**Preguntas teóricas**

1. ¿Qué es un método en Java?

Un método es una colección de declaraciones que realizan una tarea específica. Los métodos son bloques de código que pueden ser invocados y ejecutados desde otros lugares dentro del programa. Cada programa Java consta al menos de un método llamado "main", que es el punto de entrada del programa y donde comienza la ejecución.

1. ¿Cuáles son los cuatro tipos de métodos en Java?

Con parámetros y con retorno

Con parámetros y sin retorno

Sin parámetros y con retorno

Sin parámetros y sin retorno

1. ¿Cuál es la diferencia entre un método con parámetros y sin retorno y un método sin parámetros y con retorno?

La diferencia clave es si el método recibe datos de entrada a través de parámetros y si devuelve algún valor como resultado. Los métodos con parámetros y sin retorno se utilizan para realizar acciones o tareas con los datos proporcionados, mientras que los métodos sin parámetros y con retorno se utilizan para obtener un valor calculado o recuperado.

1. ¿Cuál es la diferencia entre un método con parámetros y con retorno y un método con parámetros y sin retorno?

**Método con parámetros y con retorno**: Acepta valores de entrada, realiza operaciones y devuelve un valor como resultado. La firma incluye tipos y nombres de parámetros, así como el tipo de valor de retorno.

**Método con parámetros y sin retorno:** Acepta valores de entrada, realiza operaciones y no devuelve un valor explícito (tipo de retorno "void"). La firma incluye tipos y nombres de parámetros, pero no incluye un tipo de retorno en la firma.

1. ¿Por qué son importantes los métodos en Java?

Division de la funcionalidad, Realizacion de código, modularidad, legibilidad y mantenimiento, abstracion, testing y depuración, escalabilidad.

1. ¿Qué es una estructura de datos?

Una estructura de datos define cómo se almacenan los datos y cómo se relacionan entre sí para realizar operaciones y tareas específicas de manera eficiente.

Arrays, linked lists, stacks y queues, arboles, grafos, hastables.

1. ¿Cuál es la diferencia entre un arreglo, un arrayList y un HashMap?

Array: Datos estáticos, no se puede cambiar la longitud después de su creación, acceso rápido y directo a través de un índice, son eficientes en términos de memoria, pero tienen limitaciones en la capacidad de crecimiento y manipulación de los elementos.

ArratList: Tamaño dinamico, crecer y reducirse cuando sea necesario, agregar, eliminar, acceder de manera eficiente, mas flexibles que los arreglos, menos eficientes en términos de memoria debido a la gestión interna del tamaño.

HashMap: Almacena pares de clave-valor, permite asociar con valor con clave única, se accede a través de la clave, no mantiene un orden especifico de los elementos, búsqueda, inserción y eliminación de elementos son eficientes en términos de tiempo.

La clave deben ser únicas y los valores pueden repetirse.

Los arreglos son estructuras estáticas con un tamaño fijo y acceso rápido,pero carecen de flexibilidad.

Los ArrayLists son dinámicos en tamaño y más flexibles, pero pueen requerir más memoria debido a su capacidad de crecimiento.

Los HashMaps almacenan clave-valor y ofrecen búsqueda eficiente basada en claves, pero no mantienn un orden especifico.

**Ejercicios prácticos**

1. Escribe un método sin parámetros y sin retorno que imprima en pantalla tu nombre completo.

Metodo main

public class Principal1{

public static void main(String[] args) {

SinPSinR1 SinPSinR = new SinPSinR1();

SinPSinR.imprimirNombre();

}

}

Método Sin Parametros y Sin Retorno

public class SinPSinR1 {

public void imprimirNombre() {

System.***out***.println("Mi nombre es Luis");

}

}

1. Escribe un método con parámetros y sin retorno que reciba un número entero y lo imprima en pantalla si es mayor que 10.

Metodo main

public class Principal2{

public static void main(String[] args) {

ConPSinR2 ConPSinR = new ConPSinR2();

ConPSinR.imprimirNumero(0);

}

}

Metodo Con Parametros y Sin Retorno

import javax.swing.JOptionPane;

public class ConPSinR2 {

public void imprimirNumero(int number) {

number = Integer.*parseInt*(JOptionPane.*showInputDialog*("Ingrese un numero"));

if(number <= 10) {

System.***out***.println("No se puede imprimir el numero porque no es mayor a 10");

}else {

System.***out***.println(number);

}

}

}

1. Escribe un método sin parámetros y con retorno que genere un número aleatorio entre 1 y 100 y lo devuelva como resultado.

Método main

public class Principal3{

public static void main(String[] args) {

SinPConnR3 sinPConnR3 = new SinPConnR3();

int numeroAleatorio = SinPConnR3.*imprimirNumeroAleatorio*();

System.***out***.println("Numero aleatorio: " + numeroAleatorio);

}

}

Metodo Sin Parametros y Con Retorno

public class SinPConnR3 {

public static int imprimirNumeroAleatorio() {

return (int) (Math.*random*()\*100);

}

}

1. Escribe un método con parámetros y con retorno que reciba un arreglo de enteros y devuelva el valor máximo del arreglo.

Metodo main

public class Principal4{

public static void main(String[] args) {

int[] numeros = { 1, 2, 3, 14, 5, 7, 9, 8 };

ConPConR4 conPConR4 = new ConPConR4();

int valorMaximo = conPConR4.encontrarMaximo(numeros);

System.***out***.println("El valor máximo del areglo es: " + valorMaximo);

}

}

Metodo Con Parametros y Con Retorno

public class ConPConR4 {

public int encontrarMaximo(int[] arreglo) {

int maximo = arreglo[0];

for(int num : arreglo) {

if(num > maximo) {

maximo = num;

}

}

return maximo;

}

}

1. Escribe un método con parámetros y con retorno que reciba dos arreglos de enteros del mismo tamaño y devuelva un nuevo arreglo que contenga la suma de los elementos de ambos arreglos.

Metodo main

public class Principal5{

public static void main(String[] args) {

int[] arreglo = { 1, 2, 3, 14, 5 };

int[] arreglo2 = {2, 3, 4, 15, 6 };

ConPConR5 conPConR5 = new ConPConR5();

int[] suma = ConPConR5.*sumaArreglos*(arreglo, arreglo2);

for(int num : suma) {

System.***out***.println(num + " ");

}

}

}

Metodo Con Parametros y Con Retorno

public class ConPConR5 {

public static int [] sumaArreglos(int[] arreglo, int [] arreglo2) {

int longitud = arreglo.length;

int [] newArray = new int [longitud];

for(int i = 0; i < longitud; i++){

newArray[i] = arreglo[i] + arreglo2[i];

}

return newArray;

}

}

¡Recuerda practicar y experimentar con los métodos en Java para que puedas aprovechar todo su potencial en tus programas!

**Código en Java de los ejercicios prácticos**

1. Realice un algoritmo que use un método sin parámetros y sin retorno que imprime en pantalla tu nombre completo:
2. public class SinPSinR1 {
3. public void imprimirNombre() {
4. System.***out***.println("Mi nombre es Luis");
5. }
6. }
7. Realice un algoritmo que use un Método con parámetros y sin retorno que recibe un número entero y lo imprime en pantalla si es mayor que 10:
8. import javax.swing.JOptionPane;
9. public class ConPSinR2 {
10. public void imprimirNumero(int number) {
11. number = Integer.*parseInt*(JOptionPane.*showInputDialog*("Ingrese un numero"));
12. if(number <= 10) {
13. System.***out***.println("No se puede imprimir el numero porque no es mayor a 10");
14. }else {
15. System.***out***.println(number);
16. }
17. }
18. }
19. Realice un algoritmo que use un Método sin parámetros y con retorno que genera un número aleatorio entre 1 y 100 y lo devuelve como resultado:
20. public class SinPConnR3 {
21. public static int imprimirNumeroAleatorio() {
22. return (int) (Math.*random*()\*100);
23. }
24. }
25. Realice un algoritmo que use un Método con parámetros y con retorno que recibe un arreglo de enteros y devuelve el valor máximo del arreglo .
26. public class ConPConR4 {
27. public int encontrarMaximo(int[] arreglo) {
28. int maximo = arreglo[0];
29. for(int num : arreglo) {
30. if(num > maximo) {
31. maximo = num;
32. }
33. }
34. return maximo;
35. }
36. }
37. Realice un algoritmo que use un Método con parámetros y con retorno que recibe un ArrayList de enteros y devuelve el valor máximo de la lista .
38. import java.util.ArrayList;
39. public class MaxValueFinder {
40. public static void main(String[] args) {
41. ArrayList<Integer> numbers = new ArrayList<>();
42. numbers.add(10);
43. numbers.add(5);
44. numbers.add(20);
45. numbers.add(8);
46. int maxValue = *findMaxValue*(numbers);
47. System.***out***.println("The maximum value in the list is: " + maxValue);
48. }
49. public static int findMaxValue(ArrayList<Integer> list) {
50. if (list.isEmpty()) {
51. throw new IllegalArgumentException("no esta en la lista");
52. }
53. int max = list.get(0);
54. for (int num : list) {
55. if (num > max) {
56. max = num;
57. }
58. }
59. return max;
60. }
61. }
62. Realice un algoritmo que use un Método con parámetros y con retorno que recibe dos arreglos de enteros del mismo tamaño y devuelve un nuevo arreglo que contiene la suma de los elementos de ambos arreglos:
63. public class Principal5{
64. public static void main(String[] args) {
65. int[] arreglo = { 1, 2, 3, 14, 5 };
66. int[] arreglo2 = {2, 3, 4, 15, 6 };
67. ConPConR5 conPConR5 = new ConPConR5();
68. int[] suma = ConPConR5.*sumaArreglos*(arreglo, arreglo2);
70. for(int num : suma) {
71. System.***out***.println(num + " ");
72. }
73. }
74. }
75. Realiza un algoritmo donde se simule un diccionario simple. Crea un HashMap o HashTable donde las claves sean palabras en inglés y los valores sean las traducciones al español. Permite al usuario ingresar una palabra en inglés y muestra su traducción en español si está en el diccionario.

imporgt java.util.HashMap;

import java.util.Scanner;

public class SimpleDictionary {

public static void main(String[] args) {

HashMap<String, String> dictionary = new HashMap<>();

dictionary.put("apple", "manzana");

dictionary.put("banana", "plátano");

dictionary.put("car", "coche");

dictionary.put("house", "casa");

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Ingrese una palabra en ingles: ");

String englishWord = scanner.nextLine().toLowerCase();

if (dictionary.containsKey(englishWord)) {

String spanishTranslation = dictionary.get(englishWord);

System.***out***.println("Traduccion a español: " + spanishTranslation);

} else {

System.***out***.println("Esa palabra no esta en el diccionario.");

}

}

}

1. Crea un programa que permita a un profesor registrar las notas de sus estudiantes. Utiliza un HashMap donde las claves sean los nombres de los estudiantes y los valores sean las notas correspondientes. El programa debe permitir al profesor agregar estudiantes y asignar sus notas, así como también consultar la nota de un estudiante específico.}
2. import java.util.HashMap;
3. import java.util.Map;
4. import java.util.Scanner;
5. public class RegistroNotas {
6. public static void main(String[] args) {
7. HashMap<String, Double> scores = new HashMap<>();
8. Scanner scanner = new Scanner(System.***in***);
9. while (true) {
10. System.***out***.println("1. Agregar estudiante y nota");
11. System.***out***.println("2. Consultar nota de estudiante");
12. System.***out***.println("3. Salir");
13. System.***out***.print("Seleccione una opción: ");
14. int opcion = Integer.*parseInt*(scanner.nextLine());
15. if (opcion == 1) {
16. System.***out***.print("Ingrese el nombre del estudiante: ");
17. String studentName = scanner.nextLine();
18. System.***out***.print("Ingrese la nota del estudiante: ");
19. double score = Double.*parseDouble*(scanner.nextLine());
20. scores.put(studentName, score);
21. System.***out***.println("Estudiante agregado con éxito.");
22. } else if (opcion == 2) {
23. System.***out***.print("Ingrese el nombre del estudiante: ");
24. String studentName = scanner.nextLine();
25. if (scores.containsKey(studentName)) {
26. double score = scores.get(studentName);
27. System.***out***.println("La nota de " + studentName + " es: " + score);
28. } else {
29. System.***out***.println("El estudiante no se encuentra en el registro.");
30. }
31. } else if (opcion == 3) {
32. System.***out***.println("Saliendo del programa.");
33. break;
34. } else {
35. System.***out***.println("Opción no válida. Por favor, elija una opción válida.");
36. }
37. }
38. }
39. }