Tarea PROG02: Creación de mi primer programa

Curso DAM 2024/25

Calificación obtenida: 10/10

DETALLES DE LA TAREA

En esta unidad hemos tenido la oportunidad de crear nuestro primer programa en Java. Hemos realizado pequeños ejemplos sobre cada apartado tratado, y ahora se trata de ponerlos en práctica mediante la siguiente relación de ejercicios

ÍNDICE

Ejercicio 1: Variables y tipo de datos **2**
- a) Valor máximo no modificable: 5000 2
- b) Si el nuevo empleado tiene carnet de conducir o no 2
- c) Un mes del año en formato numérico y como cadena 3
- d) El nombre y apellidos de una persona 3
- e) Sexo con dos valores posibles 'V' o 'M' 4
- f) Milisegundos transcurridos desde el 01/01/1970 hasta nuestros días 4
- g) Saldo de una cuenta bancaria 4
- h) Distancia en kms desde la Tierra a Júpiter 5

Ejercicio 2: Identificadores de variables **6**
Ejercicio 3: Evaluación de operaciones **7**
Ejercicio 4: Mayor o menor de edad **8**
Ejercicio 5: Mostrar tiempo **9**
Ejercicio 6: Enumerado para razas de perros **10**
Ejercicio 7: Alumnos matriculados **12**

Ejercicio 1: Variables y tipo de datos

Crea un proyecto en Netbeans denominado **PROG02_Ejerc1 con una clase clase y método main** y **declara e inicializa** una variable para almacenar cada uno de los siguientes valores. Trata de utilizar el tipo de datos que más se ajuste a los datos. Justifica tu elección.
Muestra el valor de cada variable en pantalla de forma que cada valor aparezca en una línea, teniendo en cuenta que NO puedes utilizar la orden **`println`**.

a. Valor máximo no modificable: 5000.
b. Si el nuevo empleado tiene carnet de conducir o no.
c. Un mes del año en formato numérico y como cadena.
d. El nombre y apellidos de una persona.
e. Sexo: con dos valores posibles 'V' o 'M'.
f. Milisegundos transcurridos desde el 01/01/1970 hasta nuestros días.
g. Saldo de una cuenta bancaria.
h. Distancia en kms desde la Tierra a Júpiter

a) Valor máximo no modificable: 5000

```
final int VALOR_MAXIMO = 5000;
       System.out.print ("a) Valor máximo no modificable: " + VALOR_MAXIMO + "\n");
```

SALIDA POR CONSOLA

```
a) Valor máximo no modificable: 5000
```

JUSTIFICACIÓN

- Como la variable va a permanecer inalterable a lo largo del programa, la declaramos como constante, utilizando final
- Utilizamos int porque es de tipo entero y 5000 está dentro del rango de int
- Como es una constante, escribimos el identificador en letras mayúsculas, separando las palabras con el guión bajo (VALOR_MAXIMO)
- Como no podemos utilizar la orden println y cada valor debe aparecer en una línea, para hacer el salto de línea usamos la secuencia de escape "\n".
- Para combinar el valor de la variable VALOR_MAXIMO con el salto de línea ("\n"), usamos el operador de concatenación (+)

b) Si el nuevo empleado tiene carnet de conducir o no

```
 boolean tieneCarnet = true;
       System.out.print ("b) El empleado tiene carnet de conducir? " + tieneCarnet + "\n");
```

SALIDA POR CONSOLA

```
b) El empleado tiene carnet de conducir? true
```

JUSTIFICACIÓN

- Como la respuesta sólo puede ser Sí o No utilizo¡amos una variable de tipo booleana (boolean)
- Imaginamos que el empleado Sí tiene carnet, por eso le asignamos el valor verdadero con true

c) Un mes del año en formato numérico y como cadena

```
int mesNumero = 10;
        String mesCadena = "Octubre";
        System.out.print ("c) Mes en formato numérico: " + mesNumero + "; Mes en formato
cadena: " + mesCadena + "\n");
```

SALIDA POR CONSOLA

```
c) Mes en formato numérico: 10; Mes en formato cadena: Octubre
```

JUSTIFICACIÓN

- Utilizamos int para el valor numérico (porque está dentro del rango)
- Utilizamos String para el mes en texto (Como lo que necesitamos es almacenar caracteres en secuencia, creamos una variable de tipo String simplemente asignándole una cadena de caracteres encerrada entre comillas dobles)

d) El nombre y apellidos de una persona

```
String nombre = "María";
        String apellido1 = "Pérez";
        String apellido2 = "García";
        System.out.print ("d) Nombre y apellidos de una persona: " + nombre + " " +
apellido1 + " " + apellido2 + "\n");
```

SALIDA POR CONSOLA

```
d) Nombre y apellidos de una persona: María Pérez García
```

JUSTIFICACIÓN

- Utilizamos String para cada parte del nombre completo (nombre, apellido1, apellido2)
- Utilizamos el operador + para concatenar cadenas de caracteres.
- Usamos una variable para el nombre, otra para el apellido 1 y otra para el apellido 2 porque así el código se ve más claro.

e) Sexo con dos valores posibles 'V' o 'M'

```
public enum Sexo {V, M} //Colocar esto antes de la clase main
Sexo sexo = Sexo.M;
        System.out.print ("e) Sexo: " + sexo + "\n");
```

SALIDA POR CONSOLA

```
e) Sexo: M
```

JUSTIFICACIÓN

- Usamos enum para limitar los valores de las posibles respuestas (*Los tipos de datos enumerados son una forma de declarar una variable con un conjunto restringido de valores*.)

f) Milisegundos transcurridos desde el 01/01/1970 hasta nuestros días

```
long milisegundosDesdeEpoch = System.currentTimeMillis();
        System.out.print("f) Milisegundos desde el 1/1/1970: " + milisegundosDesdeEpoch + "ms" + "\n");
```

SALIDA POR CONSOLA

```
f) Milisegundos desde el 1/1/1970: 1744606307558ms
```

JUSTIFICACIÓN

- Utilizamos long porque es un número entero grande (sin decimales) y está dentro del rango
- Utilizamos Epoch: La Epoch es un término que se utiliza en el contexto de la computación y la programación para referirse a un momento específico en el tiempo, que sirve como punto de referencia para calcular el tiempo en sistemas informáticos. En la mayoría de los sistemas, la Epoch es el 1 de enero de 1970 a las 00:00:00 UTC. A partir de este momento, el tiempo se mide en milisegundos o segundos transcurridos.
- Obtenemos el valor de milisegundos actuales usando System.currentTimeMillis()

g) Saldo de una cuenta bancaria

```
double saldoCuenta = 13504.35;
        System.out.print ("g) Saldo en la cuenta bancaria: " + saldoCuenta + "€" + "\n");
```

SALIDA POR CONSOLA

```
g) Saldo en la cuenta bancaria: 13504.35€
```

JUSTIFICACIÓN

- Usamos double porque es un número decimal (*Aunque es posible usar `float` para el saldo bancario, es recomendable usar `double` porque proporciona una mayor precisión y ayuda a reducir los errores de redondeo*)

h) Distancia en kms desde la Tierra a Júpiter

```
long distanciaTierraJupiter = 628_730_000L;
        System.out.printf("h) Distancia desde la Tierra a Júpiter: %,dKm\n",
distanciaTierraJupiter);
```

SALIDA POR CONSOLA

```
h) Distancia desde la Tierra a Júpiter: 628.730.000Km
```

JUSTIFICACIÓN

- Usamos long porque queremos expresar la distancia como un número entero grande, sin decimales ni notación científica.
- Uso¡amos printf para dar formato al número
- Usamos %,d para formatear el número con separadores de miles:
  - %: Indica el inicio de un especificador de formato.
  - ,: Indica que se deben incluir separadores de miles
  - d: Especifica que el valor a formatear es un número entero

Ejercicio 2: Identificadores de variables

Indica si los siguientes **identificadores de variables** en Java serían válidos. Justifica tu respuesta.

Identificador de variable	Válido / No válido	Justificación
double	No válido	Se puede usar como un tipo variable pero no como un identificador de variable. Es una palabra reservada
/horaactua	No válido	Un identificador sólo puede empezar por una letra, un símbolo de subrayado (_) o el símbolo dólar ($).
$hora	Válido	Válido, aunque se desaconseja que un identificador de variable empiece con el símbolo dólar ($)
MiHora	Válido	Válido, pero lo mejor sería que empezase con una letra minúscula.
_hora	Válido	Válido, aunque se desaconseja que un identificador de variable empiece con el símbolo de subrayado (_)
5hora	No válido	No puede empezar con un número
char	No válido	Char es una palabra reservada.

Ejercicio 3: Evaluación de operaciones

Teniendo en cuenta que var1, var2 y var3 son variables de tipo boolean y están inicializadas a los siguientes valores: var1=true, var2=true y var3=false y que las variables X, Y y Z son variables enteras con valores: X=5, Y=-8 y Z=10, **indica si las siguientes operaciones se evalúan a true o false**:

Operaciones	True / False	Justificación
var1 || var2 && var3	True	En esta expresión, el operador && tiene una mayor precedencia que el operador ||, lo que significa que se evalúa primero. ● Primero se evalúa var2 && var3: ○ Esto se convierte en true && false. ○ Como el operador && necesita que las dos variables sean true y esto no es así, el resultado es false ○ La expresión resultante es: var1 || false. ● Se evalúa var1 || false: ○ Esto se convierte en true || false. ○ Dado que var1 es true (y el operador || no evalúa el segundo operando si el primero es true), el resultado final es true.
(var1 || var3) && (var2 && !var1)	True	● (var2 && !var1): true && false = false ● (var1 || var3): true || false = true ● true && false = false
(var2 || !var1 || !var3) && var1	True	● (var2 || !var1 || !var3) ○ !var1 || !var3 = false || true = true ○ var2 || true = true || true = true ● true && var1 = true && true = true
(X > 3 || Y > 3) && Z < -3	False	● (X > 3 || Y > 3) = true || false = true ● Z < -3 = false ● true && false = false
(X+Z == 15) && (Y != 2)	True	● X+Z == 15: true ● Y != 2: true ● true && true = true

Ejercicio 4: Mayor o menor de edad

Diseña un programa Java denominado **PROG02_Ejerc4** que dada la edad de una persona, muestre un mensaje indicando si es mayor de edad. NO se puede utilizar el operador condicional **if**.

CÓDIGO

```
package prog02_ejerc04;

public class PROG02_Ejerc04 {

    public static void main(String[] args) {
        // Dada la edad de una persona
        int edad = 17;

        //Indicar si es mayor de edad sin utilizar el operador condicional if.
        String mayorMenorDeEdad = (edad >= 18) ? "mayor de edad" : "menor de edad";

        // Muestra un mensaje indicando si es mayor de edad.
        System.out.println ("Si tienes " + edad + " años eres " + mayorMenorDeEdad + ".");
    }
}
```

SALIDA POR CONSOLA

```
Si tienes 17 años eres menor de edad.
```

JUSTIFICACIÓN

- Como no podemos usar el operador condicional "if" usamos el operador condicional "?" el cual sirve para evaluar una condición y devolver un resultado en función de si es verdadera o falsa dicha condición. Es un operador ternario, por lo tanto necesita tres operandos para formar una expresión: condición ? exp1 : exp2.

Ejercicio 5: Mostrar tiempo

Diseña un programa Java denominado **PROG02_Ejerc5** que dado un número de segundos, muestre en pantalla cuántos minutos, horas y días contiene.

CÓDIGO

```
package prog02_ejerc05;

public class PROG02_Ejerc05 {

    public static void main(String[] args) {
        //Dado un número de segundos
        int segundosTotales = 36000456;

        //Calcular cuantos días son
        int dias = segundosTotales/86400;
        int restoDias = segundosTotales % 86400;

        //Calcular cuantas horas son a partir del resto de días
        int horas = restoDias/3600;
        int restoHoras = restoDias % 3600;

        //Calcular cuantos minutos son a partir del resto de horas
        int minutos = restoHoras/60;

        //Calcular los segundos que quedan
        int segundos = restoHoras % 60;

         //Muestra en pantalla cuántos minutos, horas, días y segundos contiene el número de segundos dado
         System.out.println (segundosTotales + " segundos son " + dias + " días, " + horas + " horas, " + minutos + " minutos y " + segundos + " segundos." );

    }
}
```

SALIDA POR CONSOLA

```
36000456 segundos son 416 días, 16 horas, 7 minutos y 36 segundos.
```

Ejercicio 6: Enumerado para razas de perros

Diseña un programa Java denominado PROG02_Ejerc6 que cree un tipo enumerado para las siguientes razas de perro: Mastín, Terrier, Bulldog, Pekines, Caniche y Galgo. El programa debe realizar las siguientes operaciones:

- Crea una variable denominada var1 del tipo enumerador. Asígnale un valor.
- Crea una variable denominada var2 del tipo enumerador. Asígnale un valor.
- Muestra por pantalla el valor obtenido de comparar ambas variables.

Investiga sobre la posibilidad de averiguar la posición que ocupa un determinado valor en el enumerado así como mostrar la cantidad de valores que contiene. Si lo consigues, muestra la posición de las dos variables en el tipo enumerado.

CÓDIGO

```
package prog02_ejerc06;

public class PROG02_Ejerc06 {

     public enum RazaPerros {Mastín, Terrier, Bulldog, Pekines, Caniche, Galgo} //Crea un tipo enumerado para las razas de perro
    public static void main(String[] args) {
         RazaPerros var1 = RazaPerros.Mastín; //Crea una variable denominada var1 del tipo enumerador. Asígnale un valor.
          RazaPerros var2 = RazaPerros.Galgo; //Crea una variable denominada var2 del tipo enumerador. Asígnale un valor.

        boolean sonIguales = var1 == var2; //Comparar ambas variables

        // Muestra por pantalla el valor obtenido de comparar ambas variables
          System.out.println("- Un perro " + var1 + " y un perro " + var2 + " ¿Son de la misma raza? " + sonIguales);

        //Posición que ocupa un determinado valor en el enumerado
              System.out.println("- Posición de " + var1 + " en el enumerado: " + var1.ordinal());
              System.out.println("- Posición de " + var2 + " en el enumerado: " + var2.ordinal());

        //Cantidad de valores que contiene el enumerado
```

```
                    System.out.println("- Cantidad de valores en el enumerado: " +
RazaPerros.values().length);
    }
}
```

SALIDA POR CONSOLA

```
- Un perro Mastín y un perro Galgo ¿Son de la misma raza? false
- Posición de Mastín en el enumerado: 0
- Posición de Galgo en el enumerado: 5
- Cantidad de valores en el enumerado: 6
```

Ejercicio 7: Alumnos matriculados

Diseña un programa Java denominado **PROG02_Ejerc7** que dados el número de alumnos matriculados en Programación, número de alumnos matriculados en Entornos de Desarrollo y número de alumnos matriculados en Base de datos. El programa deberá mostrar el % de alumnos matriculados en cada uno de los tres módulos. Se supone que un alumno sólo puede estar matriculado en un módulo. Trata de mostrar un solo decimal en los porcentajes.

CÓDIGO

```
public class PROG02_Ejerc07 {

    public static void main(String[] args) {
        // Número de alumnos matriculados
        int alumnosProgramacion = 25;
        int alumnosEntornosDesarrollo = 20;
        int alumnosBaseDatos = 30;
                int totalAlumnos = alumnosProgramacion + alumnosEntornosDesarrollo +
alumnosBaseDatos;

        //% de alumnos matriculados en cada uno de los tres módulos
               double porcentajeAlumnosProgramacion = (alumnosProgramacion * 100.0) /
totalAlumnos;
         double porcentajeAlumnosEntornosDesarrollo = (alumnosEntornosDesarrollo * 100.0) /
totalAlumnos;
        double porcentajeAlumnosBaseDatos = (alumnosBaseDatos * 100.0) / totalAlumnos;

        //Alumnos matriculados
        System.out.println ("Alumnos de Programación: " + alumnosProgramacion);
                    System.out.println ("Alumnos de Entornos de Dessarrollo: " +
alumnosEntornosDesarrollo);
        System.out.println ("Alumnos de Base de Datos: " + alumnosBaseDatos);
        System.out.println ("Total de alumnos: " + totalAlumnos);

        // % de alumnos en cada módulo con un solo decimal en los porcentajes
                        System.out.printf ("%% Alumnos de Programación: %.1f%%\n",
porcentajeAlumnosProgramacion);
                System.out.printf ("%% Alumnos de Entornos de Desarrollo: %.1f%%\n" ,
porcentajeAlumnosEntornosDesarrollo);
                    System.out.printf ("%% Alumnos de Base de Datos: %.1f%%\n" ,
porcentajeAlumnosBaseDatos);
```

```
    }
}
```

SALIDA POR CONSOLA

```
Alumnos de Programación: 25
Alumnos de Entornos de Dessarrollo: 20
Alumnos de Base de Datos: 30
Total de alumnos: 75
% Alumnos de Programación: 33,3%
% Alumnos de Entornos de Desarrollo: 26,7%
% Alumnos de Base de Datos: 40,0%
```