



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

24/01/2023

Hora de inicio:

09:00

Introducción a la Programación y Computación 1 [Sección B]

Diego Fernando Cortez Lopez

Agenda

- Lectura de Enunciado
- Clase 2
- Ejemplo Práctico

Fundamentos de la Programación y Java - Parte 2

Clase 2

Operadores en JAVA

Operadores Aritméticos

Operador	Descripción
+	Operador de Suma
-	Operador de Resta
*	Operador de Multiplicación
/	Operador de División
%	Operador de Resto



Operadores Unarios

Operador	Descripción
+	Operador unario Suma
-	Operador unario Resta
++	Operador de Incremento
--	Operador de Decremento
!	Operador de Complemento Lógico



Operadores Relacionales

Operador	Descripción
==	Igual a
!=	No igual a
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que

Operadores Condicionales

Operador	Descripción
&&	Operador Condicional AND
	Operador Condicional OR
?:	Operador Ternario
instanceof	Operador instanceof



Arreglos

- Es una estructura de datos que nos permite un conjunto de datos del mismo tipo.
- El tamaño se los arreglos de declarar una única vez al momento de su creación y no se puede modificar después como si se pudiera con un ArrayList
- Cada elemento se puede acceder por medio de un índice el cual empieza desde el número 0.

Formas de declaración:

1. Sin valores:

<tipo>[] <nombre> = new <tipo>[<tamaño>]

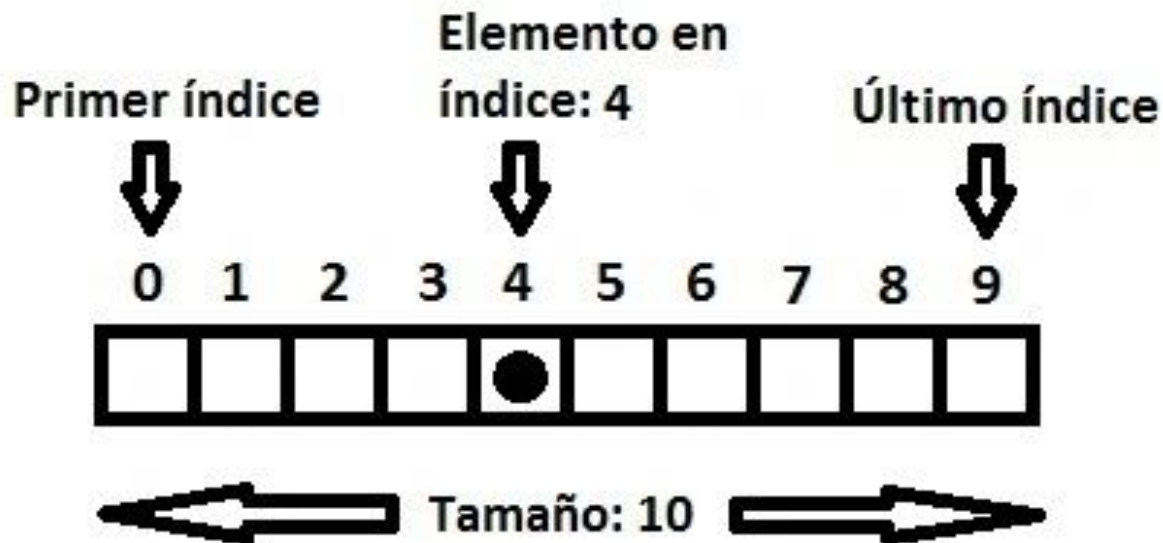
```
int[] nums = new int[10];
```

2. Con valores:

<tipo>[] nums = {<Datos almacenados>};

```
int[] nums = { 1, 7, 8, 2, 3, 10, 3, 8, 2, 10};
```

Índice de cada elemento de los arreglos



Nota: Para obtener el tamaño del arreglo se utiliza la propiedad
`length`, `nums.length`

Metodos de Ordenamiento

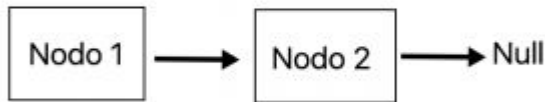
- Para poder ordenar una cantidad determinada de números almacenados en un vector o matriz, existen distintos métodos con distintas características y complejidad.
- Existe desde métodos más simples, como:
 - Metodo Burbuja
 - Método por Inserción
 - Método por Selección
 - Método Rápido



Listas Dinámicas

- Una lista dinámica es una colección de objetos enlazados entre sí.
- Estas pueden crecer tanto como se quiera.
- Está compuesta por una clase que maneja la lista y otra que contiene información de los objetos de la lista, llamados nodos.

Primero = Nodo1



Funciones

- Se indica el tipo de retorno indicado en caso de tener uno.
- Se debe indicar el tipo de cada parámetro.
- Si hay más de un parámetro de un parámetro se separan por medio de comas.
- Se utiliza la palabra reservada `return` para devolver un valor.
- Después del retorno no se ejecutará el código que esté debajo de él dentro de la función.

The diagram illustrates the components of a C++ function definition with the following code and annotations:

```
private int sumar(int numero1, int numero2)
{
    int suma = numero1 + numero2;
    return suma;
}
```

Annotations:

- Ámbito y tipo de dato del valor que retornará la función:** Points to `private int`.
- Parámetros de entrada:** Points to `int numero1, int numero2`.
- Nombre de la función:** Points to `sumar`.
- Instrucciones:** Points to the body of the function, `int suma = numero1 + numero2;` and `return suma;`.
- Valor de retorno:** Points to the value `suma` being returned.

Ámbito de la
declaración

Nombre del
procedimiento

private void limpiar ()

Instrucciones

{

txtNumero1.setText(null);

}

Ámbito de variables

El ámbito de una variable define su alcance de uso, es decir, indica en qué secciones de código una variable estará disponible.

Fuera de este ámbito, una variable no podrá ser accedida.

Tipos de ámbito:

- Local
- Global
- Estático



Main.java

```
1  class Main {  
2      public static boolean verificarEdad(int edad) {  
3          if (edad >= 18) {  
4              return true;  
5          }  
6          return false;  
7      }  
8  
9      public static void main(String[] args) {  
10         boolean respuesta = verificarEdad(18);  
11         System.out.println(respuesta);  
12         // Salidad: true  
13     }  
14 }
```


Recursividad

La recursividad es una característica que permite que un método se invoque a sí mismo. La recursividad es útil para resolver problemas definibles en sus propios términos.

- Directa o simple: El método se llama a sí mismo dentro de él.
- Indirecta, cruza o doble: Una función llama a otra y esta otra que también llama a la primera.



Recursividad Directa

```
Main.java
1 class Main {
2     public static int multiplicacion(int num1, int num2) {
3         // Caso base:
4         if (num2 == 1) {
5             return num1;
6         }
7         // Llamada recursiva:
8         return num1 + multiplicacion(num1, num2 - 1);
9     }
10
11     public static void main(String[] args) {
12         int resultado = multiplicacion(3, 2);
13         System.out.println(resultado);
14         // Salida: 6
15     }
16 }
```

Recursividad Indirecta

```
Main.java
1 ~ class Main {
2 ~     static boolean esPar(int n) {
3         if (n == 0) return true;
4         return esImpar(n - 1);
5     }
6
7 ~     static boolean esImpar(int n) {
8         if (n == 0) return false;
9         return esPar(n - 1);
10    }
11
12 ~    public static void main(String[] args) {
13        boolean resultado = esPar(3);
14        System.out.println(resultado);
15        // Salidad: falso
16    }
17 }
```

INPUT y OUTPUT JAVA

Output

```
System.out.println(); or  
System.out.print(); or  
System.out.printf();
```

Input

```
import java.util.Scanner;  
  
// create an object of Scanner  
Scanner input = new Scanner(System.in);  
  
// take input from the user  
int number = input.nextInt();
```

Versionamiento

Git

Es un sistema de control de versiones de software, diseñado para ser rápido y manejar grandes proyectos con muchos archivos de manera eficiente.

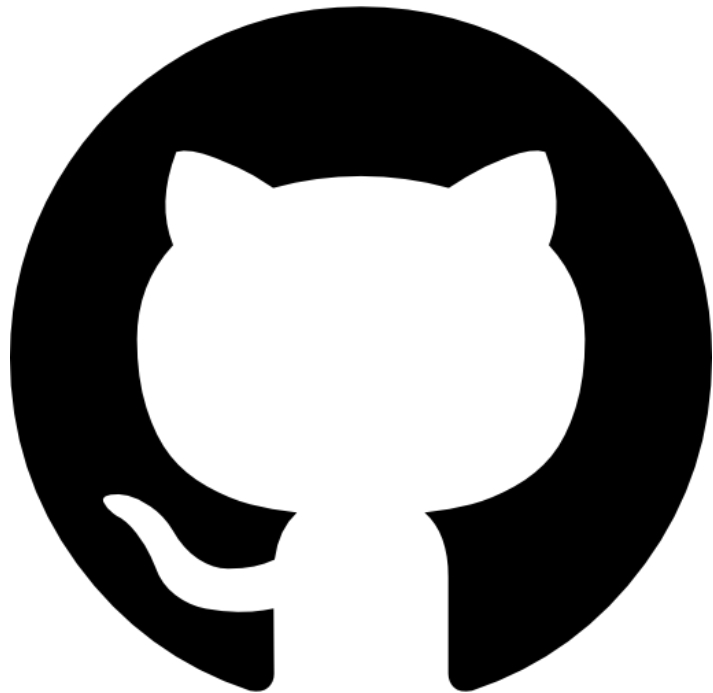
Git permite rastrear cambios realizados en el código fuente y mantener un historial de versiones, lo que facilita el trabajo en equipo y la resolución de conflictos.



GitHub

Es una plataforma en línea que ofrece alojamiento y control de versiones para proyectos de software utilizando Git.

Proporciona herramientas para colaboración, documentación y equipos de software para trabajar en productos juntos.



GitLab


Es una plataforma en línea que ofrece funciones similares a GitHub para control de versiones y colaboración en proyecto de software.

Es una solución de código abierto que se puede instalar en los servidores de la empresa.



Comandos Git

- Clonar repositorio: `git clone <url del repositorio>`
- Subir cambios al repositorio:
 - a. Inicializar un repositorio en el directorio local:
`git init`
 - b. Agregar los archivos modificados:
`git add .`
 - c. Realizar un “commit” de los cambios:
`git commit -m “Mensaje de commit”`
 - d. Conectar a un repositorio remoto:
`git remote add origin <url del repositorio>`
 - e. Sube los cambios al repositorio remoto:
`git push -u origin <nombre de la rama>`
- Descargar cambios:
 - a. `cd <nombre del directorio>`
 - b. `git pull origin <nombre de la rama>`



¿Dudas?

Tarea No. 1

Describir con sus palabras la funcionalidad de los modificadores de acceso:

- Public
- Private
- Protected

Fecha de entrega: 11/02/2023