

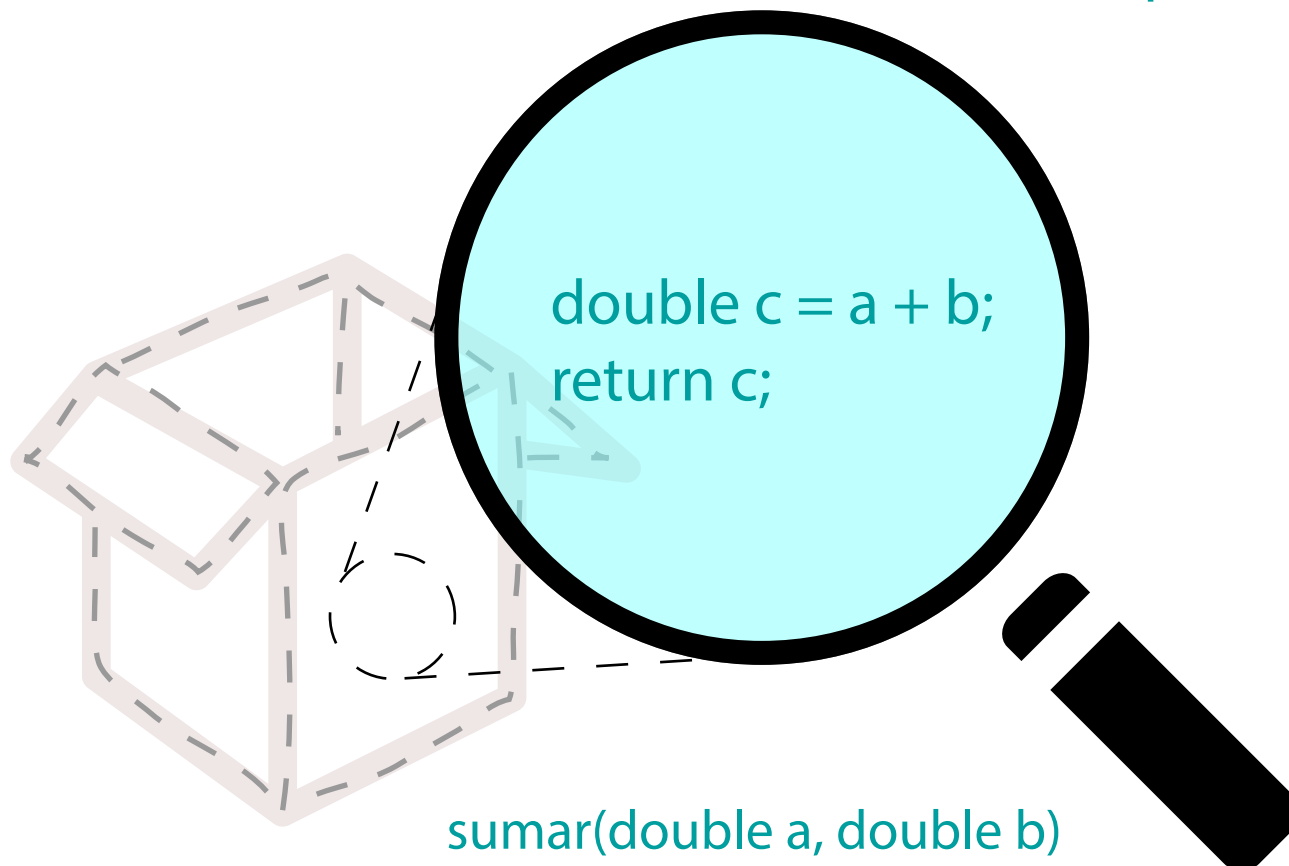


PROGRA**AMANDO**
holamundo.co



Caja blanca

Es una función de la cual se tiene conocimiento, sobre lo que pasa internamente. Son las funciones creadas por el programador.





¿Cuándo no usar retorno?

La salida se realiza implícitamente, por ejemplo una impresión en pantalla.

En java el punto de partida de un programa es una función void, ya que no es necesario retornar un valor por fuera del programa

Cuando se realiza una asignación de un valor (OOP)

Cuando se envia un valor por referencia a la función (próximamente estudiaremos como se hace esto)



```
public static void calcularPromedio(double datos[])
{
    double ans = 0;
    if(datos.length > 0)
    {
        for(double numero:datos)
        {
            ans+= numero;
        }
        ans = ans/datos.length;
        System.out.println("El promedio es "+ans);
    }
    else
    {
        System.out.println("No hay números");
    }
}
```

Retorno implícito

```
calcularPromedio(new double[]{3,4,5,6,7,8});
```



Casting implícito de parámetro

Teniendo en cuenta la compatibilidad presente entre algunos tipos de datos, es posible realizar casting implícito al realizar el retorno de una función, esto también se puede aprovechar con los parámetros de una función.

```
public static double dividir(double a, double b)
{
    return a / b;
}
...
int n1 = 3, n2 = 5;
if(n2 > 0){
    System.out.println("La división es" + dividir(n1,n2));
}
```



Casting implícito de retorno

```
public static float dividir(int a, int b)
{
    return a / b;
}
```

...

```
double resultado;
int n1 = 3, n2 = 5;
if(n2 > 0)
{
    resultado = dividir(n1,n2);
    System.out.println("La división es" + resultado);
}
```



Recursividad

Se presenta cuando una función se llama a si misma, se debe tener precacución en garantizar que la función termina.

Siempre debe tener tipo de retorno diferente de void.

Consume muchos recursos si se realizan muchas invocaciones, requiere de un algoritmo optimizado y probado, pues puede generar loops infinitos.



Factorial recursivo

```
public static long factorial(long n)
{
    if(n>0)
    {
        return factorial(n-1)*n;
    }
    else
    {
        return 1;
    }
}
...
```

Autoinvocación

Críterio de finalización

```
System.out.println(factorial(14));
```




Recomendaciones

Evitar tener muchos parámetros, lo recomendable es que no sean más de 2 datos.

El nombre no debe tener espacios, no debe ser un número o empezar por un número, se recomienda separar con underline.

En java los nombres de las funciones deben seguir la notación camel case, deben empezar en minúscula y si hay una frase compuesta cada palabra empieza con mayúscula. ej: pagarFactura.

El nombre debe ser nemotécnico, es decir que tenga implícito lo que hace internamente la función. ej: calcularPromedio.

Crear una función especializada, divide y vencerás.



Ejercicios

Realizar una función que:

Calcule el factorial de un número, tener en cuenta condiciones especiales y evitar que se presente desbordamiento.

Genere un número entero aleatorio entre n y m , donde $n < m$

Calcule la suma de los n primeros números primos

Calcule el horoscopo de una persona a partir del ingreso del día y mes de nacimiento

Convierta una frase a minúscula y cada palabra con mayúscula inicial y asigne un punto al final, si no lo tiene.