



Agencia de  
Aprendizaje  
a lo largo  
de la vida

# Codo a Codo inicial

## Clase 18

# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Clase 17

## Estructura Secuencial

- Variables y tipos.
- Operadores aritméticos.
- Operadores lógicos
- Operadores relacionales.

## Clase 18

## Instrucción de Entrada

- Objetos y métodos.
- Flujo de Entrada.
- Programa básico secuencial.
- Conversión de tipos: Casteo y Parseo
- Desafíos.

## Clase 19

## Estructura de Control

- Estructura if
- Estructura if-else



# Write Once, Run Anywhere

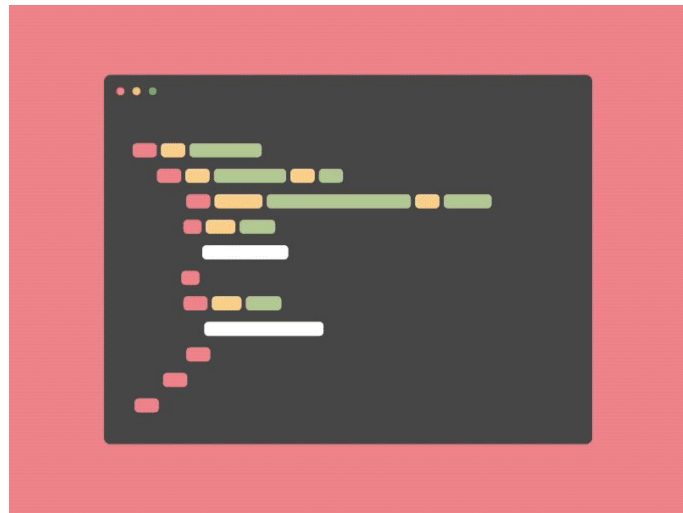
(Escríbelo una vez, ejecútalo en cualquier lugar)

# Introducción a Objetos y métodos

# Objetos

## ¿Que es un objeto?

- Un objeto es un bloque de código, con ciertas características que permiten llamarlo objeto.
- Es “un programita” dentro de nuestro espacio de trabajo que nos permite desarrollar ciertas funcionalidades.



# Objetos

- En Java es común crear un **objeto** para utilizarlo en un programa.
- Estos objetos en muchos casos ya están **creados y se los importa desde librerías**.
- Lo único que tenemos que hacer es **utilizar estos objetos en nuestro programa**.
- La analogía es como en la imagen: **crear un objeto desde uno que ya existe** para utilizarlo en nuestro contexto.



# Objetos y Métodos

- Todo objeto, así como en la vida real, tienen atributos y métodos.
- **Los atributos**, en un principio podemos definir a los atributos como **las características del objeto**
- **Los métodos**, podemos asimilarlos a **¿Qué cosas puede hacer el objeto?**.

*Vemos un ejemplo gráfico para entender el concepto...*



# Objetos y Métodos

- Tenemos un objeto Homero
- Tenemos un método correr()



# Relacionando objeto y método

Haciendo

**Homero.correr()**

Obtenemos un objeto haciendo algo...

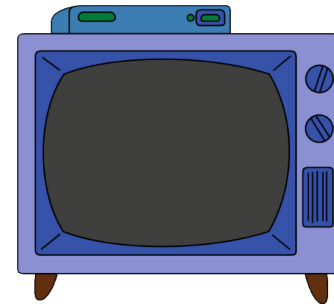


# Objetos y Métodos

- Un objeto puede combinar métodos
- Tenemos un objeto Homero
- Tenemos un método correr()
- Tenemos un método mirarTvSentado()



Agencia de Aprendizaje  
a lo largo de la vida >>>



# Relacionando objeto y método

Haciendo

**Homero.correr().mirarTvSentado()**

Obtenemos un objeto haciendo algo  
como...

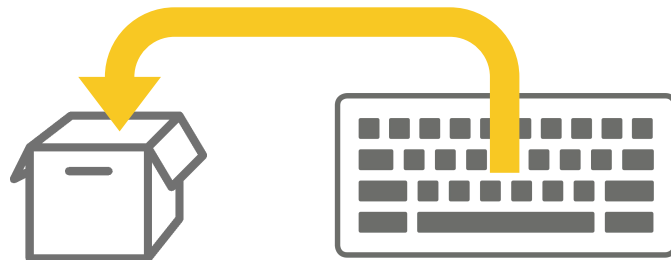


# Flujo de entrada

(Entrada de datos)

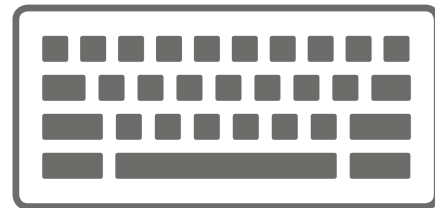
# Entrada de datos desde el teclado

- Imaginemos que necesitamos ingresar datos por teclado.
- Entonces en Java utilizaremos “un objeto virtual, un teclado virtual”, **esto en la realidad será un bloque de código que ya está escrito.**
- Este objeto nos va a permitir, mediante algún método, **captar el ingreso por teclado y almacenarlo en una variable.**



# ¿Cómo creamos este teclado virtual?

- Este **“teclado virtual”** ya está creado, nuestro trabajo es **definirlo e inicializarlo** como si fuese una variable, pero ahora **hablando con mayor propiedad** tenemos que decir **definimos nuestro objeto e instanciamos un objeto de la clase Scanner**.
- Este objeto lo traemos de la librería **java.util**;
- Entonces seguidamente vamos a crear un objeto **“miTeclado”**.



# ¿Cuál es la sintaxis para crear un teclado virtual?

```
import java.util.Scanner  
  
//Declaro el objeto  
<Objeto> <nombre_del_objeto>;  
  
//Inicializo el objeto  
<nombre_del_objeto> = new <Objeto()>;
```



# ¿Cuál es la sintaxis para crear un teclado virtual?

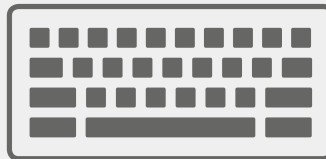
## En código será

```
/*Al principio y fuera de la clase antes de los comentarios de  
cabecera importamos*/
```

```
import java.util.Scanner
```

```
//Dentro del metodo main declaro el objeto Scanner  
Scanner miTeclado;
```

```
//Inicializo el objeto  
miTeclado = new Scanner(System.in);
```



## ¿Qué sigue después?

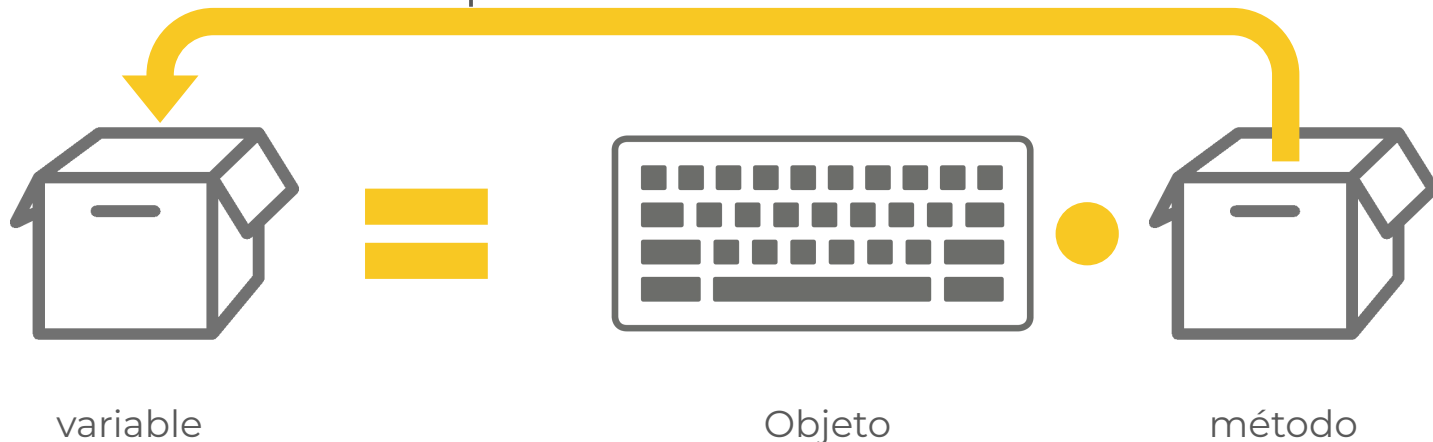
- Esta historia romántica continúa con la **inicialización de la variable** que contendrá al valor ingresado por teclado.
- Se inicializa una variable, sin más ni menos, con el **tipo que le corresponda según lo que almacenaremos**, *int*, *double*, *String*, *boolean*.

```
//Inicialización de variables  
    <tipo> <identificador>;
```

```
//Ejemplo  
int miEntero;
```

# Asignamos el objeto con su método a la variable

- Esta variable será **inicializada con el objeto** “miTeclado” y una **instrucción o método** con el cuál le indicaremos que almacene el valor por teclado en la variable.



# Inicializando una variable

```
//Declaro la variable  
<Tipo> <nombre_de_la_variable>;  
//Inicializo la variable con la siguiente expresión  
<nombre_de_la_variable> = <nombre_del_objeto>.<Metodo>;
```



## En código será

```
//Declaro la variable que almacena cadena  
String miVariable;  
  
//Inicializo la variable con objeto y método que almacena una cadena  
miVariable = miTeclado.nextLine();
```

## ¡Ojo con el método a aplicar!

- El método a aplicar va a depender del tipo de variable a almacenar.
- A continuación vas a poder ver un listado de los métodos más comunes

# Métodos de lectura de datos

```
// entrada de una cadena para una variable String
    <nombre_de_la_variable> = <nombre_del_objeto>.nextLine();
// entrada de un carácter para una variable tipo char
    <nombre_de_la_variable> = <nombre_del_objeto>.next().charAt(0);
// Entrada de datos numéricos para variables tipo byte, short, int y
float
    <nombre_de_la_variable> = <nombre_del_objeto>.nextInt();
    <nombre_de_la_variable> = <nombre_del_objeto>.nextLong();
    <nombre_de_la_variable> = <nombre_del_objeto>.nextDouble();
```

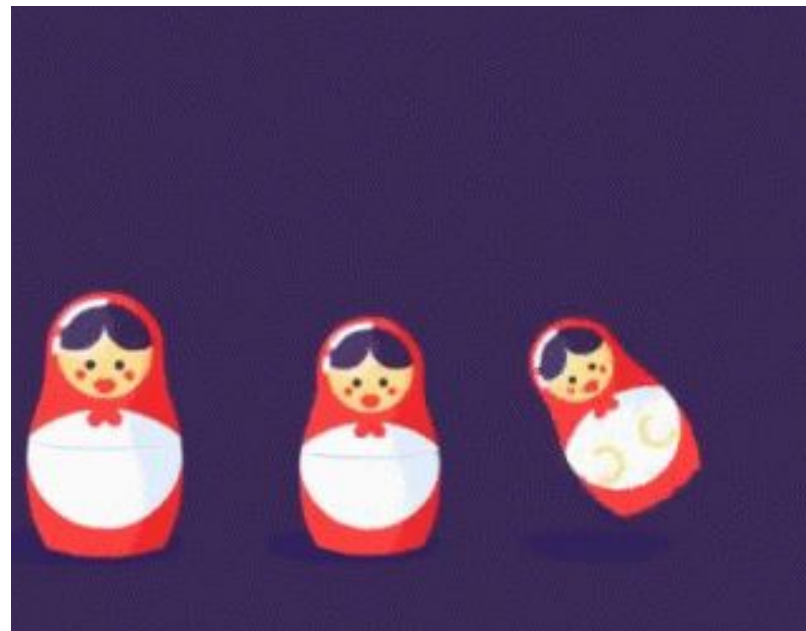


# Conversión de tipos - casteo/parseo de variables

- Muchas veces es necesario realizar conversiones de tipos cuando se evalúa una expresión aritmética.
- La conversión de tipo se da cuando queremos llevar de una primitiva compatible a otra superior o inferior en el caso de las numéricas.
- O se puede dar para convertir tipos distintos no compatibles ejemplo de String a int.

# Conversión de tipos - casteo de variables

- Imaginen que se trata de una mamushka donde un **recipiente más grande almacena a otro pequeño y viceversa**.
- Rápidamente imaginamos que un recipiente **más grande en uno pequeño resultará recortado de alguna forma** y esto será así.
- En la conversión de tipos hay veces que **se pierde información**.





# Conversión de tipos - casteo de variables

- Ampliación o conversión automática de tipos

byte → short → int → long → float → double

```
//pasamos de int a long  
int i = 100;
```

```
// automatic type conversion  
long l = i;
```

# Conversión explícita

- Casting o casteo

**double** ➡ **float** ➡ **long** ➡ **int** ➡ **short** ➡ **byte**

```
// Pasamos de int a long  
double decimal = 100.50;
```

```
// Conversión explícita  
int entero = (int)decimal; // esto mostrará 100
```

# Conversión de tipos - sintaxis

La sintaxis para la conversión explícita de tipos es la siguiente

```
<nuevo_tipo> <nueva_variable> = (<nuevo_tipo>) <variable_a_convertir>
```

Ej:

```
//quiero pasar de double a long
    double miNumero = 100.045;
//casting de tipo
    long l = (long)miNumero;
//se asigna un nuevo nombre de variable
```

# Conversión de tipos

- **Para pasar de un int a String**

```
int miInt = 43;  
  
// convierto de int a String  
String sInt = Integer.toString(miInt); //El resultado es una cadena de texto  
                                         //almacenada en sInt.
```

- **Para pasar de un String a int - Parseo**

```
// convierto de String a int  
int miInt2;  
miInt2 = Integer.parseInt(sInt); //Convierto sInt a int y lo almaceno en miInt2
```

## Desafío I - Historia de usuario

- **Como** Usuario **quiero** ingresar por teclado mi nombre y dos números **para** que me devuelva un saludo personalizado por consola y la suma de los dos números con la leyenda “La suma de los dos números es: ”

# Desafío I - Debemos hacer

- Mediante flujos de entrada y salida
  - a. Instanciar un objeto "miTeclado".
  - b. Crear una variable "nombre" como String.
  - c. Crear una variable "num1" como int.
  - d. Crear una variable "num2" como int.
  - e. Inicializar TODO.
  - f. Realizar la instrucción de salida "Ingrese su nombre: " mediante `System.out.println()`.
  - g. Realizar la instrucción de entrada que almacene el nombre en la variable "nombre".
  - h. Realizar la instrucción de salida "Ingrese el primer número: " mediante `System.out.println()`.
  - i. Realizar la instrucción de entrada que almacene el primer número en "num1".
  - j. Realizar la instrucción de salida "Ingrese segundo número: " mediante `System.out.println()`.
  - k. Realizar la instrucción de entrada que almacene el segundo número en "num2".
  - l. Realizar un flujo secuencial simple donde sumamos num1 + num 2 y lo almacenamos dentro de una variable que llamaremos resultado.
  - m. Realizar la instrucción de salida del resultado y el saludo `System.out.println()`.

# Desafíos

1. Programa Java que lea dos números enteros por teclado y los muestre por pantalla.
- 2.- Hacer una aplicación que permita ingresar 2 números y mostrar sus operaciones básicas.
3. Programa Java que lee un número entero por teclado y obtiene y muestra por pantalla el doble y el triple

## Desafío 4 - Historia de usuario

- **Como** Usuario **quiero** ingresar por teclado mi nombre y tres números **para** que me devuelva por consola un saludo personalizado me muestre los números que introduje y además la suma de los tres números con la leyenda “La suma de los tres números es: ”



## Desafío II - Debemos hacer

- Mediante flujos de entrada y salida
  - a. Instanciar un objeto “miTeclado”.
  - b. Crear una variable “nombre” como String.
  - c. Crear una variable num1, num2, num3 como int.
  - d. Realizar la insinstruccióntruccion de salida “Ingrese su nombre: ” mediante `System.out.println()`.
  - e. Realizar la instrucción de entrada que almacene el nombre en la variable “nombre”.
  - f. Realizar la instrucción de salida “Ingrese el primer número: ” mediante `System.out.println()`.
  - g. Realizar la instrucción de entrada que almacene el primer número en “num1”.
  - h. Realizar la instrucción de salida “Ingrese segundo número: ” mediante `System.out.println()`.
  - i. Realizar la instrucción de entrada que almacene el segundo número en “num2”.
  - j. Realizar un flujo secuencial simple donde sumamos num1 + num 2 y lo almacenamos dentro de una variable que llamaremos resultado.
  - k. Realizar la instrucción de salida del resultado y el saludo `System.out.println()`.

# Repo de clases

<https://app.codingrooms.com/w/p56gqFuZKKsg>

# Herramientas que utilizamos en clases



**VSCode+plugins**

# No te olvides de dar el presente

## **Recordá:**

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**
- **Realizá los ejercicios obligatorios.**

**Todo en el Aula Virtual.**