

ACTIVIDAD

Funciones y librerías

Administración de Sistemas Informáticos y Redes

Programación I



Linkia FP



Actividad

Funciones y librerías

Objetivos

- Modularizar correctamente los programas.
- Crear y utilizar funciones.
- Depurar y comentar los programas.
- Control de errores.
- Entender, crear y utilizar librerías.

¿Cómo lo hago?

1. Rellena los datos que se piden en la tabla “Antes de empezar”.
2. Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc.
3. Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar.
4. Entrega un zip que contenga todos los archivos. java que has creado. Para poder aprobar un ejercicio, éste debe poder ejecutarse sin errores.
5. Recuerda nombrar el archivo zip siguiendo estas indicaciones:
 - Ciclo_Módulo o crédito_Tema_ACT_número actividad_Nombre y apellido
 - Ejemplo: AF_M01_T01_ACT_01_Maria Garcia

Antes de empezar...

Nombre	Juan
Apellidos	Rodríguez García – San Pedro
Módulo/Crédito	M03 Programación
UF (solo ciclos LOE)	UF02
Título de la actividad	Actividad03

Se debe entregar un zip que contenga todos los archivos .java que has creado. Para poder aprobar un ejercicio, éste debe poder ejecutarse sin errores. Crea los archivos .java dentro de una carpeta de nombre actividad03

1. Comenta cada función y clase indicando su objetivo
2. En el package **ACT03_ASIX_asix.IntroDatos** crea y programa **Filtro.java**: crea las funciones `int pideEntero(String pregunta)` y `double pideDouble (String pregunta)` encargadas de mostrar por consola la pregunta pasada como parámetro, pedir un número por consola (si el usuario no introduce un valor correcto ha de volver a pedir el valor hasta que el usuario introduzca un valor correcto) y retornar el número introducido como un `int` o `double`.

```
package IntroDatos;

import java.io.*;

// Clase Filtro
// Esta clase tiene como objetivo proporcionar dos funciones estáticas para obtener
// números enteros y decimales
// del usuario, garantizando que los números ingresados estén dentro de un rango
// específico.
public class Filtro {

    // Función pideEntero
    // Objetivo: Pedir un número entero al usuario y asegurar que se encuentre dentro
    // del rango [min, max]
    // Parámetros:
    // - mensaje: Mensaje que se mostrará al usuario antes de solicitar el número
    // - min: Límite inferior del rango
    // - max: Límite superior del rango
    public static int pideEntero(String mensaje, int min, int max) throws Exception {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(isr);
        boolean correcto;
        int num = 0;
        do {
            correcto = true;
            try {
                do {
                    System.out.println(mensaje);
                    num = Integer.parseInt(br.readLine());
                } while (num < min || num > max);
            } catch (Exception e) {
                System.out.println("Error en el formato numérico, vuelve a introducir
un valor.");
            }
        } while (!correcto);
        return num;
    }
}
```

```

        correcto = false;
    }
} while (correcto == false);
return num;
}

// Función pideDouble
// Objetivo: Pedir un número decimal al usuario y asegurar que se encuentre
dentro del rango [min, max]
// Parámetros:
// - mensaje: Mensaje que se mostrará al usuario antes de solicitar el número
// - min: Límite inferior del rango
// - max: Límite superior del rango
public static double pideDouble(String mensaje, double min, double max) throws
Exception {
    InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(isr);
    boolean correcto;
    double num = 0;
    do {
        correcto = true;
        try {
            do {
                System.out.println(mensaje);
                String valor = br.readLine();
                num = Double.parseDouble(valor);
            } while (num < min || num > max);
        } catch (Exception e) {
            System.out.println("Error en el formato numérico, vuelve a introducir
un valor.");
            correcto = false;
        }
    } while (correcto == false);
    return num;
}
}

```

3. En el package **ACT03_ASIX_asix.main** crea **Ejercicio01.java**: añade el código que muestre por consola un menú para ejecutar cualquiera de las funciones definidas en el resto del documento. Ten en cuenta que se debe pedir al usuario el número de función a ejecutar mediante la función `pideEntero` creada anteriormente. El programa solo debe finalizar cuando el usuario indique que no quiere realizar ninguna otra operación.

```
4. package main;
5.
6. import java.io.*;
7. import IntroDatos.*;
8. import operaciones.logicas.*;
9. import operaciones.aritmeticas.*;
10. import operaciones.*;
11.
12. //Clase Ejercicio01
13. //Esta clase tiene como objetivo generar un menú con diferentes opciones y que el
    la función main las vaya desarrollando.
14. public class Ejercicio01 {
15.
16.     // Función main
17.     // Objetivo: Función principal que ejecuta la opción seleccionada de la
    función menu.
18.     // Parámetros:
19.     // - Cada una de las diferentes case.
20.     public static void main(String[] args) throws Exception {
21.         int op = 0; // Declaramos una variable entera llamada "op" con valor
    inicial de 0.
22.         do { // Comenzamos un bucle do-while que se ejecutará al menos una vez.
23.             op = menu(); // Llamamos al método "menu" que devuelve la opción
    elegida por el usuario y lo guardamos en "op".
24.             switch(op) { // Comenzamos un switch-case para ejecutar diferentes
    bloques de código según el valor de "op".
25.                 case 0:
26.                     System.out.println("Hasta luego Francesc!!!!"); // Imprimimos
    un mensaje de despedida.
27.                     break; // Salimos del case.
28.                 case 1:
29.                     double pi = Calculos1.muestraPi(); // Declaramos una variable
    double llamada "pi" y le asignamos el valor que devuelve el método "muestraPi" de
    la clase "Calculos1".
30.                     System.out.println("Valor de Pi: " + String.format("%.6f",
    pi)); // Imprimimos por pantalla el valor de "pi" reducido a 6 decimales.
31.                     break; // Salimos del case.
32.                 case 2:
33.                     double num = Filtro.pideDouble("Introduce un número decimal
    para redondear: ", 1, Integer.MAX_VALUE); // Declaramos una variable double
```

```

    llamada "num" y le asignamos el valor que devuelve el método "pideDouble" de la
    clase "Filtro" al pasarle los argumentos indicados.
34.         int ValorRedondeado = Calculos1.muestraValorRedondeado(num);
    // Declaramos una variable entera llamada "ValorRedondeado" y le asignamos el
    valor que devuelve el método "muestraValorRedondeado" de la clase "Calculos1" al
    pasarle "num" como argumento.
35.         System.out.println("El valor redondeado de " + num + " es: " +
    ValorRedondeado); // Imprimimos por pantalla el valor de "num" y
    "ValorRedondeado".
36.         break; // Salimos del case.
37.         case 3:
38.             int n2 = Filtro.pideEntero("Introduce un número y saco un
    número aleatorio entre ese número y el 0: ", 1, Integer.MAX_VALUE); // Declaramos
    una variable entera llamada "n2" y le asignamos el valor que devuelve el método
    "pideEntero" de la clase "Filtro" al pasarle los argumentos indicados.
39.             int ValorAleatorio = Calculos1.muestraValorAleatorio(n2); //
    Declaramos una variable entera llamada "ValorAleatorio" y le asignamos el valor
    que devuelve el método "muestraValorAleatorio" de la clase "Calculos1" al pasarle
    "n2" como argumento.
40.             System.out.println("El valor aleatorio entre 0 y " + n2 + " Es:
    " + ValorAleatorio); // Imprimimos por pantalla el valor de "n2" y
    "ValorAleatorio".
41.             break; // Salimos del case.
42.             case 4:
43.                 double n1 = Filtro.pideDouble("Introduce un ángulo entre 0º y
    360º: ", 0, 360); // Declaramos una variable double llamada "n1" y le asignamos el
    valor que devuelve el método "pideDouble" de la clase "Filtro" al pasarle los
    argumentos indicados.
44.                 double seno = Operacionesmates.muestraseno(n1); // Declaramos
    una variable double llamada "seno" y le asignamos el valor que devuelve el método
    "muestraseno" de la clase "Operacionesmates" al pasarle "n1" como argumento.
45.                 System.out.println("El seno de " + n1 + " es: " +
    String.format("%.2f", seno) + "."); // Imprimimos por pantalla el "seno" reducido
    a 2 decimales.
46.                 break; // Salimos del case.
47.                 case 5:
48.                     double n = Filtro.pideDouble("Introduce un número que no sea
    negativo: ", 1, Integer.MAX_VALUE); // Declaramos una variable double llamada "n"
    y le asignamos el valor que devuelve el método "pideDouble" de la clase "Filtro"
    al pasarle los argumentos indicados.
49.                     double raiz = Operacionesmates.calcularaiz(n); // Declaramos
    una variable double llamada "raiz" y le asignamos el valor que devuelve el método
    "calcularaiz" de la clase "Operacionesmates" al pasarle "n" como argumento.
50.                     System.out.println("La raíz de " + n + " es : " +
    String.format("%.4f", raiz) + "."); // Imprimimos por pantalla el valor "raiz"
    reducido a 4 decimales.

```

```

51.         break; // Salimos del case.
52.         case 6:
53.             double num1 = Filtro.pideDouble("Introduce el primer número:
", Integer.MIN_VALUE, Integer.MAX_VALUE); // Declaramos una variable double
llamada "num1" y le asignamos el valor que devuelve el método "pideDouble" de la
clase "Filtro" al pasarle los argumentos indicados.
54.             double num2 = Filtro.pideDouble("Introduce el segundo número:
", Integer.MIN_VALUE, Integer.MAX_VALUE); // Declaramos una variable double
llamada "num2" y le asignamos el valor que devuelve el método "pideDouble" de la
clase "Filtro" al pasarle los argumentos indicados.
55.             double num3 = Filtro.pideDouble("Introduce el tercer número:
", Integer.MIN_VALUE, Integer.MAX_VALUE); // Declaramos una variable double
llamada "num3" y le asignamos el valor que devuelve el método "pideDouble" de la
clase "Filtro" al pasarle los argumentos indicados.
56.             double media = Operacionespropias.media3Numeros(num1, num2,
num3); // Declaramos una variable double llamada "media" y le asignamos el valor
que devuelve el método "media3Numeros" de la clase "Operacionespropias" al pasarle
los argumentos "num1", "num2" y "num3".
57.             System.out.println("La media de " + num1 + ", " + num2 + ", " +
"y " + num3 + " es: " + String.format("%.3f", media) + "."); // Imprimimos por
pantalla el valor de "media" y le decimos que solo tenga 3 decimales.
58.             break; // Salimos del case.
59.             case 7:
60.                 int numb = Filtro.pideEntero("Introduce un número del 1 al 7 y
te digo que día de la semana es!: ", 1, 7); // Declaramos una variable entera
llamada "numb" y le asignamos el valor que devuelve el método "pideEntero" de la
clase "Filtro" al pasarle los argumentos indicados.
61.                 String dia = Operacionespropias.muestraNombreDia(numb); //
Declaramos una variable String llamada "dia" y le asignamos el valor que devuelve
el método "muestraNombreDia" de la clase "Operacionespropias" al pasarle "numb"
como argumento.
62.                 System.out.println("El día " + numb + " es " + dia + "."); //
Imprimimos por pantalla el valor de "numb" y "dia".
63.                 break; // Salimos del case.
64.             }
65.         } while (op != 0); // Mientras el número introducido no sea 0 el programa
sigue funcionando. Cuando sea 0 cierra.
66.     }
67.
68.
69.     // Función menu
70.     // Objetivo: Generar un menú donde el usuario puede ver las diferentes
opciones.
71.     // Parámetros:
72.     // - Un número int mediante la variable op que ha de estar comprendido entre
en 0 y el 7

```



```

73.     public static int menu() throws Exception {
74.         InputStreamReader isr = new InputStreamReader(System.in);
75.         BufferedReader br = new BufferedReader(isr);
76.         System.out.println("0-Salir.");
77.         System.out.println("1-Muestra Pi.");
78.         System.out.println("2-Muestra el valor redondeado del valor
    introducido.");
79.         System.out.println("3-Muestra un valor aleatorio entre 0 y el valor
    introducido.");
80.         System.out.println("4-Muestra el seno.");
81.         System.out.println("5-Muestra la raíz cuadrada.");
82.         System.out.println("6-Media aritmética de 3 numeros decimales.");
83.         System.out.println("7-Pide número entero e indica que día de la semana
    es.");
84.         int op = Filtro.pideEntero("Opción: ", 0, 7);
85.         return op;
86.     }
87. }

```

88. En el package **ACT03_ASIX_asix.operaciones** crea y programa **Calculos1.java** con las funciones:

```

//Clase Calculos1
//Esta clase tiene como objetivo proporcionar 3 funciones para obtener diferentes
datos como el número pi, redondear un valor o generar un número aleatorio.
public class Calculos1 {

```

88.1. **muestraPI**: que simplemente muestre por consola el valor del número PI definido en la clase Math

```

// Función muestraPi
// Objetivo: Mostrar el número Pi
public static double muestraPi() {
    return Math.PI;
}

```

88.2. **muestraValorRedondeado**: que pida un número mediante pideDouble y muestre por consola su valor redondeado.

```

// Función muestraValorRedondeado
// Objetivo: Pedir un número decimal al usuario y redondearlo.
// Parámetros:
// - num: Número a redondear.
public static int muestraValorRedondeado(double num) {
    return (int) Math.round(num);
}

```

88.3. **muestraValorAleatorio** que pida un número mediante pideEntero y muestre por consola un número aleatorio entre 0 y el número introducido.

```

// Función muestraValorAleatorio
// Objetivo: Genera un número aleatorio entre el 0 y el número pasado como
parametro. Lo convierte en entero y lo devuelve.

```

```
//Parámetros:
// - num: Número para generar el aleatorio entre 0 y este paámetro
public static int muestraValorAleatorio(int num) {
    return (int) (Math.random() * num);
}
}
```

89. En el package **ACT03_ASIX_asix.operaciones.aritmeticas** crea y programa **Operacionesmates.java** con las funciones:

```
//Clase Operacionesmates
//Esta clase tiene como objetivo proporcionar dos funciones , una que mediante un
ángulo dado por el usuario genere el seno de ese ángulo y otra
//función que calcula la raid cuadrada de un número también dado por el usuario.
public class Operacionesmates{
```

89.1. **muestraseno(double)** que muestre el seno del parámetro.

```
// Función muestraseno
// Objetivo: Pedir un número un número al usuario entre 0 y 360 simulando los °
de una circunderencia,
//luego lo pasaa a radianes y realiza el cálculo del seno.
// Parámetros:
// - num: Valor entre 0° y 360°.
public static double muestraseno(double num) {
    double radianes = Math.toRadians(num);
    return Math.sin(radianes);
}
```

89.2. **Calcularaiz(double)** que muestre la raiz cuadrada de un valor no negativo utilizando la función sqrt de la clase Math.

```
// Función calcularaiz
// Objetivo: Pedir un número al usuario y realizar su raiz cuadrada.
// Parámetros:
// - num: Valor double.
public static double calcularaiz (double num) {
    return Math.sqrt(num);
}
```

90. En el package **ACT03_ASIX_asix.operaciones.logicas** crea y programa **Operacionespropias.java** con las funciones:

```
//Clase Filtro
//Esta clase tiene como objetivo proporcionar dos funciones, una para obtener una
média aritmética de 3 valores dados por el usuario
//y otra para generar a través de un número entre en 1 y el 7 devolver el día de la
semana correspondiente.
public class Operacionespropias {
```

90.1. **media3Numeros:** debe llamar a una función que pida 3 números al usuario utilizando pideDouble y devolver la media aritmética. Muestra por consola la media.

```
public static double media3Numeros(double n1, double n2, double n3) {  
    double media = (n1 + n2 + n3) / 3;  
    return media;  
}
```

90.2. **muestraNombreDia:** debe pedir con pideEntero un número de día de la semana al usuario y indicar el nombre del día correspondiente (1 Lunes, 2 Martes...)

```
// Función muestraNombreDia  
// Objetivo: Pedir un numero entre en 1 y el 7 al usuario y generar el dia de la  
// semana correspondiente que lo devuelve a traves de la variable "nombreDia".  
// Parámetros:  
// - numb: int.  
public static String muestraNombreDia(int numb) {  
    String nombreDia;  
    switch(numb) {  
        case 1:  
            nombreDia = "Lunes";  
            break;  
        case 2:  
            nombreDia = "Martes";  
            break;  
        case 3:  
            nombreDia = "Miércoles";  
            break;  
        case 4:  
            nombreDia = "Jueves";  
            break;  
        case 5:  
            nombreDia = "Viernes";  
            break;  
        case 6:  
            nombreDia = "Sábado";  
            break;  
        case 7:  
            nombreDia = "Domingo";  
            break;  
        default:  
            nombreDia = "No es un número válido de día de la semana";  
            break;  
    }  
    return nombreDia;  
}
```