LE0NARDO SEPULVEDA BEDOYA

RESPUESTAS MD

Seccion 2

if / if-else: Se utiliza cuando necesitamos tomar decisiones basadas en condiciones específicas. Es muy útil para validar datos, evaluar estados o elegir entre diferentes caminos de ejecución.

switch: Es ideal cuando tenemos múltiples opciones para una sola variable. Por ejemplo, en menús de opciones numéricas o al evaluar constantes como caracteres.

for: Perfecto para recorrer rangos conocidos, como contar del 1 al 10 o iterar a través de un array usando índices.

while: Se emplea cuando no sabemos cuántas veces se repetirá el ciclo, pero sí tenemos claro cuál es la condición de salida. Un ejemplo sería seguir pidiendo datos hasta que el usuario escriba "salir".

do-while: Funciona de manera similar al while, pero asegura que el bloque se ejecute al menos una vez. Es ideal para validar contraseñas o repetir una acción al menos una vez.

break y continue: Se utilizan para controlar el flujo dentro de los bucles. break permite salir completamente del bucle, mientras que continue salta una iteración y continúa con la siguiente.

Bucles anidados: Son necesarios cuando trabajamos con estructuras como tablas, matrices o cálculos combinados, como las tablas de multiplicar.

for-each: Es una forma más limpia y directa de recorrer arreglos o colecciones sin tener que preocuparnos por los índices.

Recursión: Es útil cuando un problema puede dividirse en subproblemas del mismo tipo, como calcular Fibonacci o factoriales, aunque hay que tener cuidado con el uso de memoria.

Colecciones (ArrayList, Set) y utilidades (Collections.sort, Arrays.asList): Se utilizan para manipular datos dinámicos. Son preferibles a los arrays cuando el tamaño de los datos puede cambiar o se necesita funcionalidad adicional, como ordenamiento o eliminación de duplicados.

Seccion 3

Punto 12

La diferencia de List, Set y Map en Java es que: List es una colección ordenada que permite duplicados y también permite el acceso por posición; Set es una colección que no permite duplicados, aunque no tiene un orden específico; y Map almacena pares clave-valor donde las claves son únicas

Reflexion, punto 15 – seccion 3

ArrayList: Ideal cuando necesito una lista ordenada que pueda crecer dinámicamente y acceder a elementos por índice rápidamente.

LinkedList: Útil si voy a insertar o eliminar muchos elementos al principio o al medio de la lista. Más eficiente que ArrayList en esos casos.

HashSet: Lo uso cuando quiero almacenar elementos únicos (sin duplicados) y no me importa el orden. Es rápido para buscar si un elemento existe.

HashMap: Perfecto cuando necesito asociar claves con valores, como un nombre con su edad. Es rápido para insertar, buscar y eliminar por clave.

Seccion 4

Punto 15

Diferencias entre == y equals() en Java:

== compara referencias (es decir, si dos variables apuntan al mismo objeto en memoria).

equals() compara contenido (es decir, si los datos dentro de los objetos son iguales), si está sobreescrito correctamente.