**DOCUMENTACIÓN**

**1.- INTRODUCCIÓN A JAVA**

**TIPOS DE DATOS**

**Números enteros:**

byte: Representa enteros de 8 bits con signo. Rango: -128 a 127.

short: Representa enteros de 16 bits con signo. Rango: -32,768 a 32,767.

int: Representa enteros de 32 bits con signo. Rango: -2,147,483,648 a 2,147,483,647.

long: Representa enteros de 64 bits con signo. Rango: -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807.

**Números de punto flotante:**

float: Representa números de punto flotante de 32 bits. Rango: aproximadamente ±1.4e-45 a ±3.4e+38. El tipo float tiene una precisión limitada en comparación con el tipo double.

double: Representa números de punto flotante de 64 bits. Rango: aproximadamente ±4.9e-324 a ±1.7e+308. El tipo double proporciona una mayor precisión que el tipo float.

**Caracteres:**

char: Representa un solo carácter Unicode de 16 bits. Puede almacenar caracteres individuales y valores numéricos Unicode. Rango: '\u0000' (0) a '\uffff' (65,535).

**Booleanos:**

boolean: Representa un valor booleano que puede ser true o false. No se puede especificar el rango de almacenamiento ya que solo existen dos valores posibles.

**MÉTODOS**

1. Métodos sin retorno (void): Estos métodos no devuelven ningún valor. Son utilizados cuando solo se necesita ejecutar un conjunto de instrucciones sin retornar un resultado específico.
2. Métodos con retorno: Estos métodos devuelven un valor de un tipo específico. El tipo de retorno se especifica al declarar el método, y el resultado se devuelve mediante la palabra clave return.
3. Métodos con retorno de tipo primitivo: Pueden devolver valores de tipos primitivos, como int, double, boolean, etc.
4. Métodos con retorno de tipo objeto: Pueden devolver objetos de una clase específica. El tipo de retorno debe ser el nombre de la clase o una superclase del objeto devuelto.

**2.- ARREGLOS**

En el lenguaje de programación Java, un arreglo (también conocido como array) es una estructura de datos que almacena una colección de elementos del mismo tipo. Los arreglos son útiles cuando necesitas trabajar con conjuntos de datos relacionados y acceder a ellos de forma secuencial.

Para declarar un arreglo en Java, debes especificar el tipo de elementos que contendrá y el nombre del arreglo. Por ejemplo, si deseas crear un arreglo de enteros, puedes hacerlo de la siguiente manera:

int[] numeros;

Aquí, hemos declarado un arreglo llamado "numeros" que almacenará valores enteros.

Una vez declarado el arreglo, necesitas asignarle una cantidad específica de elementos. Puedes hacerlo de dos maneras:

1. Especificando la longitud del arreglo al momento de su creación:

int[] numeros = new int[5];

En este caso, hemos creado un arreglo llamado "numeros" con una longitud de 5 elementos. Todos los elementos se inicializarán con el valor predeterminado del tipo de datos (en este caso, 0 para enteros).

1. Inicializando el arreglo con valores específicos:

int[] numeros = {1, 2, 3, 4, 5};

Aquí, hemos creado un arreglo llamado "numeros" con los valores 1, 2, 3, 4 y 5. La longitud del arreglo se determina automáticamente en función del número de elementos proporcionados.

Una vez que tienes un arreglo creado, puedes acceder a sus elementos utilizando un índice. Los índices en Java comienzan desde 0, lo que significa que el primer elemento tiene un índice de 0, el segundo elemento tiene un índice de 1, y así sucesivamente. Por ejemplo:

int primerNumero = numeros[0]; // Accede al primer elemento (1) int segundoNumero = numeros[1]; // Accede al segundo elemento (2)

También puedes modificar el valor de un elemento utilizando su índice:

numeros[2] = 10; // Modifica el tercer elemento y le asigna el valor 10

Es importante tener en cuenta que los arreglos en Java tienen una longitud fija una vez que se crean. Si intentas acceder a un índice fuera del rango válido o asignar un valor a un índice inválido, se producirá una excepción en tiempo de ejecución.

Esto ha sido solo una introducción básica a los arreglos en Java. Los arreglos en Java tienen muchas más características y funcionalidades, como métodos para recorrerlos, ordenarlos o buscar elementos específicos. Con una comprensión sólida de los conceptos básicos, puedes explorar estas características y utilizar los arreglos de manera efectiva en tus programas Java.