**Actividad**

Elementos de un programa

**Objetivos**

* Utilizar un IDE como Eclipse , NetBeans o Visual Studio Code y depurar sus errores.
* Reconocer y crear los elementos propios de un programa informático.
* Reconocer y utilizar conversiones de tipos.
* Utilizar los comentarios en el código.

|  |
| --- |
| **¿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 1. Elementos de un programa. |

***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 actividad01***

**Identificación de los elementos de un programa informático y utilización de las estructuras de control.**

Configura tu IDE y crea un archivo de nombre **Ejercicio01.java** dentro de un package (una carpeta) de nombre **actividad01**. Copia el siguiente código, añade los *import* necesarios (import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader;) y ejecútalo.

public class Ejercicio01 {

public static void main(String[] args) throws IOException {

int manzanas\_tienda;

int manzanas\_queremos\_comprar;

double precio\_manzana;

double dinero\_pagado;

manzanas\_tienda = 20;

manzanas\_queremos\_comprar = 2;

precio\_manzana = 0.40;

System.out.println("Vamos a comprar " + manzanas\_queremos\_comprar + " manzanas");

// modificamos el contenido de las variables numericas para representar la

// compra de manzanas

dinero\_pagado = precio\_manzana \* manzanas\_queremos\_comprar;

manzanas\_tienda = manzanas\_tienda - manzanas\_queremos\_comprar;

// mostramos por consola el valor de las variables numericas. De forma implícita

// se convierte de de numero a string.

System.out.println("Nos han costado:" + dinero\_pagado);

System.out.println("En la tienda quedan " + manzanas\_tienda + " manzanas");

System.out.println("Quantas mazanas más quieres comprar?");

// Declaramos una variable compleja de nombre "br" y de tipo "BufferedReader".

// Las variables del tipo "BufferedReader" contienen funciones para leer datos

// por consola

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

// utilizamos la función "readLine" de la variable "br" para leer un dato por

// consola. Guardamos el número introducido en la variable "valor\_escrito"

String valor\_escrito = br.readLine();

/\*

\* Convertimos mediante una conversion explicita la variable "valor\_escrito" a

\* un valor entero para poder operar con él y lo almacenamos en la variable

\*/

manzanas\_queremos\_comprar = Integer.parseInt(valor\_escrito);

System.out.println("Vamos a comprar " + manzanas\_queremos\_comprar + " manzanas");

// modificamos el contenido de las variables numericas para representar la

// compra de manzanas

dinero\_pagado = precio\_manzana \* manzanas\_queremos\_comprar;

manzanas\_tienda = manzanas\_tienda - manzanas\_queremos\_comprar;

// mostramos por consola el valor de las variables numericas. De forma implícita

// se convierte de de numero a string.

System.out.println("Nos han costado:" + dinero\_pagado);

System.out.println("En la tienda quedan " + manzanas\_tienda + " manzanas");

}

}



**No le voy a mentir profesor, no tengo ni idea de cómo funcionan los paquetes de Java asi que pongo captura de mi codigo por si por error mio no lo envio bien. Toda mi experiecia es en C y Python y mucho texto incluso comparado a C.**

**Este apartado no ha sido muy complejo, solo eh adaptado las anteriores lineas a su nueva función, declarando melones y su precio, y borrando la linea de compra automatica de las manzanas a solo tener la manual.**

Al código anterior (ejercicio01.java) añade después del último System.out.println(); un código que indique que la tienda tiene 20 melones 2,30€ el melón y a continuación pida “Cuantos melones vas a comprar?”, se reciba un número escrito por el usuario y finalmente se indique cuantos melones quedan en la tienda y cuanto a costado la compra (un valor con decimales).

**Ejercicio02.java:** Crea un nuevo archivo de nombre **Ejercicio02.java** dentro del package de nombre **actividad01**.

Para cada tipo de dato simple añade dentro de la función **main** de **Ejercicio02**  un comentario indicando el tipo de dato. A demás para cada tipo de dato simple declara una variable de ese tipo, asígnale un valor y muestra un mensaje que explique las características del tipo de dato y muestre el valor de la variable.   
Ej:

//Tipo de dato entero de 32 bits de longitud

**int** variableEntera =10;

System.*out*.println("Variable tipo int muestra datos enteros de 32 bits. Ejemplo:"+variableEntera);

Dentro de la clase de nombre **Ejercicio02**, añade un ejemplo de uso de cada uno de los operadores aritméticos, relacionales, lógicos, unitarios y de asignación. Indicando con un comentario y con un mensaje por consola cual será el resultado de cada uno de ellos.  
Ej:  
System.***out***.**println**("--Operadores lógicos--");

System.***out***.**println**("El resultado de true && false es:"+ (**true** && **false**) );



**Hablando en plata, ha sido en parte doloroso escribir una wikipedia en un IDE, pero es la tarea en mano. He tenido un pequeño problema con los operadores aritmeticos unarios y de asignación. El codigo corre pero el resultado es un poco 'wonky' en los operadores aritmeticos previamente mencionados.**

RECUERDA QUE EL CÓDIGO DEBE DE PODER SER EJECUTABLE