



El futuro digital
es de todos

MinTIC



Vigilada Mineducación

CICLO II: Programación Básica en Java

Misión
TIC2022





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Sesión 5: Introducción a Java

Programación Orientada a Objetos (POO)

Misión
TIC2022



Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Definir el diagrama de clase para la clase creada con método sin paso de parámetros.
2. Declarar un método con paso de parámetros
3. Definir el diagrama de clase para la clase creada con método con paso de parámetros.
4. Implementar los atributos de una clase como variables de instancias
5. Definir el diagrama de clase aplicando variables de instancias y el uso de los métodos set y get



Diagrama de Clases

- Recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos.
- Elementos de la Clase

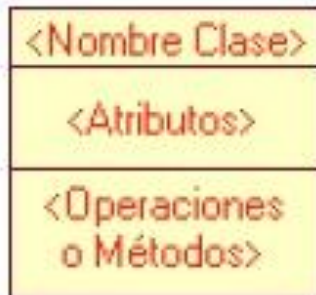




Diagrama de Clases: Atributos

- Los atributos describen a una clase. Pueden ser **Públicos, Privados o Protegidos**.
- **public (+)**: Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- **private (-)**: Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).
- **protected (#)**: Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven.



Diagrama de Clases: Métodos

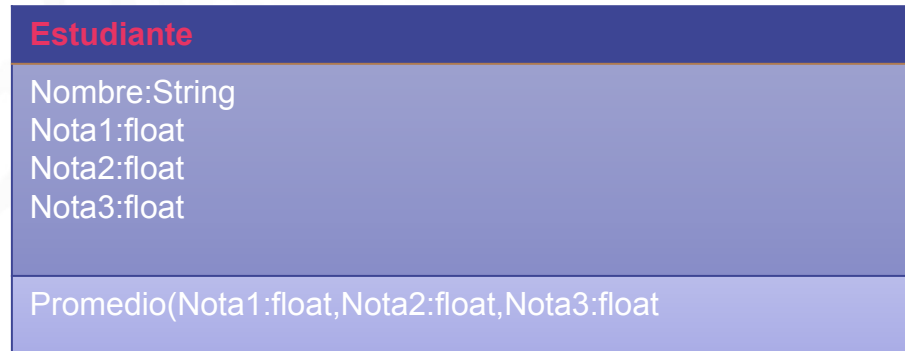
- Los métodos de una clase describen la forma en la cual ésta interactúa con su entorno. Pueden ser Públicos, Privados o Protegidos.
- **public (+):** Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- **private (-):** Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la misma clase lo pueden acceder).
- **protected (#):** Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser utilizado por métodos de la clase además de las subclases que se deriven (herencia)



Ejemplo

Enunciado: Construya una clase denominada estudiante, con los siguientes atributos Nombre, Nota1, Nota2, Nota3 y Definitiva. Solicite al usuario los valores para un estudiante y almacénelos en una instancia de la clase estudiante. Calcule la nota definitiva como el promedio simple de las tres notas. $(\text{nota1} + \text{nota2} + \text{nota3}) / 3$

Diagrama de Clase





Ejemplo – Estructura en Java

```
import java.util.Scanner;

public class Estu {
    public static void main(String[] args) {
        Scanner lea = new Scanner(System.in);
        Estudiante e1 = new Estudiante();//creando objeto para instanciar atributos de la clase
        /*implementación*/
    }
    //método promedio
    public static float prom(float nota1,float nota2, float nota3) {

        return (nota1+nota2+nota3)/3;
    }
}
//Clase estudiante
class Estudiante {

    public String nombre;
    public float nota1, nota2, nota3;
}
```




POO - Atributos

- Los objetos tienen atributos que llevan consigo cuando se usan en un programa.
- Los atributos son especificados como parte de la clase de la cual se toma el objeto.
- Los atributos existen antes que el método sea llamado en un objeto, mientras el método se esté ejecutando y después que el método completa la ejecución.
- Las clases constan de uno o más métodos que manipulan los atributos que pertenecen a un objeto particular de la clase.
- Son los datos que caracterizan a los objetos de una clase y determinan el estado de un objeto



POO - Atributos

- Los atributos son representados como variables en una declaración de clase.
- Estas variables son declaradas dentro de la clase, pero fuera de la declaración de los métodos dentro de la clase.
- Cada objeto de una clase mantiene su propia copia del atributo, y el campo que representa el atributo es conocido como variable de instancia.
- Cada objeto de una clase conoce los valores de sus atributos, pero no los valores de los mismos atributos en otros objetos definidos sobre la misma clase.



Encapsulamiento

¿Qué es?

- Proceso para ocultar los atributos y métodos de una clase

¿Por qué?

- El comportamiento y los atributos de un objeto no debe ser alterado

¿Para qué?

- Proteger los datos de modificaciones incontroladas



Encapsulamiento en Java

- En Java se definen métodos y atributos públicos para poder revisar su contenido e incluso ser modificados.
- El encapsulamiento consiste en ocultar los atributos y métodos de una clase, para evitar el acceso y la modificación desde otra clase.
- Para implementar el encapsulamiento se definen estos atributos y métodos con acceso privado.
- El encapsulamiento es útil al compartir las clases con otros programadores.
- Los métodos encapsulan un conjunto de instrucciones que se puede ejecutar tantas veces como sea necesario. El código del método queda oculto y sólo es necesario conocer su interfaz (parámetros y valor de retorno)





Métodos set y get

- En ocasiones es necesario dar acceso a algunos atributos de nuestra clase, sin eliminar el encapsulamiento.
- Estos métodos permiten el acceso a los atributos de una clase, que están encapsulados como privados.
- Con el método set se establecen o asignan valores a los atributos encapsulados en una clase. Este recibe un parámetro de entrada con el cual se establece el valor al atributo encapsulado y no retorna nada.
- Con el método get se obtiene o retorna el valor de un atributo encapsulado. El nombre del método se define con el tipo de valor que se desea retornar. No recibe parámetros.



Diagrama de Clase con variable encapsulada, set y get

Vertebrado
- nombreVertebrado : String
+ setNombreVertebrado(nombreV: String) + getNombreVertebrado() : String + imprimeMensaje()



Métodos set y get Java - Sintaxis

Los métodos get y set, son simples métodos que usamos en las clases para mostrar (get) o modificar (set) el valor de un atributo.

Esta es la sintaxis de cada uno:

Método get

```
public tipo_dato_atributo getAtributo (){  
    return atributo;  
}
```

Método set

```
public void setAtributo (tipo_dato_atributo variable){  
    atributo = variable;  
}
```




El futuro digital
es de todos

MinTIC



Ejercicios para practicar





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS
POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!



Mision
TIC 2022