

# Solución de la tarea para PROG02.

---

## Ejercicios número 2, 3, 5-10

El código de los programas correspondiente a los ejercicios 2, 3, y del 5 al 10 se encuentra resuelto en el archivo:

[PROG02\\_SolucionTarea.zip](#) (108.00 KB)

## Justificación de los ejercicios 1, 2 y 4

1. Dados los siguientes identificadores, indicar si son válidos o no. Justificar respuesta.

- a. mi variable
- b. num\_de\_cte
- c. \_\_\_\_programa
- d. \$alguna
- e. 3tema
- f. cierto?
- g. númerodeCliente
- h. jose~
- i. año
- j. PI
- k. int

Debemos tener en cuenta que los identificadores están formados por letras y dígitos Unicode, y tienen que empezar por una letra, subrayado o el símbolo del \$, aunque este último se utiliza sólo de forma especial.

- a. Incorrecto porque tiene un espacio en medio
- b. Correcto, está formado por letras y por el carácter de subrayado
- c. Correcto, está formado por letras y por el carácter de subrayado
- d. Correcto, está formado por letras y por el símbolo de \$ que se considera también una letra
- e. Incorrecto, pues empieza por un número
- f. Incorrecto, porque el carácter de interrogación no se considera una letra
- g. Correcto, pues todos los caracteres son letras Unicode, aunque no es recomendable utilizar signos de puntuación a la hora de nombrar variables
- h. Incorrecto, pues el símbolo ~ no se considera una letra
- i. Correcto, pues la letra ñ forma parte del código Unicode
- j. Correcto, pues está formado por letras. Además, se trata de una constante pues está escrito en mayúsculas.
- k. Incorrecto, porque int es una palabra reservada y no se puede usar para nombrar variables.

2. Dado el siguiente programa, modifícalo para utilizar las variables que se indican. El tipo de dato elegido debe ser el de menos bits posibles que puedan representar el valor. Justifica tu elección.

```
public class ejerciciovariables {  
    public static void main(String[] args) {  
        boolean casado;  
        final int MAXIMO=999999;  
        byte diasem;  
        short diaaño;  
        char sexo;  
        long miliseg;  
        float totalfactura;  
        long poblacion;  
    }  
}
```

a. Si un empleado está casado o no.

**Boolean.** Sólo son posibles dos respuestas: Sí o No (Verdadero o Falso)

b. **Valor máximo no modificable: 999999.**

**Int.** Es el tipo de menos bits necesario para representar esa cantidad.

c. **Día de la semana**

**Byte.** Permite representar números enteros de -128 a 127, más que suficiente para guardar los valores del 1 al 7.

d. **Día del año.**

**Short.** Tenemos que utilizar un tipo que permita representar números del 1 al 366. El tipo byte no nos vale, por lo tanto el tipo más pequeño después del byte es short.

e. **Sexo: con dos valores posibles 'V' o 'M'**

**Char.** Con una sola letra podemos guardar los dos valores posibles.

También podemos representarlo como tipo enumerado, en ese caso la declaración de la variable iría fuera del main:

```
public enum sexo {V,M};
```

f. **Milisegundos transcurridos desde el 01/01/1970 hasta nuestros días.**

**Long.** Los milisegundos transcurridos hasta, por ejemplo, el 01/01/2015 alcanzará la cantidad de 1,42007E+12, la cual puede representarse en una variable de tipo **long**.

g. **Almacenar el total de una factura**

**Float.** Las cantidades en euros llevan decimales, por tanto, debemos utilizar un tipo de dato en coma flotante, y el más pequeño es **float**. No necesitamos el tipo **double** porque asumimos la posible pérdida de precisión en los decimales.

h. **Población mundial del planeta tierra.**

El tipo de dato **int** no podría representar la población mundial del planeta, ya que el valor máximo que alcanza es de 2.147.483.647. Si se desea representar el valor correspondiente a la población mundial del planeta (cerca de 7 mil millones de habitantes), tendríamos que utilizar un tipo de dato **long**.

**4. Indica los errores presentes en el siguiente código:**

```
1.    /
2.    operadoresaritmeticos.java
3.    Programa que muestra el uso de los operadores aritméticos
4.    /

5.    public class operadoresaritmeticos {
6.    public static main(String[] args)
7.    short x = 7;
8.    int y = 5;
9.    float f1 = 13.5;
10.   float f2 = 8f;
11.   System.out.println("El valor de x es ", x, " y el valor de y es ", y);
12.   System.out.println("El resultado de x + y es " + (x + y));
13.   System.out.println("El resultado de x - y es " + (x - y));
14.   System.out.printf("\n%s%s\n","División entera:", "x / y = ", (x/y));
15.   System.out.println("Resto de la división entera: x % y = " + (x % y));
16.   System.out.printf("El valor de f1 es %f y el de f2 es %f\n",f1,f2);
17.   System.out.println("El resultado de f1 / f2 es " + (f1 / f2))
18.   }
```

Instrucciones 1 y 4 falta el asterisco del comentario.

Instrucción 6 falta el void del main y la llave de comienzo del bloque.

Instrucción 9 falta colocar la "f" al final del valor, para indicar que es de tipo float.

Instrucción 11 las cadenas se concatenan con +.

Instrucción 14 no se muestra el resultado de x/y porque falta un %s en el primer argumento.

Instrucción 17: falta punto y coma al final.

Instrucción 18: falta la llave que cierra la clase operadoresaritmeticos.

