para organizar las clases de un proyecto.

Las clases se agrupan por la funcionalidad que realizan.

Por ejemplo, si utilizamos el patrón Model-View-Controller (MVC), se tendría una paquete para las clases de la Vista, otro paquete para las clases del Modelo y finalmente otro paquete para las clases del controlador.



# IDENTIFICADOR

Nombre que se le da a una variable, método, clase, interface, constante.

Existen identificadores predefinidos, llamados palabras reservadas, ejemplo: class, public, private, protected, if, while, etc.

|  |  |
| --- | --- |
| **Tipo de Identificador** | **Ejemplo** |
| Variable | * contador * tipoDocumento * contador003 |
| Método | * calcularInteres * procesar * traerCategorias |
| Clase | * Factura * VentaController * VentaModelImpl |
| Inteface | * IVentaModel * VentaModelEspec |
| Constabte | * PI * IGV * PORC\_GANANCIA |

# COMENTARIOS

Tres tipo de comentarios: en línea, varias líneas y de documentación.

## Línea

// Esto es un comentario

cont++; // Incrementa el contador

## Varias líneas

/\* Esto es un comentario

De varias líneas \*/

## Documentación

/\*\*

\* Método para encontrar el promedio de 3 notas.

\* Las notas deben estar en el rango de 0 a 20.

\* @param n1 Nota de practicas

\* @param n2 Examen parcial

\* @param n3 Examen final

\* @return

\*/

public int promedio(int n1, int n2, int n3){

return (n1 + n2 + n3) / 3;

}

# TIPOS DE DATOS

## Tipos de Datos Primitivos

int, byte, short, double, float, boolen, char.

## Tipos Referenciados

Toda clase o interface es un tipo referenciado.

Las variables deben apuntar a un objeto del tipo.

# OPERADORES

## Operadores de Asignación

**Operador =**

public static void main(String[] args) {

int n;

n = 5;

System.out.println("n = " + n);

n = n \* n - n;

System.out.println("n = " + n);

n = Math.multiplyExact(n, n);

System.out.println("n = " + n);

}

**Operadores +=, -=, \*=, /=**

n += 5; // n = n + 5

System.out.println("n = " + n);

n -= 50; // n = n - 50

System.out.println("n = " + n);

**Operadores unarios: ++ y –**

n--; // Disminuye en uno el valor de n

--n; // Disminuye en uno el valor de n, mayor prioridad

System.out.println("n = " + n--);

System.out.println("n = " + --n);

## Operadores de Comparación

Se utilizan para crear una condición simple.

<, <=, >, >=, ¡=, ==

public static void main(String[] args) {

int a = 10;

int b = 15;

boolean mayor = (a>b);

boolean mayorIgual = (a>=b);

boolean menor = (a<b);

boolean menorIgual = (a<=b);

boolean igual = (a==b);

boolean diferente = (a!=b);

System.out.println("Mayor = " + mayor);

System.out.println("Mayor Igual = " + mayorIgual);

System.out.println("Menor = " + menor);

System.out.println("Manor Igual = " + menorIgual);

System.out.println("Igual = " + igual);

System.out.println("Diferente = " + diferente);

}

## Operadores Lógicos

Para crear condiciones complejas en base a condiciones simples.

!, &, &&, |, ||

public class OperadoresLogicos {

public static void main(String[] args) {

Integer a = 10;

Integer b = null;

boolean c = (a != null & b != null) && (a < b);

System.out.println("c = " + c);

}

}

## Operador Ternario (?:)

**(Condicion)?ResultadoTrue:ResultadoFalse**

# ESTRUCTURAS DE CONTROL

## IF

**Sintaxis 1:**

if(condición) sentencia;

Ejemplo

if(nota>=14)

System.out.println(“Pasaste de año”);

**Sintaxis 2:**

if(condición) {

sentencia1;

sentencia2;

}

## IF-ELSE

**Sintaxis:**

if(condición){

sentencia1;

sentencia2;

} else {

sentencia3;

sentencia3;

}

# Ejercicio 01

Desarrollar un programa que permita calcular el sueldo de un trabajador y el impuesto a la renta:

Datos:

* Pago por hora.
* Horas por día.
* Días trabajados

Calcular:

* Ingresos
* Renta (10% si los ingresos son mayores a 1500).
* Neto

import javax.swing.JOptionPane;

public class Ejercicio01 {

public static void main(String[] args) {

//datos

Double pagoPorHora =

Double.parseDouble(llenar("ingrese el pago por hora"));

Double horasPorDia =

Double.parseDouble(llenar("ingrese las horas por dia"));

Double diasTrabajados =

Double.parseDouble(llenar("ingrese dias trabajados"));

//variables

Double ingreso = 0.0, renta = 0.0, neto = 0.0;

//procesos

ingreso = pagoPorHora \* horasPorDia \* diasTrabajados;

/\*if (ingreso >= 1500) {

renta = ingreso \* 0.1;

}\*/

renta = (ingreso>=1500.0)?ingreso\*0.10:0.0;

neto = ingreso - renta;

System.out.println("el ingreso es " + ingreso);

System.out.println("la renta es " + renta);

System.out.println("neto " + neto);

}

private static String llenar(String dato) {

String text = JOptionPane.showInputDialog(dato);

return text;

}

}

# Ejercicio 2

Encontrar los días que tiene un mes.

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Alegría

\*/

public class Ejercicio2 {

public static void main(String[] args) {

//datos

String mes = llenar("ingresar el mes");

int aaa = Integer.parseInt(llenar("ingresar el año"));

//variables

int diasqueTieneunMes = 0;

int mesdeFebrero;

//proceso

diasqueTieneunMes = (mes.equals("enero")) ? 31 : diasqueTieneunMes;

diasqueTieneunMes = (aaa % 4 == 0 || mes.equals("febrero")) ? 29 : 28;

diasqueTieneunMes = (mes.equals("marzo")) ? 31 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("abril")) ? 30 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("mayo")) ? 31 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("junio")) ? 30 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("julio")) ? 31 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("agosto")) ? 31 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("setiembre")) ? 30 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("octubre")) ? 31 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("noviembre")) ? 30 : diasqueTieneunMes;

diasqueTieneunMes = (mes.equals("diciembre")) ? 31 : diasqueTieneunMes;

System.out.println(" el mes de " + mes + "tiene " + diasqueTieneunMes + "dias");

}

private static String llenar(String dato) {

String tex = JOptionPane.showInputDialog(dato);

return tex;

}

}

# EJERCICIO 03

Hacer un programa que permita convertir:

* Soles a Dolares
* Dolares a Soles
* Soles a Euros
* Euros a Soles
* Dolares a Euros
* Euros a Dolares

# ESTRUCTURAS REPETITIVAS

## Contador

Variable que se utiliza para contar las veces que sucede un evento.

## Acumulador

Variable que acumula los valores de un cálculo, generalmente dentro de un bucle.

## while

while(condicion){

-------

-------

}

## do-while

do{

---------

---------

} while(condicion);

## for

for(int cont = 0; cont <= n; cont++){

-------

-------

}

## break

Interrumpe la ejecución de un bucle.

## continue

Regresa al inicio del bucle.

# EJERCICIO 4

Desarrollar un programa que permita mostrar los primeros "N" números múltiplos de 3 y 5.

# EJERCICIO 5

Desarrollar un programa que permita mostrar los primeros "N" números, que la suma de sus dígitos sea múltiplos de 3 y 5.

69

78

168

# EJERCICIO 6

Desarrollar un programa que permita descomponer una cantidad monetaria (Nuevos Soles) y billetes y monedas extactas:

Por ejemplo:

Importe: 587

Billetes de 200: 2

Billetes de 100: 1

Billetes de 50: 1

Billetes de 20: 1

Billetes de 10: 1

Monedas de 5: 1

Monedas de 2: 1

Monedas de 1: 0

# EJERCICIO 7

Convertir un número a letras.

|  |  |
| --- | --- |
| Número | En Letras |
| 1 | Uno |
| 10 | Diez |
| 11 | Once |
| 16 | Dieciséis |
| 40 | Cuarenta y dos |
| 100 | Cien |