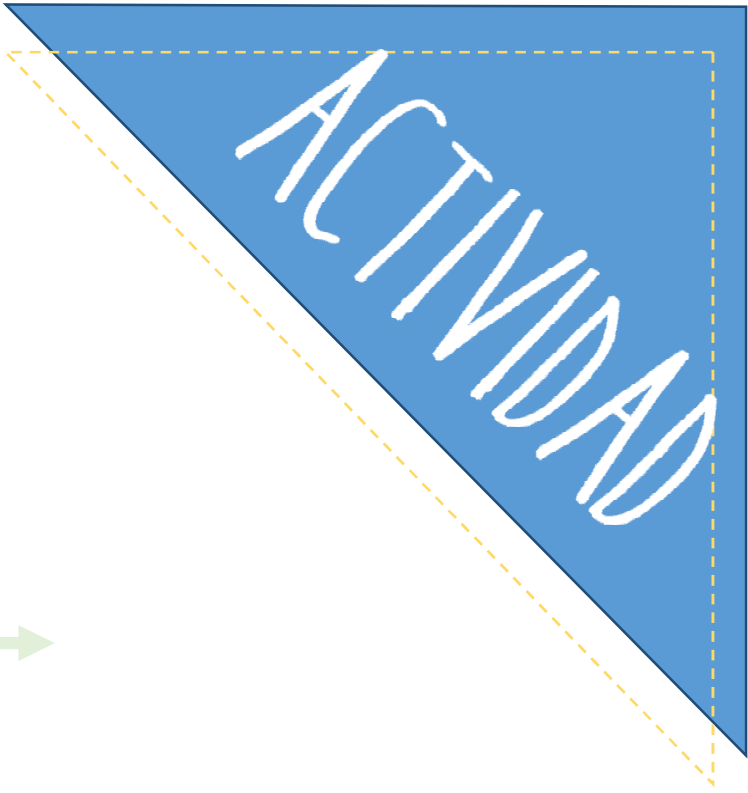

Funciones y librerías

Desarrollo de Aplicaciones Web/Desarrollo de Aplicaciones Multiplataforma

Programación

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	GARA
Apellidos	GONZÁLEZ SOSA
Módulo/Crédito	M03 PROGRAMACIÓN
UF (solo ciclos LOE)	UF2: DISEÑO MODULAR
Título de la actividad	FUNCIONES Y LIBRERIAS

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 **actividad03.introduceDatos** crea y programa **Pregunta.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.

   Primero hacemos la estructura de las carpetas. En las funciones pideEntero y pideDouble le entra por parámetros un String desde el main y la entrada del scanner. En mi programa principal creo una instancia del scanner y esa instancia se la paso a todos los submétodos para pedir datos al usuario. En el mismo programa principal la cierro. Lo hice así porque de tanto abrir y cerrar la entrada se me estaba trabando y no me recogia bien algunos valores y así sí me funcionó. Le ponemos la función do/while y try/catch para tratar posibles errores.

3. En el package **actividad03.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.

   En el main creo el menu y lo muestro por consola. También con un do/while hago que se repita hasta que el usuario quiera salir con la opción 0. Importo las clases necesarias para poder acceder a los otros métodos.

4. En el package**actividad03.operaciones** crea y programa **OperacionesLinkia.java** con las funciones:
   4.1. **convertirLinkiaCoins**() que pida mediante pideDouble un valor numérico con decimales y muestre el valor numérico introducido multiplicado por 1,5.

   Creo la clase y la función referida, le entra como parámetro solo la entrada del scanner. Llama al submetodo pideDouble y recoge la variable introducida por el usuario. Le enviamos la pregunta a pideDouble y recibimos la variable tipo double valorNumerico, aplicamos la operación (*1,5) y se muestra por consola el resultado.

5. En el package **actividad03.operaciones** crea y programa **Valores.java** con las funciones:

5.1. **muestraPI** : que simplemente muestre por consola el valor del número PI definido en la clase Math
5.2. **muestraValorAbsoluto**: que pida un número mediante pideDouble y muestre por consola su valor absoluto.
5.3. **muestraValorAleatorio**: que pida un número mediante pideEntero y muestre por consola un número aleatorio entre 0 y el número introducido.

Creamos la clase y los 3 submetodos (muestraPI, muestraValorAbsoluto muestraValorAleatorio).
Importamos la librería de java Math para poder acceder a las fórmulas.
MuestaPI solo muestra por consola el valor de PI que recoge de la librería math.
muestraValorabsoluto y muestraValoraleatorio llaman a los métodos pideDouble y pideEntero respectivamente, y recoge la variable introducida por el usuario. Con ese valor ya recogido mostramos por consola el resultado utilizando la librería.

6. En el package **actividad03.operaciones.aritmeticas** crea y programa **Operaciones.java** con las funciones:
6.1. **muestraLogaritmo(double)**: que muestre el logaritmo en base E del parámetro.
6.2. **calculaPotencia(double,double)** que muestre el resultado de elevar el primer parámetro al segundo utilizando la función pow de la clase Math.

Creo las dos funciones en la clase Operaciones. Muestralogaritmo recibe del main una variable de tipo double. En el main se llama al método pideDouble se recoge la variable y se envia a muestraLogaritmo. Con la librería se calcula el log y se muestra por consola.
En el método calculaPotencial lo hacemos igual pero le entra por parámetros 2 double que se envían también desde el main en donde se recogen llamando a pideDouble.

7. En el package **actividad03.operaciones.geometricas** crea y programa **Operaciones.java** con las funciones:
7.1. **muestraSeno**: que llame a pideDouble para obtener un número con decimales y muestre su seno.
7.2. **muestraCoseno**: que llame a pideDouble para obtener un número con decimales y muestre su coseno.

Creo las dos funciones en otra clase Operaciones: muestraCoseno y muestraSeno.
Le llega por parámetros un String desde el main y la entrada del scanner. Llamamos al submetodo pideDouble y recogemos el resultado. Con la librería calculamos el seno y el coseno y se muestran por pantalla.
Al haber dos clases llamadas operaciones tuve que utilizar toda la ruta para llamarlos desde el main.

A continuación se muestra un ejemplo del arbol de packages resultante:

Linkia FP

ACTIVIDAD