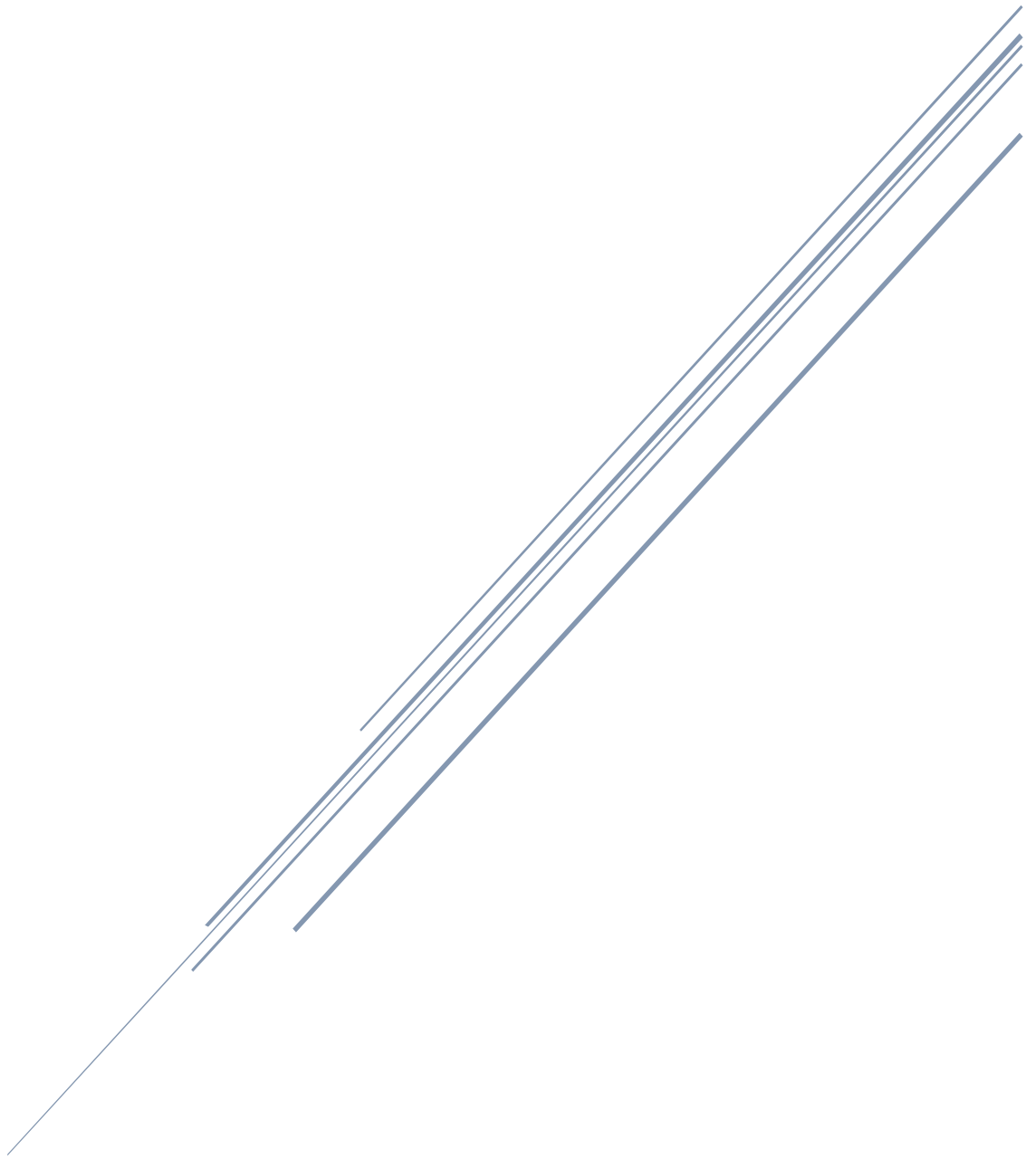


—
METODOS
PROGRAMACIÓN



[Escuela]
[Título del curso]

Contenido

Métodos y Funciones	2
MÉTODOS.....	2
FUNCIÓN	2
EJEMPLO DE MÉTODOS Y FUNCIONES	2
USO DE LOS MÉTODOS Y FUNCIONES.....	2
Ejercicios.....	3
Ej- Q601	3
Ej-Q602.....	4

Métodos y Funciones

MÉTODOS

Es un conjunto de código reutilizable del cual se puede hacer un uso continuo que se encuentra disponible en cualquier momento deseado dentro de la aplicación.

FUNCIÓN

Las funciones son similares a los métodos la diferencia es que estas retornan un resultado lo cual puede ser almacenado dentro de una variable o impreso directamente en pantalla, mientras que los métodos no retornan ningún tipo de dato pero a si mismo podrían meterse variables al mismo método lo cual podría usarse como un tipo de retorno aunque no sería lo mismo.

EJEMPLO DE MÉTODOS Y FUNCIONES

Veamos un ejemplo, primero creamos un método que realice la suma de dos números, el código sería el siguiente:

```
private void MetodoSuma(float numero1, float numero2){  
    float Suma=numero1+numero2;  
    System.out.println(Suma);  
}
```

Ahora creamos una función que realice lo mismo pero que retorne el resultado para almacenarlo dentro de una variable, el código lo vemos seguidamente:

```
private float FuncionSuma(float numero1, float numero2){  
    float Suma=numero1+numero2;  
    return Suma;  
}
```

USO DE LOS MÉTODOS Y FUNCIONES

En el primero caso haremos el uso del método cargándole con dos datos fijos. Creamos los dos datos de tipo float y luego instanciamos al método de la siguiente forma:

```
float Num1=7, Num2=4;  
MetodosFunciones C=new MetodosFunciones();  
C.MetodoSuma(Num1, Num2);
```

Primero debemos instanciar a la clase, en este caso a MetodosFunciones porque es el nombre de la clase principal

```
MetodosFunciones C=new MetodosFunciones();
```

Luego haciendo uso de su variable C instanciamos al método o función que deseemos llamar como se ve seguidamente

```
C.MetodoSuma(Num1, Num2);
```

Habiendo realizado el código hasta este punto ya debería imprimir la suma de ambos números, entonces al ejecutar el código debería mostrarnos el resultado de la suma de ambos números.

Ahora para poder hacer uso de la función sería lo mismo, pero en esta oportunidad podríamos hacer uso de la clase ya instanciada con su variable C instanciando directamente a la función de la siguiente forma:

```
Sumax=C.FuncionSuma(Num1, Num2);
```

En este caso asignamos a una variable Sumax la función ya que esta función retornara un resultado y haría falta hacer impresión en pantalla de este dato lo cual podemos hacerlo de la siguiente forma

```
System.out.println(Sumax);
```

También podría realizarse la impresión directamente sin necesidad de almacenarlo en una variable como se muestra seguidamente lo cual daría el mismo resultado

```
System.out.println(C.FuncionSuma(Num1, Num2));
```

Ejercicios

Ej- Q601

Crea un archivo "XXX_misMetodosMain.java" que implemento un menú como se muestra en la imagen.

Pulsando la letra correspondiente a cada apartado, el programa principal debe llamar a la función/procedimiento (método) correspondiente para que así se ejecute y cumpla con su cometido.

Este menú debe estar implantado dentro de un

```
do{  
    } while ();
```

Que terminará cuando pulsemos z.

Notas:

Comenzar por diseñar este primer apartado y que muestre en pantalla parte del menú que se muestra. Después, con un switch, realizar las llamadas a los métodos oportunos.

Selecciona una opción:

1.- Operaciones

- a.- Suma
- b.- Resta
- c.- Multiplicación
- d.- División

2.- Otras operaciones

- e.- Números Primos
- f.- Divisores
- g.- Es Números Perfectos
- h.- Imprimir Números Perfectos
- i.- De Binaria a Decimal
- j.- De Decimal a Binario
- k.- Matriz Identidad

3.- NIF

- l.- Letra NIF
- m.- NIF

4.- Textos

- n.- Contar Letras
- o.- Es Palíndromo

5.- Juegos

- p.- Ahorcado
- q.- Tres En Raya
- r.- Conecta Cuatro
- s.- Ajedrez (jjj :)

6.- Salir

- z.- Salir

Ej-Q602

Lista de métodos a implementar:

public float **fSuma**(float numero1, float numero2)

Devuelve la suma de dos números, (la función no solicita datos por pantalla)

public float **fResta**(float numero1, float numero2)

Devuelve la resta de dos números, (la función no solicita datos por pantalla)

public float **fMultiplicacion**(float numero1, float numero2)

Devuelve la resta de dos números, (la función no solicita datos por pantalla)

public float **fDivision**(float numero1, float numero2)

Devuelve la resta de dos números, (la función no solicita datos por pantalla)

Validar división por 0

Public void **mPrimos**(int numMax)

Llama al método “números primos” pasando el máximo de números a representar (el método no solicita datos por pantalla, pero si imprime los datos)

public void XXX_ **divisores**(int num)

Imprime los divisores del número que se le pasa por parámetros

public boolean XXX_ **esNumeroPerfecto**(int num)

Devuelve verdadero o falso en función de si el número que se le pasa por parámetros es “Numero Perfecto”

public void XXX_ **imprimeNumerosPerfecto**(int num)

Imprime una lista de los “Numero Perfecto” que existen hasta “num” (el método no solicita datos por pantalla)

Public char XXX_ **fLetraDNI**(int DNI)

Devolverá la letra resultante del DNI (la función no solicita datos por pantalla)

Public String XXX_ **fNIF**(int DNI)

Devolverá NIF completo resultante del DNI (la función no solicita datos por pantalla)

public String XXX_ **binarioADecima**(String binario)

Devuelve el decimal del número binario (la función no solicita datos por pantalla)

public String XXX_ **decimalABinario**(String binario)

Devuelve el decimal del número binario (la función no solicita datos por pantalla)

public void XXX_ **cuentaLetras**(String str)

Imprime el número de todas las letra y símbolos de un string (el método no solicita datos por pantalla pero si los imprime)

public boolean **palindromo**(String str)

Devuelve verdadero o falso indicando si una frase es palíndromo (la función no solicita datos por pantalla)

*public void **imprimirMatrizIdentidad**(int dimensión)*

Imprime la matriz identidad (ejercicio del examen) pasándole por parámetros la dimensión de esta.

*public void XXX_ **AhorcadoArray**()*

Inicia el juego del ahorcado de forma independiente al programa principal

*public void XXX_ **tresEnRaya**()*

Inicia el juego del ahorcado de forma independiente al programa principal

Nota aclaratoria:

Cada alumno debe sustituir la triple "XXX" por las siglas del nombre y apellido, como hacemos en todos los programas.

Los nombres de los métodos son orientativos, cada uno usará el nombre que ya tenga por defecto, modificando así lo menos posible de sus propios ejercicios.