

Práctico II – String, IF, StringBuilder

Ejercicio 1

Considera la siguiente frase = “Hemos cosechado el doble de duraznos que el año anterior”

Responde (*resolviendo cada punto mediante código en Java*)

- ¿Cuál es la longitud, es decir, cuántos caracteres hay en esa frase?
- ¿Qué devuelve el método frase.charAt(9)?
- Escribe una instrucción que obtenga la letra *d* de duraznos en dicha frase.

Ejercicio 2

Dada la siguiente frase = “¿Eso que vi fue un auto o un gato?” ¿Qué se obtiene al hacer la siguiente instrucción frase.substring(12,23)? ¿Qué longitud tiene el resultado?

Ejercicio 3

Dada una frase ingresada por el usuario, determina su cantidad de letras e imprímela en pantalla.

Ejercicio 4

Dada una frase ingresada por el usuario, reemplaza todas las letras *t* que tenga, por letras *p*.

Ejercicio 5

Pedir una frase al usuario, investigar en la documentación de JAVA, lo que hace falta para hacer cada punto del ejercicio.

- Devolverla en mayúsculas.
- Devolverla en minúsculas.
- Devolver un texto que sea esa frase duplicada.
- Crear una nueva frase, donde se hayan reemplazado todas las vocales por espacios.
- Determinar si esa frase termina con un signo de admiración.
- Determinar si esa frase empieza con la letra H.

Ejercicio 6

Escribir programitas que calculen las siguientes expresiones matemáticas. Pedir los datos al usuario que corresponda (*para x, y, a, b, c, etc*) y mostrar los resultados en pantalla, como siempre.

a. $\frac{x}{y} + 1$

b. $\frac{x + y}{x - y}$

c. $\frac{b^2}{2}$

d. $(a + b) \frac{c}{d}$

e. $\frac{xy}{ab}$

$$f. \quad (x + y)^2(a - b)$$

Ejercicio 7

Dadas las siguientes expresiones booleanas, determine, en el papel (si hacer ningún código), cuál es el orden de evaluación de las mismas, donde X, Y, Z son expresiones que devuelven true o false y numero es una variable de tipo int.

- a. $X \parallel Y \ \&\& \ (\text{numero} > 0)$
- b. $((X \ \&\& \ Y) \ \&\& \ X \ \&\& \ Z) \parallel !Y$
- c. $\text{numero}=0 \parallel \text{numero}<>0 \ \&\& \ \text{numero}<23$
- d. $!Z \ \&\& \ Y$
- e. $(Z \ \&\& \ X) \parallel Y$
- f. $(Y \parallel Z \ \&\& \ Z \parallel X) \parallel !Z \ \&\& \ Y$

Ejercicio 8

¿Qué imprimirá en pantalla el siguiente programita?

```
public class MyClass
{
    public static void main(String[] args)
    {
        int x = 3;
        if(x < 3)
        {
            System.out.print("x");
        }
        else
        {
            System.out.print("y");
        }
    }
}
```

Ejercicio 9

¿Qué imprimirá en pantalla el siguiente programita?

```
public class MyClass
{
    public static void main(String[] args)
    {
        int a = 47;
        int b = 0;

        if(b == 0)
        {
            System.out.print("No se puede dividir por cero");
        }
        else
        {
            System.out.print("El resultado de dividir a por b es =" + a/b);
        }
    }
}
```

Ejercicio 10

Realizar un programita que pida un número y devuelva +1 si éste es positivo y -1 si éste es negativo.

Ejercicio 11

Realizar un programita que pida un número y devuelva +1 si éste es positivo y 0 si éste es negativo.

Ejercicio 12

Realizar un programita que pida dos números enteros y devuelva 1 si ambos números son iguales, y 0 sino.

Ejercicio 13

Realizar un programita que pida 3 números enteros n , a y b . Y determine si n se encuentra en el intervalo cerrado $[a, b]$ devolviendo verdadero o falso según corresponda.

Ejercicio 14

Realizar un programita que pida dos números y determine cuál de ellos es el mayor.

Ejercicio 15

Realizar un programita que pida dos números y determine cuál de ellos es el menor.

Ejercicio 16

Realizar un programita que pida un número entero y determine si es par o impar.

Ejercicio 17

Realizar un programita que pida 3 números enteros menores a 550 y determine si son los 3 iguales, o todos diferentes entre sí, o si, ni son iguales ni son todos diferentes.

Ejercicio 18

Realizar un programita que pida 3 números enteros e imprima en pantalla si están en orden creciente, o en orden decreciente, o no cumplen un orden.

Stringbuilder - teoría

StringBuilder es una clase que nos permite efectuar algunas operaciones en las cadenas de manera más eficiente.

Para crear un objeto StringBuilder podemos proceder de la siguiente forma:

```
StringBuilder stringBuilder = new StringBuilder();
```

Lo que hace potente a esta clase es la posibilidad de concatenar valores de distintos tipos, encargándose la clase de efectuar las conversiones por nosotros. Por ejemplo en la siguientes instrucciones:

```
StringBuilder buffer = new StringBuilder();  
buffer.append("En ").append(15).append(" minutos").append(" partimos.");  
System.out.println(buffer.toString());
```

Nos muestran en la consola: "En 15 minutos partimos."

Nótese que la clase hace el proceso de construcción por nosotros.

Además de esto, la cadena formada es única siendo esto una gran diferencia para el caso en que hubiéramos usado objetos Strings para cada palabra y luego concatenándolos.

Ejercicio 1 - SB

Los palíndromos son palabras que se leen de igual forma, ya sea de derecha a izquierda o viceversa.

Algunos ejemplos son:

Ana, Ata, rotor

Con la ayuda del método reverse de la clase StringBuilder verifique si las siguientes palabras son

palíndromos:

Anana

Sotos

Sus

Motor

Sometemos

Suertes

Ejercicio 2 - SB

Repita el proceso anterior pero indicando si los siguientes números son capicúa:

Int number1 = 7337;

Int number2 = 11311;

Int number3 = 113131;

Ejercicio 3 - SB

Con la ayuda del método replace de la clase Stringuilder proceda a modificar los strings según se indica a continuación

- "Nuestros" cambiar a "Nosotros"
- "JAVA Programming Language" cambiar a "JAVA Platform for Windows"
- "Poderío" cambiar a "Potrero"