**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?** |
| * Rellena los datos que se piden en la tabla “Antes de empezar”. * Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc. * Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar. * Entrega un zip que contenga todos los archivos. java que has creado. Para poder aprobar un ejercicio, éste debe poder ejecutarse sin errores. * 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 | Jon |
| Apellidos | Sanchez Eguia |
| Módulo/Crédito | 21-22\_1\_2\_DAW\_M03A\_Programación |
| UF (solo ciclos LOE) |  |
| Título de la actividad | Actividad 3. Uso de 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***

* **Comenta cada función y clase indicando su objetivo**
* 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.





* En el package **actividad03.main** crea **Ejercicio02.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.





**>Para mantener orden dentro del main, he relegado la función de imprimir menú a imprimirMenu(), y una vez introducida la opción deseada esta es mandada a opcionEleida que leugo llamará a las diferentes funciones. Estas dos son llamadas desde el main asi que supongo que es buena idea hacer esto. Dejarlo todo dentro del main podria ser una opción pero a futuro podria ser dificil de leer.**

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





**>He llamado 'euros' al valor introducido por el usuario ya que no se especifica en el ejercicio y me parecia procedente que éste valor tuviera un nombre.**

* En el package **actividad03.operaciones** crea y programa **Valores.java** con las funciones:
* **muestraPI** : que simplemente muestre por consola el valor del número PI definido en la clase Math
* **muestraValorAbsoluto**: que pida un número mediante pideDouble y muestre por consola su valor absoluto.
* **muestraValorAleatorio** : que pida un número mediante pideEntero y muestre por consola un número aleatorio entre 0 y el número introducido.





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





**>He llamado a la libreria Math.h desde el main por lo que no necesito declararla aquí**

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





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

