

Preguntas

1. ¿Cuál es la mejor descripción que define el concepto de clase en la POO?

Respuesta:

c. Es un modelo o plantilla a partir de la cual creamos objetos.

2. ¿Cuáles son los elementos que definen un objeto?

Respuesta:

b. Atributos y métodos.

3. ¿Señale cuál de los siguientes fragmentos de código emplea el concepto de herencia?

Respuesta:

a. `public class Componente extends Producto`

4. ¿Qué es instanciar una clase?

Respuesta:

d. Crear un objeto a partir de una clase.

5. Defina una clase en la cual se describa el concepto de persona.

La clase debe poseer las siguientes propiedades:

- Nombre
- Apellido
- Altura
- Peso

Métodos:

- **Comer:** A medida que el usuario coma, aumentará su peso.
- **Creecer:** A medida que crezca, aumentará su altura.

Los métodos anteriores deben retornar tanto el peso como la altura. Además, debe incluirse un constructor que reciba los parámetros: nombre, apellidos, altura y peso.

Ejemplo en código:

```
class Persona(  
    var nombre: String,  
    var apellido: String,  
    var altura: Double,  
    var peso: Int  
) {  
    fun comer(aumento: Int = 1) {  
        peso += aumento  
    }  
  
    fun creecer(incremento: Double = 0.05) {  
        altura += incremento  
    }  
}
```

...

6. **¿Qué es una arquitectura de aplicaciones?**

Respuesta:

Es una forma de organizar una aplicación dividiendo responsabilidades en capas o módulos, lo que facilita el mantenimiento y la escalabilidad al evitar el acoplamiento. Además, mejora la reutilización del código, facilita la colaboración en equipo y permite una adaptación más ágil ante cambios.

7. **¿Con qué arquitecturas estás familiarizado y cuál prefieres?**

Respuesta:

Estoy familiarizado con la Clean Architecture y con el enfoque de módulos por feature. No tengo una preferencia fija, ya que la elección depende de las necesidades específicas de cada proyecto.

- **Clean Architecture:** Organiza el código en módulos separados (Domain, Data y UI).
- **Módulo por Feature:** Integra las capas de Domain, Data y UI dentro de cada módulo, permitiendo que equipos grandes trabajen en diferentes features de forma independiente y minimizando el impacto entre ellas.

8. **¿Cómo funcionan los hilos de programación?**

Respuesta:

Son tareas diferentes que se ejecutan simultáneamente, aprovechando al máximo los múltiples núcleos de los dispositivos.

9. **¿Qué procesos se pueden realizar y cuáles no en una tarea de segundo plano en un dispositivo móvil?**

Respuesta:

- **Se pueden ejecutar en segundo plano:**
 - Procesamiento de datos
 - Descargas
 - Conversión de formatos (por ejemplo, mp3)
 - Tareas de cálculo (con herramientas como WorkManager para algunas operaciones)
- **No se pueden ejecutar en segundo plano:**
 - La mayoría de las interacciones de UI, ya que dependen de la interacción directa del usuario.

10. **¿Una notificación es un elemento que comúnmente se ejecuta en una tarea de primer plano o segundo plano? Justifique su respuesta.**

Respuesta:

Las notificaciones generalmente se disparan mediante componentes como Broadcast Receivers o servicios en segundo plano, ya que responden a eventos sin requerir la interacción inmediata del usuario.

11. **¿Qué es una API Rest?**

Respuesta:

Una API REST es un conjunto de reglas para construir servicios en la nube y comunicarse mediante HTTP, Sockets o herramientas como GraphQL, permitiendo consumir y almacenar información de múltiples aplicaciones y usuarios.

12. **¿Mencione y explique los principales métodos que componen una API Rest.**

Respuesta:

- **GET:** Recupera datos de un recurso.

- **POST:** Crea un nuevo recurso.
- **PUT:** Actualiza un recurso existente.
- **DELETE:** Elimina un recurso.
- **PATCH:** Actualiza parcialmente un recurso.

13. **¿Cuáles son las ventajas de usar JSON?**

Respuesta:

Es ligero, fácil de leer y ampliamente soportado.

14. **¿Cuál es generalmente el código de respuesta exitosa de una API REST?**

Respuesta:

Los códigos 200 y 201 indican respuestas exitosas.

15. **¿Qué es un recurso?**

Respuesta:

Es una entidad accesible mediante una URL que representa datos o funcionalidades.

16. **¿Qué ventajas ofrece una API REST para el desarrollo?**

Respuesta:

Facilita la escalabilidad al reducir el acoplamiento y aumentar la cohesión, permite la integración entre plataformas, mejora la estabilidad y el rendimiento, y simplifica el mantenimiento de las aplicaciones.

17. **¿Qué librerías de base de datos has utilizado para aplicaciones móviles?**

Respuesta:

He utilizado Room, SQLite (nativo) y Realm.

18. **¿Qué librerías o mecanismos has utilizado para la inyección de dependencias?**

Respuesta:

He utilizado Dagger, Dagger Hilt, Koin y, en algunos casos, he creado todas las dependencias manualmente.