Práctico 7: Introducción a Java

SIEMPRE:

- **En las entregas dejar documentados todos los archivos internamente con nombre y apellido, trabajo práctico nº y enunciado del ejercicio, al menos resumido.**
- **Comentar el código fuente a fin de dar una mayor legibilidad.**
- **La salida por pantalla debe contener toda la información necesaria para el usuario, a fin de que sea entendible el programa.**
- **Si utiliza algún concepto o comando investigado por su cuenta, siempre que entienda como funciona, muestre a la cátedra su programa, hará felices a los profes. Siempre que no sea, para evitar razonamientos, sino por la búsqueda de algo más óptimo o implementar una mejora en su programa respecto al problema original que se le haya pedido resolver.**

Ejercicios básicos

Se intentará comenzar con ejercicios más sencillos, aunque el nivel de complejidad puede ir escalando.

1. Realizar un programita que permita ingresar una palabra y luego dibuje la frase en pantalla en diagonal. Ejemplo: Si se ingresa "Hola"mostrará

```
H
 o
  l
   a
```

2. Crear un programita que determine el peso de una persona, en Marte. La gravedad en Marte es aproximadamente un 38 % de la gravedad en la Tierra. Luego de diseñado, re diseñarlo como una función que permita pasarle el peso de una persona como argumento, y retorne su peso en el planeta Marte como resultado.

3. Crear un programita que permita ingresar una edad y determine la cantidad de días que esa edad representa.

4. Determinar si un número es un **Número de Armstrong**, es decir, si dicho número es igual a la suma de sus dígitos *(cada uno además, elevado a la cantidad de digitos que tiene el número)*. Algunos ejemplos a continuación.
   9 si es un número Armstrong, ya que $9 = 9^1 = 9$
   10 no es un número Armstrong, ya que $10 \neq 1^2 + 0^2 = 1$
   153 si es un número Armstrong, ya que $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

5. Para una partida de un juego de rol, Dungeons & Dragons, cada jugador empieza por generar un personaje con el cual jugarán. Este personaje que debe generar, tiene que tener 6 habilidades, a saber: fuerza, destreza, constitución, inteligencia, sabiduria y carisma *(simplificando)*.
   Estas seis habilidades tienen cada una, un puntaje determinado de modo aleatorio. Esto se consigue tirando 4 dados de rol, de 6 caras. Y calculando la suma de los tres dados de mayor número *(descartando el dado menor)*. Este proceso de tirar los 4 dados y calcular dicha suma de los mayores, se repite para cada habilidad, hasta poder completar el personaje.
   Además, tu personaje tendrá una sangre o salud de 10 puntos iniciales sumado a un modificador que dependerá de la constitución de dicho personaje particular.
   Ese modificador se determina sustrayendo 10 de su constitución, dividiendo luego por 2, y redondeando hacia abajo.
   Teniendo en cuenta todos esos requisitos, crear un Generador de Personaje de Dungeons y Dragons, aleatorio.

6. Dibujar en pantalla una pirámide de caracteres X, dada la longitud de la base ingresada por el usuario. Ejemplo: el usuario ingresa 5, entonces deberá verse

   ```
   X
   XXX
   XXXXX
   ```

Manipulación de Cadenas de Caracteres

7. Permita ingresar una frase al usuario, y muestre las letras en posiciones alternadas, una si, otra no de la misma.

8. Permita ingresar una frase, y muestre 6 letras de posiciones aleatorias de la misma, siempre que la frase ingresada tenga al menos 10 caracteres de longitud.

9. Permita ingresar una palabra al usuario, y una letra. Reemplace las letras a que encuentre en la frase ingresada, por el símbolo de numeral.

10. Realizar un programita que permita ingresar una palabra al usuario, y muestre en pantalla todas sus consonantes, una por línea. Luego, diseñe una función que genere una cadena de caracteres formada por las consonantes de una palabra que recibe de argumento.

11. Realizar un programa que reemplace todos los espacios que encuentre en una frase, por guiones bajos. Este método lo puede nombrar *chauEspacios()*. Tanto espacios internos entre palabras, como espacios que hubiese al inicio o final del texto, deberán ser reemplazados.
    Considere implementarlo dentro de una clase pública llamada **ParsearTexto** *(como si fuese una librería)*, cuyos métodos internos, serán claro, públicos y static en su mayoría, salvo que necesite algún método auxiliar privado a la clase. Iremos agregando métodos que nos interesen en esta clase ParsearTexto.
    ¿Cómo deberá hacer en su programa principal, que será otra clase, para utilizar el método

chauEspacios? Investigue, primero piense ideas propias y haga pruebas. Antes de buscar en Internet o preguntar a los profes.

```
"hola     pepe" => "hola_____pepe"
```

12. Deberá crear un método en la clase ParsearTexto implementada anteriormente, llamado *kebabToCamel* que deberá saber convertir palabras escritas con la convención de nombrado de Kebab-Case a la convención que utiliza en particular Java, para el nombrado de variables, llamado CamelCase.

```
"a-bc" => "aBc"
"cant-pares" => "cantPares"
```

kebap-case o lisp-case: Las palabras son separadas con guiones y en minúscula. Muy utilizado en lenguajes funcionales y scripts de terminal.

camelCase: Las palabras son separadas con mayúsculas, salvo la primera letra que debe ir en minúscula. Muy utilizado en lenguajes orientados a objetos (Java por ejemplo).

snake_case: Es la convención que compone las palabras separadas por un guión bajo (underscore) en vez de espacios y con la primera letra de cada palabra en minúscula. (Python por ejemplo)

13. Deberá crear otro método, para ser agregado a la clase anterior ParsearTexto, llamado soloLetras, que limpiará de la frase, cualquier símbolo o caracter que no sea una letra del abecedario. Considere un conjunto de símbolos específico, grande pero no infinito.

```
"a$#.b" => "ab"
```

14. Crear un método que convierta una frase a su acrónimo. Es decir, debe recibir un nombre largo, como Portable Network Graphics y retornar PNG *(cualquier símbolo raro se puede omitir o quitar desde el principio, excepto los espacios que separan las palabras claro).*
Por ejemplo:
Ciudad Autónoma de Buenos Aires $->$ CABA
Banco Central de la República Argentina $->$ BCRA
A Todo Ritmo $->$ ATR
Nada Que Ver $->$ NQV

Arrays

15. Realizar un programita que pida al usuario 6 palabras, y las guarde en un vector. Luego pedirle una palabra y verificar si esta palabra existe en el vector o no.

16. Realizar un programita que dado un arreglo de 20 números, todos ordenados al azar, cada número siendo 1 o 0, ordene el mismo, poniendo todos los ceros a la izquierda del arreglo, y los unos a la derecha del mismo.
Ejemplo: $[1, 0, 0, 1, 0, 1, 1]$ quedaría como $[0, 0, 0, 1, 1, 1, 1]$

17. Dada la siguiente matriz cuadrada a continuación, escribir un programita que encuentre la suma de todos los elementos que no pertenecen a la diagonal principal de la misma.

$$A_{4\times4} = \begin{pmatrix} 9 & 5 & 0 & -3 \\ 7 & -2 & 8 & 1 \\ 3 & 5 & 7 & 8 \\ 6 & 3 & 0 & -1 \end{pmatrix}$$

18. Implementar un programa que genere una matriz de $10x10$ en cuya diagonal, aparezcan los números del 0 al 9. Y en el resto de la matriz, haya ceros. Adicionalmente, calcular la suma de los elementos de la diagonal y mostrar el resultado.

19. Generar una matriz de $6x6$ con valores numéricos al azar. Calcular el promedio de los números de las filas pares de la misma.

20. Dados los siguientes fragmentos de código, realice una versión equivalente para cada uno, utilizando el tipo de bucle while.

```
// fragmento 1
        int suma = 0;
        for (int i=0; i <500; i++){
            suma = suma + i;
            System.out.println(suma + " " + i);
        }

// fragmento 2
        int sillas = 0;
        for (int j=0; j<arraySillas.length; j++){
         if (arraySillas[j] == 1){
             sillas = sillas + 1;
            }
        }
        System.out.println(sillas);
```

21. Generar una matriz de $20x20$ con números cualesquiera. Encontrar el valor máximo de la matriz. Como es usual, no puede utilizar comandos predefinidos que obtengan dicho valor.

Array Dinámico

22. Puede investigar por su cuenta, luego de haber utilizado el array de Java en los ejercicios anteriores, el ArrayList *(Considerado un array dinámico, más parecido a una lista)*, haga un machete de los métodos que son importantes para manipular un ArrayList. Luego que tenga su machete de comandos, uso y declaración de una variable del tipo ArrayList, etc. Inicie la resolución de los ejercicios a continuación.

23. Pedir colores al usuario de los autos que ve pasar en su viaje de 3 horas en tren. Ir cargando los colores hasta que el usuario ingrese una letra *t* de terminar. El programa deberá entonces determinar cual fue el color de auto más común de ver, durante su viaje en tren.

24. Crear un método que permita modificar un array dinámico, de modo tal que, se le de un parámetro numérico *(int)* que servirá para saber cuántas veces deberemos repetir el array dinámico original que se nos ha dado.

```
Por ejemplo, si el array dinámico original es: ["a", "b", "c"]
llamar al método repetir(miarray1, 3)
daría como resultado un nuevo array dinámico:
["a", "b", "c", "a", "b", "c", "a", "b", "c"]
```

Preguntas de Revisión

25. ¿Qué significa que Java es un lenguaje de programación de alto nivel, orientado a objetos y de propósito general? ¿Qué diferencias tiene comparado con Python, y comparado con C++?

26. Nombra las principales características que diferencian al Lenguaje Java de otros lenguajes de programación.

27. ¿Qué es el bytecode en Java?

28. ¿Qué es la máquina virtual de Java (JVM)? ¿Qué ventajas y desventajas posee? Investigue.

29. ¿Qué son los paquetes en Java? ¿Cuál es su utilidad?
(import nombreDelPaquete.nombreDeLaClase;)

30. Investigue como se pueden organizar los archivos en un proyecto de Java. Paquetes. Directorios, clases, etc.

31. ¿Qué son las excepciones en Java? ¿Para qué sirven? ¿Y en Python, qué son, recuerda? Investique.

32. Escriba en su cuaderno, un listado de los comandos básicos de Java, y sus carácteristicas como lenguaje, principales. Repita para C++ y para Python. Analice las similitudes y las diferencias entre los 3 lenguajes. Hágalo prolijo, los deberá entregar aparte.