System.out.println(C + (C%10==0 ? " es múltiplo de 10 " : " no es múltiplo de 10 "));

/\* si se cumple que el resto de C dividido para 10 =0: imprimiría C + es múltiplo de 10, si no lo fuera imprimiría C+ no es múltiplo de 10.+/

Math.floor ();o algo así redondea el número…. buscarlo

Apuntes Java

* Para cerrar Scanner. Ej:

Scanner entrada= new Scanner (System.in);

Para cerrarlo, al final del código tienes que poner: entrada.close();

Video 11, manipulación de cadenas

* String no es un tipo primitivo (no es un tipo de dato como puede ser int o double), es una clase
* String minombre= “Jorge”;
* length () : devuelve la longitud de una cadena de caracteres… minombre.length()
* charAt(n): devuelve la posición de un carácter dentro de una cadena (las posiciones empiezan a contar desde 0)… minombre.charAt (2)= r
* substring (x,y): devuelve una subcadena dentro de la cadena, siendo X el carácter a partir del cual se extrae e Y el nº de caracteres que se quieren extraer.
* equals (cadena): devuelve true si dos cadenas que se comparan son iguales y false si no lo son. Distingue mayúsculas y minúsculas.
* equalsignoreCase (cadena): igual que el anterior pero sin tener en cuenta las mayúsculas y minúsculas.

Video 15, entrada y salida de datos 1

* **int** IntValue = (**int**) DoubleValue;🡪 para refundir un tipo double en tipo int
* String edad= JOptionPane.showInputDialog(“Introduce tu edad por favor”);

/\*Muestra una pantalla emergente que devuelve lo escrito en la misma en dato tipo \*/String

Si se quiere crear la variable y refundir al mismo tiempo sería así:

int i= (int)(Math.*random*()\*20);

* Cuando en la API de Java te pone que un método es estático, significa que tienes que escribir delante del mismo su clase seguida de un punto: parseInt()🡪 se utiliza para refundir una variable de tipo String a tipo int… al ser un método estático deberá estar precedido por su clase (Integer) seguida por un punto.

Ej: String edad= JOptionPane.showInputDialog(“Introduce tu edad por favor”);

/\*Esta variable nos devuelve un dato de tipo String y queremos que sea entero, por lo \*/tanto haremos lo siguiente.

int edad\_usuario= Integer.parseInt(“edad”);

/\*Creamos la variable de tipo entero edad\_usuario a partir de la variable de tipo String \*/edad

* String longitud="23.6";

double longitud\_cm= Double.parseDouble("23.6");

//Sería para pasar de tipo String a tipo double, como en la anterior.

* Clase Numberformat🡪 se utiliza para cambiar el número de decimales de un double por ejemplo:

System.out.printf(“%1.2f”, Math.PI\*(Math.pow(radio,2)));

/\* imprimiría dos decimales. Si fuera un 4 en vez de el 2, imprimiría 4 decimales\*/

Video 17, Condicionales II

* \n 🡪 dentro de una cadena de texto hace salto de línea. Ej: “Elige una opción: \n1: cuadrado \n2: círculo”
* switch (nombre de la variable){

case 1:

…..

break; // al ejecutarse un break se sale de switch, ya no ejecuta más cases

case2:

…..

break;

default: 🡪 // esto sería como el else de los if.

…..

}

* Math.pow(int,2); donde int es la variable que queramos elevar a un número y 2 el exponente. Si quisiéramos al cubo sería 3. Devuelve un double.

Video 18, TIPOS DE BUCLES

* Bucles indeterminados: no sabes cuantas veces se va a repetir el código que hay en su interior sin haber ejecutado el programa
  + while
  + do – while
* Bucles determinados: sabes, echando un vistazo al código, cuantas veces se va a repetir el código que hay en su interior sin necesidad de ejecutar el programa
  + for
  + for-each

Video 19, BUCLES II DO WHILE:

* do{

genero=JoptionPane.showInputDialog(“Introduce tu genero (M/H)”);

}while (genero.equalsIgnoreCase(“H”)==false && genero.equalsIgnoreCase (“M”)==false);

/\*de esta manera aunque no se cumpliera la condición del while la primera vez, se ejecutaría el código dentro del do y luego ya, para que se volviera a ejecutar más veces sí que tendría que cumplir los requisitos del while… realizaría el bucle constantemente hasta que insertara una h o una m.

Video 20, Bucles IV FOR:

* for (inicio del bucle; condición; contador del bucle){

}

EJ: for (int i=0; i<10; i++){

}

Video 22, Bucle:

* if (correo.charAt(i)==’@’){

// condición para cuando ‘i’ valga ‘@’ en la variable correo

}

Video 23, Arrays I:

* Como declarar una matriz/array:

int[] mimatriz= new int[10];

mimatriz[0]= 45;

mimatriz[1]= 5;

mimatriz[2]= 7;

mimatriz[3]= -4;

/\*los arrays pueden albergar datos de todo tipo, dependiendo del tipo de dato que queramos que sean, se creará de tipo entero, string, double… Al final se pone el número de datos que rtendrá \*/

* Los arrays empiezan a enumerar sus posiciones desde el 0.
* Otra forma de declarar la matriz más cómoda sería la siguiente:

int[] mimatriz={45,5,7,-4};

Video 24, Arrays II, for each:

* Para recorrer la matriz también se puede usar el método **for**:

Misma matriz que antes:

int[] mimatriz={45,5,7,-4};

for (int i=0, i<mimatriz.lenght,i++){

}

* O se puede usar el método **for each** (más simple):

for (int i: mimatriz){

}

* //Para que los datos del array los introduzca otro usuario se utilizaría el JOptionPane:

String [] países= new String [4];

for (int i=0; i<4; i++){

países[i]= JOptionPane.showInputDialog(“Introduce país” + (i+1));

}

* numeros [i]= (int) Math.round(Math.random()\*200);

// de esta manera introducirías los valores de la matriz de manera aleatoria

Video 25,Arrays III, Arrays bidimensionales:

Video 26,Arrays IV, Arrays bidimensionales II:

* //Forma de introducir datos matriz de dos dimensiones: (Fijarse en el ‘;’ del final)

int matriz [][]={

{5,13,14,8,19,7},

{6,3,1,8,64,3},

{16,13,4,83,1,7},

{25,33,124,38,14,67}

};

* /\*Como utilizar un bucle for each en un array bidimensional (el array de arriba por ejemplo):\*/

for (int [] i:matriz){

for(int j :i){

System.out.print(j);

}

{