

Portfolio: Objetos en Programación – Unidad 4

Apuntes y notas de clase

Conceptos claves

Atributos: Son las características o propiedades de una clase, generalmente representadas por sustantivos. Deben estar precedidos por un modificador de acceso (usualmente privado por seguridad) y su tipo de dato.

```
private int edad;
```

```
protected String nombre;
```

Métodos: Son las acciones que una clase puede realizar, representadas por verbos. Deben llevar un modificador de acceso (generalmente público para su uso en otras clases) y su tipo de retorno.

```
public void registrarGol();
```

Clase: Es un modelo que define la estructura de los objetos. Sus principales componentes son los atributos y los métodos.

```
class Persona { ... }
```

Objeto: Es una instancia de una clase. Se crea utilizando los constructores definidos en la clase.

```
Coche miCoche = new Coche("Seat Ibiza");
```

Constructor: Es un método especial que se usa para inicializar un objeto. Puede ser sin parámetros, con parámetros o por defecto.

```
Persona(String nombre, String apellidos, int edad) { ... }
```

```
Persona() { ... } // Constructor sin parámetros
```

this: Palabra clave utilizada para acceder a los atributos de una clase cuando hay una variable local con el mismo nombre. También se puede usar para llamar a otro constructor.

```
this.nombre = nombre;
```

```
this();
```

Modificadores de Acceso: Determinan el nivel de acceso a los atributos, métodos o constructores dentro de una clase.

- **public (+):** Accesible desde cualquier parte del programa.
- **private (-):** Solo accesible dentro de la misma clase.
- **protected (#):** Accesible por las subclases y dentro del mismo paquete.
- **Sin modificador:** Solo accesible dentro del mismo paquete.

```
protected String dni;
```

Getters y Setters: Son métodos públicos que permiten acceder y modificar atributos privados.

```
public void setEdad(int edad) { this.edad = edad; }
```

```
public int getEdad() { return this.edad; }
```

Estático: Atributos o métodos que pertenecen a la clase en lugar de a las instancias individuales.

```
static void mostrarFecha(int dia) { ... }
```

Enumeraciones: Tipo de dato que solo puede contener ciertos valores predefinidos.

```
enum Genero { MASCULINO, FEMENINO, OTRO }
```

Herencia: Permite que una clase herede atributos y métodos de otra clase. Se usa la palabra clave extends.

```
class Coche extends Vehiculo { ... }
```

super: Se utiliza para hacer referencia a la superclase, ya sea para llamar a su constructor o acceder a sus métodos y atributos.

```
super();
```

```
super.velocidadMaxima;
```

Override: Permite redefinir un método de una clase padre en una subclase.

```
@Override
```

```
public String toString() { ... }
```

Polimorfismo: Una subclase puede implementar de manera diferente algunos métodos de la superclase.

```
Atleta jugador = new Futbolista(...);
```

Object: Es la superclase de todas las clases en Java. Contiene métodos fundamentales para la manipulación de objetos.

```
toString(), equals(), getClass()
```

Métodos Abstractos: Son métodos sin implementación que deben ser definidos en las subclases.

```
public abstract void emitirSonido();
```

Clases Abstractas: Son aquellas que contienen al menos un método abstracto y no pueden ser instanciadas directamente.

```
abstract class Animal { ... }
```

Atributos Finales: Son constantes cuyo valor no puede modificarse después de su inicialización.

```
final double PI = 3.141592;
```

Métodos Finales: No pueden ser sobrescritos en una subclase.

```
public final boolean validarUsuario(String contraseña) { ... }
```

Clases Finales: No pueden ser heredadas por otras clases.

```
final class Utilidad { ... }
```

Interfaces: Definen un conjunto de métodos que una clase debe implementar. Permiten una forma de "herencia múltiple".

```
public interface Volador { ... }
```

```
public class Pajaro implements Volador { ... }
```

Paso de Argumentos:

- **Por Valor:** Se pasa una copia del valor original, por lo que los cambios no afectan a la variable original. Se aplica a tipos primitivos.
- **Por Referencia:** Se pasa la referencia a un objeto, por lo que cualquier modificación afecta directamente al original.

Alejandro González Román 1º DAW.