

¿Qué es la Programación Orientada a Objetos(POO)?

1. Programación Orientada a Objetos (POO): Es una forma de programar donde organizas el código en "objetos" que combinan datos y acciones.

¿Cuál es la historia de la Programación Orientada a Objetos?

2. Historia de la POO: Nació en los años 60 con el lenguaje Simula y luego Smalltalk en los 70 fue el primero en usarla completamente. Desde entonces, lenguajes como Java y Python la han adoptado.

¿Cuáles son los lenguajes de programación, que soportan la POO?.

3. Lenguajes que soportan POO:Java, Python, C++, C#, Ruby, entre otros.

¿Qué es un objeto?

4. Objeto: Es algo creado a partir de una clase que combina datos y acciones.

¿Qué es un atributo?

5. Atributo: Es la información que guarda un objeto, como sus características.

¿Qué es un método?

6. Método: Son las acciones o comportamientos que un objeto puede realizar.

¿Qué es un método void?

7. \*Método void: Es un tipo de método que no devuelve ningún resultado, solo realiza una acción.

¿Cuáles son los tipos de método?

8. \*Tipos de método: Los que devuelven algo, los que no (void), los que son estáticos (static), y los abstractos (sin implementación).

¿Qué es una clase, describa sus características?

9. Clase: Es el plano o molde que define cómo serán los objetos y qué podrán hacer.

¿Para qué sirve Getter y Setter?

10. Getter y Setter: Métodos para obtener y modificar los datos privados de una clase.

¿Para qué sirve this?

11. This: Sirve para referirse al objeto actual dentro de la clase.

¿Para qué sirve null?

12. Null: Significa que una variable no tiene ningún valor o no está apuntando a nada.

¿Qué es instanciar?

13. Instanciar: Es crear un objeto a partir de una clase.

¿Cuáles son los pilares de la POO?

14. Pilares de la POO: Son la abstracción, encapsulamiento, herencia y polimorfismo, que son las bases para estructurar el código.

¿Qué es abstracción?

15. Abstracción: Es ocultar los detalles complicados y mostrar solo lo esencial.

¿Qué es polimorfismo?

16. Polimorfismo: Es cuando un método puede comportarse de diferentes formas dependiendo del objeto que lo use.

¿Qué es encapsulamiento?

17. Encapsulamiento: Es proteger los datos dentro de una clase y solo permitir acceder a ellos de forma controlada.

¿Cuáles son los tipos de modificadores de acceso en JAVA?

18. Modificadores de acceso: Son reglas para controlar quién puede ver o usar los datos y métodos (public, private, protected).

¿Qué es herencia?

19. Herencia: Es cuando una clase nueva puede usar las características de otra clase.

¿Qué es un constructor?

20. Constructor: Es un método que se ejecuta al crear un objeto para inicializar sus datos.

¿Qué es un destructor?

21. Destructor: Libera la memoria usada por un objeto cuando ya no se necesita (en Java esto lo hace automáticamente el recolector de basura).

¿En qué consiste el concepto de memoria stack y heap.

22. Memoria stack y heap: Stack es donde se guardan variables temporales y heap es donde se almacenan los objetos.

¿Qué es una clase anidada?

23. Clase anidada: Es una clase dentro de otra clase.

¿Qué es una clase abstracta?

24. Clase abstracta: Es una clase que no puede ser usada directamente, solo puede ser "completada" por otras clases.

¿Qué es un mensaje en POO?

25. Mensaje en POO: Es simplemente llamar a un método de un objeto para que haga algo.

¿En qué consiste la sobrecarga de constructores y métodos?

26. Sobrecarga: Es cuando puedes tener varios métodos o constructores con el mismo nombre pero con diferentes parámetros.

¿En qué consiste el uso de la palabra Final?

27. Final: Hace que algo no se pueda cambiar. Puede aplicarse a variables, métodos o clases.

¿Qué es una colección?

28. Colección: Es un grupo de objetos, como una lista o conjunto.

¿Qué es arraylist?

29. ArrayList: Es una lista en Java que puede cambiar de tamaño automáticamente.

¿Qué es una interfaz?

30. Interfaz: Es como un contrato que dice qué métodos debe implementar una clase, pero no cómo deben funcionar.

¿Qué es una excepción?

31. Excepción: Es un error o situación inesperada que interrumpe el programa, y se puede manejar para evitar que el programa falle.

Á	Á	V	J	R	C	Q	A	Ü	Ü	S	D	Í	U	I	D	C	Ó	O
S	Ü	J	L	V	J	Ú	P	É	A	Ó	Í	F	H	T	E	Y	Q	Ó
K	B	L	V	I	S	I	B	I	L	I	D	A	D	J	P	N	F	U
A	Á	W	L	Ó	L	Í	M	A	R	Q	Y	T	G	J	E	Á	H	M
P	Ó	B	H	Í	O	K	L	D	N	K	V	R	Z	P	N	Á	P	X
K	N	B	C	J	C	R	N	B	Ñ	J	I	I	Ñ	F	D	Y	Ü	A
L	K	F	G	H	Z	Q	U	L	J	P	N	B	Ó	C	E	K	Ó	Y
P	Q	Á	J	X	R	O	Á	M	Ñ	U	C	U	Z	J	N	R	Z	A
H	B	F	V	O	O	W	W	O	Á	M	Q	T	N	K	C	É	A	S
C	Ú	Á	L	F	W	I	P	O	L	I	M	O	R	F	I	S	M	O
C	O	M	P	O	S	I	C	I	O	N	R	S	N	P	A	L	E	C
Í	G	V	N	F	J	O	B	J	E	T	O	Ñ	F	P	L	I	T	I
F	I	G	L	C	H	C	L	A	S	E	P	Z	D	W	W	G	O	A
H	Q	X	S	O	B	R	E	C	A	R	G	A	T	I	E	É	D	C
É	B	T	T	N	W	M	O	D	I	F	I	C	A	D	O	R	O	I
A	D	I	N	S	T	A	N	C	I	A	C	I	O	N	K	C	U	Ó
Ó	Q	G	É	T	H	E	R	E	N	C	I	A	Z	I	R	U	R	N
Ü	O	P	E	R	A	C	I	O	N	E	S	G	Í	L	X	H	U	Ú
R	I	J	E	U	B	O	S	P	W	P	I	R	S	K	Q	J	P	B
B	V	Y	R	C	S	Ü	L	Ü	O	A	R	E	Q	Í	P	D	Ó	L

1. OPERACIONES
2. CONSTRUCTOR
3. COMPOSICION
4. POLIMORFISMO
5. MODIFICADOR
6. HERENCIA
7. METODO
8. SOBRECARGA
9. INSTANCIACION
10. OBJETO
11. DEPENDENCIA
12. ASOCIACIÓN
13. VISIBILIDAD
14. CLASE
15. AGREGACIÓN
16. ATRIBUTOS
17. ABSTRACTA
18. INTERFACE