**RESUMEN UNDIAD 1Y2**

Joaquín García Téllez 1ºDAW B

**UNIDAD 1**

1.5. El programa principal

Para empezar a crear un programa es necesaria la siguiente estructura:

package nombreprograma;->nombre del proyecto

public class nombreprograma{

public static void main(Sting() argsw){}}

1.6. Palabras reservadas

-Existen ciertas palabras reservadas, que solo se pueden usar en determinadas ocasiones; como por ejemplo: package, class o public.

-Existen otras las cuales tienen un valor literal: true, false y null.

1.7. Concepto de variable

-Una variable es aquella con un valor determinado.

-El nombre que se le da se considera el identificador, donde se distingue entre minúsculas y mayúsculas, y deben seguir ciertas reglas:

-Comenzar por: letra, \_, $.

-Seguir con: letra, digito, \_, $.

-Sin longitud máxima.

1.8. Tipos primitivos

-Las variables se guardan en la memoria del ordenador, y cada una necesita un tamaño fijo por seguridad. Esto se solucione definiendo la variable con uno de estos tipos: byte, short, int, long, float, doublé, boolean y char.

double importe;

double importe=100.75;

1.8.2 Rangos

-Es muy importante tener en cuenta los rangos de cada variable para saber cuál usar, para que no sufra un desbordamiento de memoria.

1.9. Variables de objetos

Si a una variable con tipo primitivo no se le asigna un valor este automáticamente adopta el valor de: null.

1.10. Constante

Una constante es aquella donde su valor nunca cambia la cual se escriben:

final tipo nombreconstante;

1.11. Comentarios

-Si quisieras dar información sobre algún código o algo que no queremos que se ejecute, esto se puede comentar con los siguientes signos: /\*\*/, //, /\*\*.

1.12. API de java

-Existen una cosa llamada clases las cuales se utilizan para multitud de tareas:

Lecturas de datos, cálculos complejos, manejo de errores, escritura de datos.

1.12.1 Paquetes

Las clases se agrupan en paquetes, y estos se pueden agrupar en otros paquetes.

Las clases se identifican mediante su nombre, que se compone del nombre del paquete y del suyo:java.lang.Math.

Una alternativa para no tener que llamar siempre es : import

La cual tiene 2 formas:

-Estática: se usa directamente el método.

-No Estática: Scanner sc=new Scanner (System.in);

1.12.2 Salida por consola

Para poder mostrar un mensaje en la consola utilizaremos el siguiente script:

System.out.print(“Mensaje”)

Atajo: syso ctrl+espacio.

1.12.3 Entrada de datos

-Para recabar información atraes del teclado se utiliza el script: Scanner.

Seguido de sc.next (tipe).

Para utilizar esta función antes debemos ejecutar:

Import java.util.Scanner;

1.13. Operaciones básicas

-El operador = se utiliza para modificar el valor de una variable.

-Una expresión no es más que un conjunto de operaciones.

1.13.2. Operadores aritméticos

-Existen los siguientes operadores los cuales se utilizan en operaciones matemáticas: +,-,\*,/,%.

1.13.3. Operadores racionales

-Existen operadores los cuales se utilizan en casos lógicos o booleano, para comparar cosas: ==!=,<,<=,>,>=.

1.13.4. Operadores lógicos

-Existen operadores los cuales sustituye a expresiones lógicas: &&->y, ||->o!->no-

1.13.5. Operadores opera y asigna

-Estos operadores se utilizan para añadir el valor a la variable indicada, esto se realiza: operación=.

1.13.6. Operador ternario

-Este operador devuelve un valor entre dos posibles

1.13.7. Precedencia

-La precedencia establece que operaciones se realizan antes.

1.14. Conversión de tipos

-Las variables tienen asignado un tipo, la cual debe ser acorde a él.

-Y si intentáramos darle un valor no acorde nos saltaría un error.

**UNIDAD2**

2.1. Expresiones lógicas

Primero hablemos del tipo boolean que tiene valor: true, false.

2.1.1. Operadores relacionales

Los operadores relacionales son aquellos que comparan expresiones numéricas para generar valores booleanos.

2.1.2. Operadores lógicos

Los operadores lógicos permiten construir condiciones más complejas. &&||!.

2.2. Condicional simple: if

La sentencia if proporciona un control sobre un conjunto de instrucciones que se pueden ejecutar o no, dependiendo de la condición. Si dicha condición se cumple se ejecutaran las siguientes instrucciones, si no se da la condición no se ejecutaran las instrucciones dentro del if.

if(condición){}

2.3. Condicional doble: if-else

Existe otra sentencia denominada else, la cual se pone después de un if , entonces si la condición del if no se cumple sucede las instrucciones indicadas en el else.

if(condición){

}else{}

2.3.1. Operador ternario

Un operador ternario permite seleccionar un valor entre dos posibles, dependiendo de la condición.

Variable= condición? valor1: valor2

2.3.2. Anidación de condicionales

Cuando queremos utilizar múltiples comprobaciones lo suyo es utilizar else if, que cumple la misma función que el if, con la diferencia de que lo omite si no hace falta.

2.4. Condicional múltiple: switch

En ocasiones de que usemos muchos if anidados produce un código poco legible; la solución es usar la sentencia switch:

Switch (expresión) {

case valor:

instrucciones}

break;

Con esto lo que conseguimos hacer el Codigo mas legible y óptimo.