CURSO
BACKEND 1

Subprogramas

Material de lectura

OBJETIVOS DE LA GUÍA

En esta guía aprenderemos a:

- Definir y utilizar funciones, manejo de retornos
- Definir y utilizar de procedimientos
- Realizar llamados de funciones y procedimientos

egg | Argentina programa 4.0

Subprogramas

Un método muy útil para solucionar un problema complejo es dividirlo en subproblemas —problemas más sencillos— y a continuación dividir estos subproblemas en otros más simples, hasta que los problemas más pequeños sean fáciles de resolver. Esta técnica de dividir el problema principal en subproblemas se suele denominar “divide y vencerás”.

El problema principal se soluciona por el correspondiente programa o algoritmo principal, mientras que la solución de los subproblemas será a través de subprogramas, conocidos como **procedimientos** o **funciones**. Un subprograma es un como un mini algoritmo, que recibe los *datos,* necesarios para realizar una tarea, desde el programa y devuelve los *resultados* de esa tarea.

Funciones

Las funciones o métodos son un conjunto de líneas de código (instrucciones), encapsulados en un bloque, usualmente según los parámetros definidos en la función, esta recibe argumentos, cuyos valores se utilizan para efectuar operaciones y adicionalmente retornan un valor. En otras palabras, una función según sus parámetros puede recibir argumentos (algunas no reciben nada), hace uso de dichos valores recibidos como sea necesario y retorna un valor usando la instrucción return, si no retorna es otro tipo de función. Los tipos que pueden usarse en la función son: int, doble, long, boolean, String y char.

A estas funciones les vamos a asignar un tipo de acceso y un modificador. Estos dos conceptos los vamos a ver mejor más adelante, pero por ahora siempre vamos a crear las funciones con el acceso public y el modificador static. **Para saber más sobre estos dos temas, leer la explicación del método main.**

```
[acceso][modificador][tipo] nombreFuncion([tipo] nombreArgumento, …….){
      /*
      * Bloque de instrucciones
      */
    return valor;
}
```

Consejos acerca de return:

- Cualquier instrucción que se encuentre **después de la ejecución de return NO será ejecutada**. Es común encontrar funciones con múltiples sentencias return al interior de condicionales, pero una vez que el código ejecuta una sentencia return lo que haya de allí hacia abajo no se ejecutará.
- El tipo de valor que se retorna en una función debe coincidir con el del tipo declarado a la función, es decir si se declara int, el valor retornado debe ser un número entero.


```
  public static void main(String[] args) {

      Scanner leer = new Scanner(System.in);
      int num1 = 5;
      int num2 = 7;

      //Puedo invocar el retorno de esta funcion de esta manera
      System.out.println("La suma de ambos es: " + sumar(num1,
num2));

      int retorno = sumar(num1, num2);

      System.out.println("La suma de ambos es: " + retorno);
      }

      // Pero, recomendamos hacerlo de esta manera, ya que los
retornos deben alojarse en variables para su posterior uso

      public static int sumar(int num1, int num2) {
      int suma;
      suma = num1 + num2;
      return suma;
      }
```

¡MANOS A LA OBRA!

Ejercicio 11

Escribir un programa que procese una secuencia de caracteres ingresada por teclado y terminada en punto, y luego codifique la palabra o frase ingresada de la siguiente manera: cada vocal se reemplaza por el carácter que se indica en la tabla y el resto de los caracteres (incluyendo a las vocales acentuadas) se mantienen sin cambios.

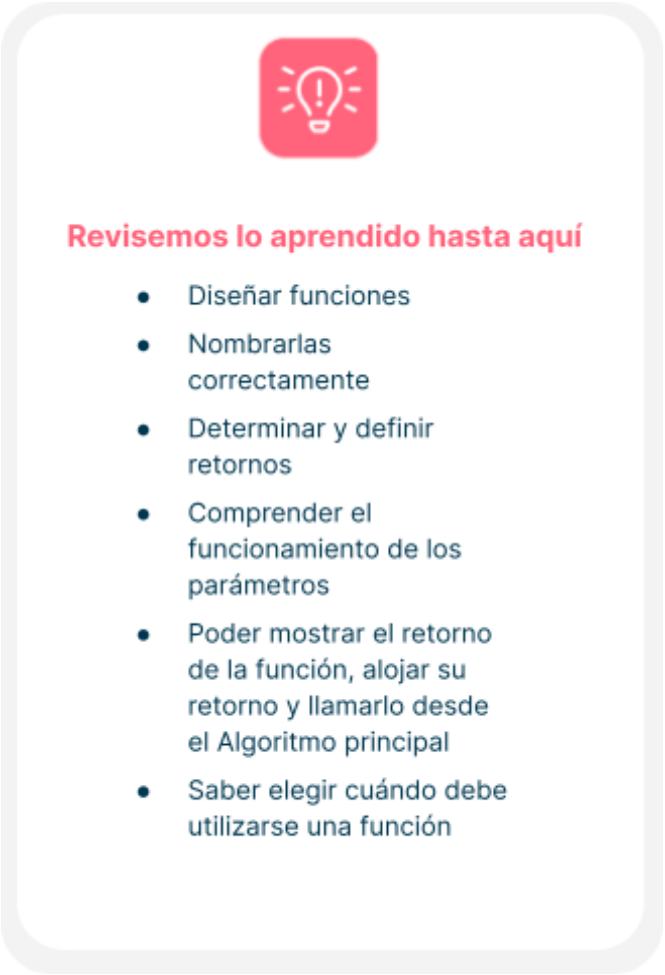
a	e	i	o	u
@	#	$	%	*

Realice un subprograma que reciba una secuencia de caracteres y retorne la codificación correspondiente. Utilice la estructura “según” para la transformación.

Por ejemplo, si el usuario ingresa: Ayer, lunes, salimos a las once y 10.

La salida del programa debería ser: @y#r, l*n#s, s@l$m%s @ l@s %nc# y 10.

Para realizar este ejemplo, deberás investigar el método concat de la clase String. Puedes encontrar estos ejemplos al final de la guía.



Procedimientos (Funciones sin retorno)

Los procedimientos son básicamente un conjunto de instrucciones que se ejecutan sin retornar ningún valor, hay quienes dicen que un procedimiento no recibe valores o argumentos, sin embargo, en la definición no hay nada que se lo impida. En el contexto de Java un procedimiento es básicamente una función cuyo tipo de retorno es void, los que indica que devuelven ningún resultado.

```
[acceso][modificador] void nombreFuncion([tipo] nombreArgumento){

	/*

	* Bloque de instrucciones

	*/

}
```

Acerca de los argumentos o parámetros:

- Hay algunos detalles respecto a los argumentos de un método, veamos:
- Una función, un método o un procedimiento pueden tener una cantidad infinita de parámetros, es decir pueden tener cero, uno, tres, diez, cien o más parámetros. Aunque habitualmente no suelen tener más de 4 o 5.
- Si una función tiene más de un parámetro cada uno de ellos debe ir separado por una coma.
- Los argumentos de una función también tienen un tipo y un nombre que los identifica. El tipo del argumento puede ser cualquiera y no tiene relación con el tipo del método.
- Al recibir un argumento nada nos obliga a hacer uso de éste al interior del método, sin embargo, para que recibirlo si no lo vamos a usar.
- En java los argumentos que sean variables de tipos primitivos (int, double, char, etc.) se van a pasar por valor, mientras que los objetos (String, Integer, etc.) y los arreglos se van a pasar por referencia.
  Nota: el concepto de objetos lo vamos a ver más adelante.

¿NECESITAS UN **EJEMPLO**?

```
public static void main(String[] args) {

      String nombre = "Mariano";

      int edad = 29;

      mostrarInfo(nombre, edad);
      }


      public static void mostrarInfo(String nombre, int edad) {

      System.out.println("El nombre del usuario es: " + nombre + "y
su edad:" + edad);

      }
```

¡MANOS A LA OBRA!

Ejercicio 12

Crea un procedimiento *EsMultiplo* que reciba los dos números pasados por el usuario, validando que el primer número múltiplo del segundo e imprima si el primer número es múltiplo del segundo, sino informe que no lo son.

En las guías que vienen luego de esta, usaremos MUCHAS funciones y procedimientos. Recuerda que utilizaremos los procedimientos ÚNICAMENTE cuando el objetivo del mismo es MOSTRAR información. Caso contrario debe retornar el resultado de la operación que realice.

Revisemos lo aprendido hasta aquí

- Diseñar procedimientos
- Nombrarlos correctamente
- Comprender el funcionamiento de los parámetros
- Llamar al procedimiento desde el algoritmo principal
- Saber elegir cuándo debe utilizarse un procedimiento.