Se realiza la instalación de eclipse en fedora 30, la instalación de git 2.1 y se integra git a eclipse con un repositorio alojado en github. También se instala JavaFX

1 de agosto, 2019

**Para utilizar la librería math**

    Se debe tomar ciertas precauciones, tanto la naturaleza de las variable a operar cómo la variable que almacena los resultados operados. En el caso de **Math.round();** se necesita ingresar un número que sea **float** para que devuelva un valor **int** o ingresar un **double** para que retorne un valor **long**, de lo contrario no va a operar.

    Nota: Para operar con variables de tipo **float** se necesita agregar el sufijo **F** como ejemplo:

**float** numero1 = 5.35F

**Refundición; concepto**

    Para poder cambiar un tipo de variable a otra en java, se utiliza la refundición. Convertir un tipo de dato a otro

En este ejemplo, el método está retornando un valor de tipo **long** pero al ingresar **(int)** antes del método, se convierte a un tipo float.

Nota: Al realizar una operación, se puede almacenar operaciones

**Manipulación de String; clase String**

    Estamos haciendo referencia a una clase y no a un tipo de datos primitivos.

    Se utiliza para almacenar una cadena de caracteres

    La sintaxis es similar a la variable; **String** nombre  = “kevin”; donde nombre resulta ser un objeto(instancia, ejemplar) de una clase.

**Métodos más usados:**

* .length(); devuelve la longitud de una cadena de caracteres. Ej hola -> 4 caracteres
* .charAt(); devuelve la posición de un carácter dentro de una cadena de texto. Se coloca la posición deseada y devuelve el carácter.
* .substring(x,y); devuelve una subcadena dentro de la cadena, siendo **x** el carácter a partir del cual se extrae además de **y** el cual es el número de caracteres que se quiere extraer.
* .equals(cadena); devuelve true si dos cadenas son iguales o false si no lo son. Distingue entre mayúsculas y minúsculas. **nombre.equals(otro\_nombre);**

**API de Java; biblioteca de clases predefinidas**

El paquete java.lang aparece por defecto, no necesita importarse.

Si se desea utilizar otro tipo de paquete, debe importarse con la etiqueta **import** + el nombre del paquete (Ej java.util.\*;)

**import** java.util.\*;   // El asterisco indica que estamos importando todas las clases (Consume más espacio)

**import** java.util.Scanner; // Solo importa la clase Scanner

**Entradas y Salida de datos**

IGU → Interfaz Gráfica de usuario. Crear un formulario que permite ver e introducir datos.

* Scanner: permite introducir datos por medio de texto nextLine(); números enteros nextInt(); o números decimales nextDouble();
* JOptionPane: Se utiliza un un método estático llamado showInputDialog(). para ingresar datos. Un método estático implica poner el nombre de la clase antes del método.

Nota: Las clases no estáticas implican que los métodos tengan un objeto creado antes del método a utilizar

    Las clases contienen **métodos constructores**, los cuales son estados iniciales de la clase y poseen el mismo nombre de la clase. **La clase Scanner** posee un método constructor donde se especifica el nombre el medio por el cual se ingresa información. En el caso de la consola de comandos, System.out es salida de la terminal y System.in es entrada de la terminal

            Scanner entrada = **new** Scanner(System.**in**);

    Se tiene que especificar el tipo de datos que se le van a ingresar, con el nombre del objeto creado.

            String nombre = entrada.nextLine();