|  |  |
| --- | --- |
| NOMBRES:  Alan Josué Zepeda Reyes | MATRICULA:  07090960 |
| NOMBRE DEL PROFESOR:  HECTOR ANTONIO AGUILAR MOGOLLAN | NOMBRE DEL CURSO: Programación Orientada a Objetos |
| MODULO 1 | ACTIVIDAD: ACTIVIDAD 2 |
| FECHA: Viernes 31 de Enero de 2025 |  |
| BIBLIOGRAFIA |  |

Parte 1. Strings y estructuras de selección

* Crea un programa Java que solicite al usuario ingresar su nombre.
* Utiliza estructuras de selección (*if* o *switch*) para verificar si el nombre ingresado coincide con uno específico que definas en tu programa; dependiendo de la coincidencia, muestra un mensaje personalizado.
* Aprovecha las operaciones de manejo de cadenas de texto para, así, realizar comparaciones con sensibilidad a mayúsculas o minúsculas y, de esta manera, proporcionar mensajes adecuados.

**CAPTURA DE TODO EL CODIGO**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Importación de la clase Scanner**: Esta línea importa la clase Scanner que se utiliza para obtener la entrada del usuario desde la consola.



**Creación del objeto Scanner**: Aquí se crea un objeto Scanner llamado scanner que lee la entrada del usuario desde la consola (System.in).



**Solicitud al usuario**: Esta línea muestra un mensaje en la consola solicitando al usuario que ingrese su nombre.



**Lectura de la entrada del usuario**: Esta línea lee la entrada del usuario y la almacena en una variable String llamada nombre.



**Definición del nombre específico**: Aquí se define una variable String llamada nombreEspecifico con el nombre específico que se utilizará para la comparación.



**Estructura de selección**: Esta es la parte donde se realiza la comparación. Se utiliza el método equalsIgnoreCase para comparar el nombre ingresado por el usuario con el nombre específico (nombreEspecifico) sin distinguir entre mayúsculas y minúsculas. Dependiendo del resultado de la comparación, se muestra un mensaje personalizado.



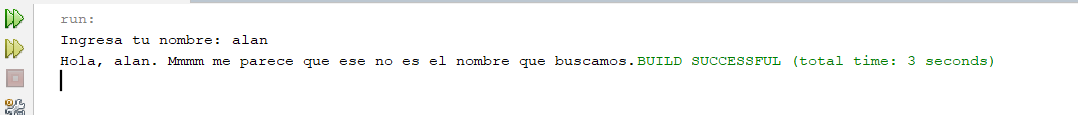
**Cerrar el objeto Scanner**: Finalmente, esta línea cierra el objeto Scanner para liberar los recursos utilizados



**EJECUCION**

Imagen que contiene Diagrama

Descripción generada automáticamente



Parte 2. Estructuras de repetición y arreglos

* Desarrolla un programa que genere una lista de números enteros pares desde dos hasta 100; para lograrlo, puedes utilizar una estructura de repetición (*for* o *while*).
* Almacena estos números en un arreglo.
* Luego, utiliza una estructura de repetición para recorrer el arreglo y mostrar en pantalla los números almacenados.

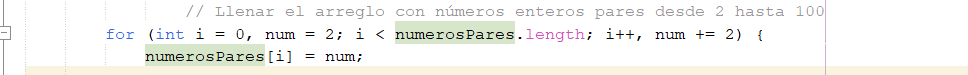
**CAPTURA DE TODO EL CODIGO**

Texto

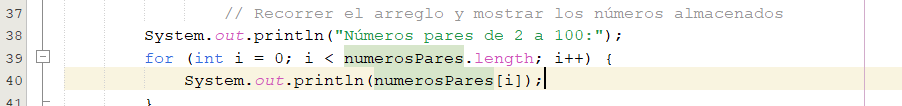
Descripción generada automáticamente

**Solicitar y leer el nombre**: System.out.println("Por favor, introduce tu nombre:"); muestra un mensaje pidiendo el nombre del usuario, y String nombre = scanner.nextLine(); lee el nombre ingresado por el usuario y lo almacena en una variable llamada nombre**.**

**Crear un arreglo**: int[] numerosPares = new int[50]; define un arreglo de enteros llamado numerosPares con capacidad para 50 elementos.

**Llenar el arreglo**: El ciclo for se utiliza para llenar el arreglo con números enteros pares. int i = 0, num = 2; inicializa el índice i y la variable num. i < numerosPares.length; continúa el ciclo mientras i sea menor que la longitud del arreglo. i++, num += 2 incrementa i y num en cada iteración. numerosPares[i] = num; asigna el valor de num a la posición i del arreglo.

**Recorrer y mostrar el arreglo**: System.out.println("Números pares de 2 a 100:"); muestra un título para la lista de números. El ciclo for (int numero : numerosPares) recorre cada elemento del arreglo numerosPares y System.out.println(numero); imprime cada número par en la consola.



**Cerrar el objeto Scanner**: scanner.close(); cierra el objeto Scanner para liberar los recursos asociados con él.



**EJECUCION**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente