



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO (ESCOM)



PROGRAMACIÓN ORIENTADA A OBJETOS

NOMBRE DEL ALUMNO:

- SANTOS MÉNDEZ ULISES JESÚS

TEMA:

- NIVELES DE ACCESO EN JAVA

NÚMERO DE TAREA: 1

FECHA DE ENTREGA:

- 23/04/2021

GRUPO:

- 2CM11

Niveles de acceso en Java

El ámbito de las variables se combina con su nivel de acceso, inicialmente el ámbito de las variables se establece en el momento de crear una variable y relacionarla con la posibilidad de acceder a la misma desde cualquier parte del programa.

Por norma, la variable tiene validez dentro del bloque en el que está encerrada entre llaves y depende del lugar donde fue declarada. Esto determinará que fragmento de código va a ser leído y escrito en la variable.

Para acceder a las variables, debemos considerar si en la clase está declarada como **public** o como **private**.

Desde la declaración de cualquier función propia de una clase, podemos acceder a las variables internas de esa clase en forma directa. Es posible realizar esta declaración en el bloque de código de una clase, en el bloque de código de una función o también en el bloque de código que se encuentra en el interior de una función.

Esta ubicación es importante pues solo podrá utilizarla el código que corresponde al bloque donde declaramos la variable.

Un conjunto de palabras clave nos permite controlar el nivel de acceso de una variable, estas palabras clave se utilizan en la declaración y deben de ser ubicadas delante del tipo de datos que corresponde a la variable.

Los niveles de acceso se utilizan para efectuar la declaración de una variable en el interior de una clase, así entonces, no se pueden usar en el interior de un método.

En los niveles de acceso se tiene una relación con las reglas de visibilidad, las reglas de visibilidad complementan o refinan el concepto de **encapsulamiento**. Los diferentes niveles de visibilidad dependen del lenguaje de programación pero siguen el estándar de C++, estos niveles de visibilidad son:

- **Nivel Privado:** Es el nivel más fuerte y es totalmente invisible para otras clases, solo miembros de la misma clase pueden acceder a atributos localizados en la sección privada.
- **Nivel Protegido:** Es posible aliviar el nivel de ocultamiento situando algunos atributos en la sección protegida de la clase. Estos atributos son visibles tanto para la misma clase como para las clases derivadas de la clase. Para las clases restantes permanecerá invisible.
- **Nivel Público:** Es el nivel más débil, se obtiene situando los atributos en la sección pública de la clase con lo cual se hacen visibles a todas las clases.

Los atributos están encapsulados dentro de una clase, se necesitará definir cuáles son las clases que tienen acceso a visualizar y cambiar los atributos. Esta característica se conoce como visibilidad de los atributos.

Los modificadores de acceso se utilizan para definir la visibilidad de los miembros de una clase (atributos y métodos) y de la propia clase.

A continuación se tienen los niveles de acceso de las variables:

| | |
|--------------------|---|
| Private | <ul style="list-style-type: none">• Donde la clase está definida es donde se utilizará la variable.<ul style="list-style-type: none">• private String nombre; |
| Protected | <ul style="list-style-type: none">• La variable se utiliza en la clase donde está definida, en las subclases de esta clase y en las clases que forman parte del mismo paquete.<ul style="list-style-type: none">• protected String codigo; |
| Public | <ul style="list-style-type: none">• La variable es accesible desde cualquier clase sin importar el paquete.<ul style="list-style-type: none">• public String nombre; |
| Ningún Modificador | <ul style="list-style-type: none">• La variable es accesible desde todas las clases que forman parte del mismo paquete.<ul style="list-style-type: none">• int num1, num2; |
| Static | <ul style="list-style-type: none">• Se usa para transformar una declaración de variable de instancia en declaración de variable de clase (utilizable sin que exista una instancia de clase).<ul style="list-style-type: none">• static float impuesto; |

Esto cumple con un principio fundamental de la programación orientada a objetos que es la **ocultación de la información**, que significa que determinados datos del interior de una clase no pueden ser accedidos por funciones externas de la clase.

Referencias Bibliográficas:

Paul, D (2016). *Cómo programar en Java*. Pp (79-80). Pearson.

Malik D (2013). *Programación JAVA-análisis de problemas*. Pp (388-391). CENGAGE Learning.