

# Módulo Programación.

## 1º DAW.



PRÁCTICA2: Variables y constantes.

UNIDAD DE TRABAJO 2.

Profesor: Pedro Antonio Santiago Santiago.



# 1. Introducción.

Las variables y constantes son unos de los fundamentos de los lenguajes de programación, en los lenguajes fuertemente tipados como Java se han de definir el tipo de dato que reserva espacio en la memoria para almacenar los posibles valores de la variable. Con estas variables y constantes se pueden realizar diferentes tipos de operaciones como son las lógicas, aritméticas o de comparación entre otras, por último suele ser necesario realizar conversiones de variables entre diferentes tipos, esta conversión implica la asignación de valores a direcciones de memoria, que pueden ser menores, iguales o mayores que la de la variable a convertir. En esta práctica se tratan la definición y uso de variables.

## 2. Materiales.

Ordenador con JDK 8 instalado, editor de texto.

## 3. Desarrollo de la práctica.

Es imprescindible documentar el código.

### 3.1. Ejercicios con editor de texto.

#### 3.1.1. Ejercicio 1 (Solucionado).

Se desea crear un pequeño programa que a partir del peso en kg de una persona y la altura en centímetros devuelva el IMC, solicitando los datos por la terminal.

Se abre un editor de texto como Notepad++, Kate, Gedit o Notepad (mejor los 3 primeros ya que remarcan la sintaxis con colores).

Crear un fichero con el siguiente código:

```
import java.util.*;
import java.lang.Math;
class Ejercicio1 {
    public static void main (String args []){
        int peso=0,altura=0;
        double resultado;
        Scanner input= new Scanner (System.in);
        System.out.println("Introducir la altura en cm:");
        altura=input.nextInt();
```

```
        System.out.println("Introduccir el peso en Kg:");  
        peso=input.nextInt();  
        resultado= (double)peso/Math.pow( (float)altura/100,2);  
        System.out.println("El resultado es "+resultado+" ");  
    }  
}
```

- En el punto 1 se incluye el código externo que se utilizará, en este caso el código importado (se añade en el “linkado”) permite obtener datos del teclado con Scanner (es una clase, se tratará en temas posteriores) y realizar la operación de potencia con Math.pow.
- En el punto 2 se definen la clase Ejercicio1, este concepto se trata en temas posteriores, y el método estático main, que indica el punto de entrada.
- En el punto 3 se definen las variables y su tipo.
- En el 4 se define una variable llama input de tipo Scanner (es un objeto, no prestarle mucha atención ahora) y se crea con new (es un tipo complejo).
- Por último en el punto 5, se solicitan los datos y se realizan los cálculos, destacar la conversiones necesarias para obtener valores válidos, ya que una división entre enteros devuelve un entero, y se desea un doble o float.

```
1  
2  import java.util.*;  
3  import java.lang.Math;  
4  class Ejercicio1 {  
5      public static void main (String args []) {  
6          int peso=0, altura=0;  
7          double resultado;  
8          Scanner input= new Scanner (System.in);  
9          System.out.println("Introduccir la altura en cm:");  
10         altura=input.nextInt();  
11         System.out.println("Introduccir el peso en Kg:");  
12         peso=input.nextInt();  
13         resultado= (double)peso/Math.pow( (float)altura/100,2);  
14         System.out.println("El resultado es "+resultado+" ");  
15     }  
16 }
```

### 3.1.2.

Crear un programa que solicite los lados de un rectángulo en centímetros e indique su área.

### 3.1.3.

Calcular el área y el perímetro de un círculo, PI se ha de definir como constante. Solicitar el resto de valores por teclado

### 3.1.4.

Llegan las rebajas, es necesario crear un programa al que se le introduce el precio en euros y céntimos y realiza de forma automática un descuento de un 15%. Debe aparecer el importe original, el importe del descuento y el precio final por pantalla. Ese 15% a de definirse como constante.

### 3.1.5.

Nos piden crear un conversor de Euros a Dolares, se ha de solicitar la cotización y la cantidad de Euros.

### 3.1.6.

Crear un programa que resuelva una ecuación de segundo grado, a partir del siguiente código, teniendo en cuenta que se pueden dar 3 casos posibles en función del discriminante, siendo este

$$\Delta = b^2 - 4 \cdot a \cdot c$$

Dependiendo del valor del discriminante se tienen los siguientes casos:

- Si  $\Delta > 0$ , hay dos soluciones reales distintas.
- Si  $\Delta = 0$ , hay dos soluciones reales iguales.
- Si  $\Delta < 0$ , no hay soluciones reales.

Recordar que la expresión de una ecuación de segundo grado es:

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

```
import java.util.*;
import java.lang.Math;
class Ejercicio6 {

    public static void main (String args []){
        int a,b,c,discriminante;
        double temporal,s1,s2;
```

```
//objeto que permite leer de teclado
Scanner input= new Scanner (System.in);
System.out.println("Introduccir el parámetro a:");
a=input.nextInt();
System.out.println("Introduccir el parámetro b:");
b=input.nextInt();
System.out.println("Introduccir el parámetro c:");
c=input.nextInt();
//usar el método Math.pow para realizar la potencia
discriminante=
//caso 1
if(){
}else{
    //caso 2
    if(){

    }
    //caso 3
    else{
        System.out.println("No tiene soluciones reales");
    }
}
}
```

### 3.1.7.

Realizar un programa que solicite los días, horas y minutos y devuelve la cantidad de horas, el total de minutos y el total de segundos.

### 3.1.8.

Modificar el ejercicio anterior para que muestre la cantidad de días en decimales, por ejemplo si se introduce 1 día, 6 horas, el resultado ha de ser 1,25 días.

### 3.1.9.

Crear un programa que indique cuantas décadas y años tiene una persona. Un ejemplo de entrada:

Indicar su edad en años: 86

La persona tiene 8 décadas y 6 años.

```
Introduccir la edad:  
86  
La persona tiene 8 dÃ©cadas y 6 aÃ±os
```

### 3.1.10.

Nos han contratado para crear una pequeña aplicación que indica si un alumno posee derecho a beca o no y la cuantía, a partir del código siguiente completar las condiciones para que funcione correctamente.

Por ejemplo, en la declaración de la renta se tiene que si se hace conjunta (2 personas) y los ingresos son mayores de 30000 € se pagarán 5000€. La siguientes expresiones lógicas que devuelven cierto o falso cumplen con lo anterior:

```
if(conjunta==true && ingresos>3000){  
...  
}  
if (!conjunta==false && !(ingresos<=3000)){  
...  
}  
if(conjunta !=false && ingresos>3000){  
...  
}
```

Las preguntas que se realizan son las siguientes:

- Menor de 24 años.
- Familia numerosa.
- Nota media.

Las cuantías son:

1. Si menor de 24 años, familia numerosa y nota media mayor o igual que 7.5, o simplemente nota mayor o igual que 7.5: 3000 €.
2. Si no es menor de 24 años, pero si familia numerosa y la nota mayor o igual que 5 y menor de 7.5 o si es menor 24 años, no es familia numero y la nota mayor o igual que 5 y menor de 7.5: 2500 €
3. Si no es menor de 24, no es familia numerosa y la nota está mayor o igual que 5 y menor de 7,5: 2000 €

#### 4. En otro caso no tiene derecho a beca.

```
import java.util.*;

/*Nos han contratado para crear una pequeña aplicación que indica si un alumno
posee derecho a beca o no y la cuantía, a partir
del código siguiente completar las condiciones para que funcione
correctamente.*/

class Ejercicio10 {
    static final int decada=10;
    public static void main (String args []){
        int edad;
        boolean numerosa;
        float notamedia;
        //objeto que permite leer de teclado
        Scanner input= new Scanner (System.in);
        System.out.println("Introduccir la edad:");
        //caso 1
        if( ){
            System.out.println("La beca es de 3000 euros");
        }
        else{
            /*caso 2*/
            if(){
                System.out.println("La beca es de 2500 euros");
            }
            else{
                //caso 3
                if ( ){
                    System.out.println("La beca es de 2000 euros");
                }
                // caso 4 no tiene derecho a beca
                else{
                    System.out.println("No tiene derecho a beca");
                }
            }
        }
    }
}
```

### 3.1.11. .

Se necesita crear un programa que realice operaciones AND entre direcciones IP's, a partir del código que se facilita y utilizando operaciones a nivel de bits, realizar dicha operación mostrando el resultado en decimal, hexadecimal, octal y binario.

Para escribir una variable en hexadecimal usar:

```
String.format("%02x", octetor1)
```

Para escribir una variable en octal:

```
String.format("%03o", octetor1)
```

Y para escribir una variable en binario:

```
String.format("%8s", Integer.toBinaryString(octetor1)).replace(' ', '0')
```

Un ejemplo para escribir una ip en hexadecimal:

```
System.out.println("Resultado en  
hexadecimal:"+String.format("%02x", octetor1)+"."+String.format("%02x", octetor2)  
+"."+String.format("%02x", octetor3)+"."+String.format("%02x", octetor4));
```

```
PS G:\Mi unidad\encanta2021_22\programacion\unidades\unidad1_introducción_a_la_programación\soluciones_practia2> java Ejercicio11
Introducir el primer octeto de la primera ip:
45
Introducir el segundo octeto de la primera ip:
67
Introducir el tercer octeto de la primera ip:
54
Introducir el cuarto octeto de la primera ip:
78
Introducir el primer octeto de la segunda ip:
48
Introducir el segundo octeto de la segunda ip:
54
Introducir el tercer octeto de la segunda ip:
36
Introducir el cuarto octeto de la segunda ip:
78
IP 1:45.67.54.78
IP 2:48.54.36.78
Resultado en decimal:32.2.36.78
Resultado en hexadecimal:20.02.24.4e
Resultado en octal:040.002.044.116
Resultado en binario:00100000.00000010.00100100.01001110
```

### 3.1.12.

El siguiente programa realiza un cálculo simple relacionado con la electricidad. ¿Qué hace? Modificarlo para que el cálculo tenga decimales. Razonar si es conveniente o no comentar el código fuente.

```
import java.util.*;
```



```
class Ejercicio12 {

    public static void main (String args []){

        int a,b,c;
        Scanner input= new Scanner (System.in);
        //se obtiene el primer octeto
        System.out.println("Introduccir a:");
        a=input.nextInt();
        System.out.println("Introduccir b:");
        b=input.nextInt();
        c=b/a;
        System.out.println("Resultado es:"+c);

    }

}
```

### 3.1.13. .

El siguiente programa calcula la fórmula del interés simple,pero no lo está haciendo bien. Solucionarlo.

```
import java.util.*;
class Ejercicio13 {

    public static void main (String args []){

        //Interes= Capital* %interes(mensual)*tiempo(meses);
        int intereses_anual=0, capital_inicial=0, tiempo_anyos=0;
        int importe_interes=0;
        Scanner input= new Scanner (System.in);
        //se obtiene el primer octeto
        System.out.println("Introduccir el interes anual %:");
        intereses_anual=input.nextInt();
        System.out.println("Introduccir el capital inicial en euros:");
        intereses_anual=input.nextInt();
        System.out.println("Introduccir el tiempo en anyos:");
        tiempo_anyos=input.nextInt();
        importe_interes+=intereses_anual*12*tiempo_anyos;
        System.out.println("Resultado es:"+importe_interes);

    }

}
```

**Recordar que para poder ejecutar programas Java es necesario tener instalado el Java Runtime Environment o JRE. Para el desarrollo se necesita el Java Developer Kit o JDK. En este caso se instalará la versión 8.**

## **4. Entrega.**

La práctica se entrega en formato PDF, , en el campus virtual [ww.aules.edu.gva.es](http://ww.aules.edu.gva.es) (pendiente de matricula e inicio de curso). En esta práctica la fecha de entrega se encuentra abierta ya que aún no se han dado de alta a los alumnos en el curso.

El documento de entrega ha de tener los siguientes puntos.

1. Portada. (Título de la práctica y autor).
2. Introducción.
3. Desarrollo de la práctica. **Documento de texto con el código fuente comentado, capturas de ejecución y respuestas a las preguntas.**
4. Conclusiones. (Pequeño comentario sobre la práctica: dificultad, problemas encontrados...).

## **5. Evaluación.**

**Unos días después de la entrega se realizará la corrección de forma presencial donde el profesor preguntará cuestiones sencillas sobre la práctica para comprobar la autoría de la misma.**

RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

CE1d. Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.

CE1e. Se ha modificado el código de un programa para crear y utilizar variables.

CE1f. Se han creado y utilizado constantes y literales.

CE1g. Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.

CE1h. Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.

CE1i. Se han introducido comentarios en el código.