

FUNDAMENTOS DE PROGRAMACIÓN EN JAVA



FUNDAMENTOS DE PROGRAMACIÓN EN JAVA



¿QUÉ VAMOS A VER?

- Ciclos en Java.
 - Ciclo While.
 - Ciclo Do-While.
 - Ciclo For.
 - Ciclo For-Each.





CICLOS EN JAVA

- También conocidos como bucles, los ciclos en Java nos permiten repetir una o varias instrucciones las veces que necesitemos.
- Nos permiten automatizar bloques de código para evitar repetir innecesariamente las instrucciones.
- Los diferentes bucles cumplen exactamente la misma función, pero cambia su sintaxis y forma de aplicar las condiciones.





CICLO WHILE

- Nos permite ejecutar un bloque de código mientras la condición de salida sea verdadera.
- Cuando la condición cambia a falsa, el bucle deja de ejecutarse.
- Es importante saber que este tipo de ciclo evalúa la condición al inicio, es decir que, si en primera instancia es falsa, no se va a ejecutar el bloque de código ni siquiera una vez.
- Tampoco podemos olvidar la necesidad de manejar manualmente el contador o sentencia evaluadora de la condición, para evitar la posibilidad de un ciclo infinito.





EJEMPLO CICLO WHILE

- Desarrolle un programa que permita imprimir los números del 0 al 9.

```
int x = 0;
int y = 10;

while(x < y) {
    System.out.println("Vuelta número: " + x);
    x++;
}
```





CICLO DO-WHILE

- Su forma de trabajar es muy similar a la del ciclo While.
- La gran diferencia es que su condición se evalúa al final, es decir, el bloque de código se va a ejecutar por lo menos una vez.
- También necesita que manejemos de manera manual el contador en caso de existir.





EJEMPLO CICLO DO-WHILE

- Desarrolle un programa que permita imprimir los números del 0 al 9.

```
int x = 0;
do {
    System.out.println("Vuelta número: " + x);
    x++;
} while (x < 10);
```





CICLO FOR

- Es necesario tener un valor inicial, valor final y opcionalmente podemos usar el aumento o decremento de nuestro contador.
 - **Inicialización:** Valor inicial de nuestro contador.
 - **Terminación:** Condición sobre la cual se evaluarán las repeticiones.
 - **Incremento:** Define la forma en que se va a incrementar o decrementar el valor declarado en la inicialización.





EJEMPLO CICLO FOR

- Desarrolle un programa que permita imprimir los números del 0 al 9.

```
for (int i = 0; i < 10; i++) {  
    System.out.println("Vuelta número: " + i);  
}
```





CICLO FOR-EACH

- A diferencia de los otros ciclos, se utiliza especialmente para recorrer colecciones o arreglos de múltiples elementos.
- Su sintaxis es mucho más amigable que la de los bucles clásicos.
- Por ahora es importante saber de su existencia y conocer un poco su sintaxis, puesto que lo utilizaremos cuando veamos colecciones de datos.

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
  
for (String e : cars) {  
    System.out.println(x: e);  
}
```





EJERCICIO

- Introduce un número por teclado y comprueba de que sea mayor o igual que cero.
- Si no lo es, se debe volver a pedir el ingreso por teclado.
- En caso de que la condición se cumpla, se debe mostrar el número por pantalla.





FUNCIÓN MATH.RANDOM()

- Primero, debemos importar la clase **Math** desde *java.lang.Math*;
- Con esto ya podemos comenzar a generar números al azar desde 0 a un número en específico, o aún mejor, dentro de un rango que nosotros estimemos conveniente.
- Importante, *Math.random()*; genera números de tipo **double**.
- También tengamos en cuenta que la clase **Math** tiene la característica de ayudarnos a la hora de querer redondear un número decimal.

```
double roundedN = Math.floor(rNumber * 100) / 100;
```



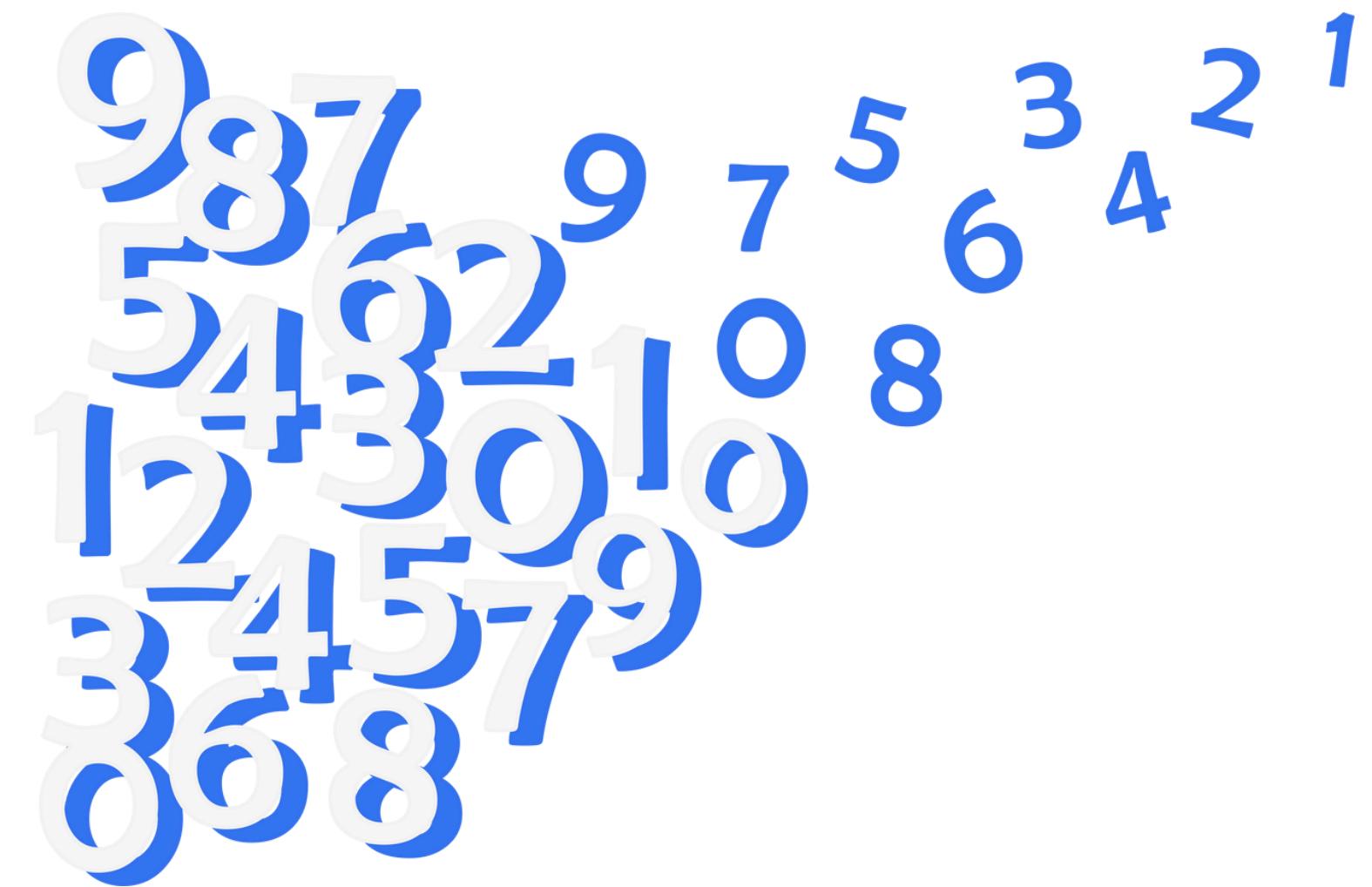
```
double number = 30;  
double rNumber = Math.random() * number;
```

```
int number = 30;  
int rNumber = (int) (Math.random() * number);
```



EJERCICIO

- Pídele al usuario que ingrese dos números por teclado.
- Utiliza los datos proporcionados para generar 5 números enteros de manera aleatoria.



A decorative graphic featuring a cluster of floating numbers from 0 to 9. The numbers are rendered in a 3D style with blue outlines and shadows, appearing to float in space. The arrangement is roughly triangular, with larger numbers at the bottom and smaller ones towards the top right.



EJERCICIO

- Crea un programa que pida ingresar por teclado una frase o una palabra.
- Con ayuda de un ciclo, cuenta cuantas vocales hay en total en la frase/palabra recién ingresada.
- **chartAt** y **length** te pueden ayudar a resolverlo.





EJERCICIO

- Utiliza la frase ingresada en el ejercicio anterior.
- Reemplaza todas las vocales encontradas por la letra ‘a’.
- Muestra el resultado por pantalla.





EJERCICIO

- Crea un pequeño programa que permita al usuario ingresar una palabra o una frase.
- Almacena el dato en una variable.
- Utiliza los conocimientos adquiridos para invertir el texto letra por letra y almacenar el resultado en una nueva variable.
- **Ejemplo:**
 - hola mundo -> odnum aloh





EJERCICIO

- Es hora de aplicar los conocimientos en nuestra simulación de un banco.
- **¿Cómo podemos mejorarla?**
- Quizás es hora de pensar en un menú interactivo con ayuda de los ciclos...





AWAKELAB

#programmingbootcamp

nodovirtual.awakelab.cl

 jELOU futurejob by  Adalid Chile