Tarea UD 3. Clases y objetos. POO. Herencia. Arrays (I).

INTRODUCCIÓN

WORDLE

La siguiente tarea, con el fin de realizar un trabajo constante y resolver las dudas sin lagunas temporales, será realizada en 3 ***entregas periódicas****, semanales, y con* ***hitos****. De este modo evitaremos atracones finales para entregar la tarea e iremos añadiendo dificultad y nuevos conceptos a medida que vamos trabajando en la tarea.*

En la primera parte trabajaremos con ***enumeraciones****, creación de* ***clases*** *básicas y sobreescritura de métodos de Object (toString,..), además de hacer uso de conceptos ya aprendidos en unidades anteriores.*

El resultado final de esta tarea será ***la implantación básica de un juego de Wordle por consola/terminal****. Para los que no conozcan el juego, pueden consultar las siguientes páginas:*

https://es.wikipedia.org/wiki/Wordle

https://www.nytimes.com/games/wordle/index.html (implantación del juego en el New York Times).

El juego consiste en adivinar una palabra de 5 letras, en la que el jugador tiene 6 intentos para realizarlo.

Cada jugador debe introducir una palabra de 5 letras válida en cada intento.

El color de cada letra en los sucesivos intentos indica cuán aproximado está el jugador de la palabra a adivinar:

- *Si la letra introducida coincide con la posición de la palabra a adivinar aparecerá en VERDE.*
- *So la letra introducida aparece en la palabra a adivinar, pero en otra posición aparecerá en AMARILLO.*
- *Si la letra introducida no aparece en la palabra a adivinar aparecerá en GRIS.*

W is in the word and in the correct spot.

I is in the word but in the wrong spot.

U is not in the word in any spot.

COLORES POR CONSOLA/TERMINAL

*Para imprimir las letras por consola de comandos, existe un **estándar de códigos de escape ANSI X3.64** (equivalente a ECMA-48 ISO/IEC 6429, FIPS 86, JIS X 0211) (https://es.wikipedia.org/wiki/C%C3%B3digo_escape_ANSI), que permite dar salida en un terminal de texto de vídeo o emuladores de terminal, permitiendo **controlar la localización del curso, el color, estilo del tipo de letra y otras opciones**.*

Existe una secuencia de escape que el terminal interpreta como una secuencia de comandos y no de texto, permitiendo, entre otras cambiar el color del texto o de fondo. Para deshacer el formato se precisa enviar un código de RESET.

Por ejemplo, el siguiente texto se imprime en rojo (en separado los códigos para que podáis diferenciar el texto de los códigos de escape):

```
System.out.println("\033[41m" + "EN VERMELLO" + "\033[0m" + " y NORMAL");
```

Para los terminales que admiten color verdadero (24 bits) el formato de los caracteres de escape es el siguiente:

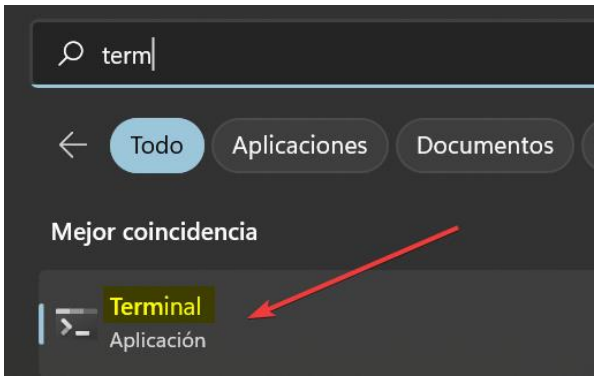
"\033[38;2;r;g;b" para color frontal (de texto).
"\033[48;2;r;g;b" para color de fondo.

*Siendo **r, g, b los niveles de rojo, verde y azul**, respectivamente. Sus valores van de 0 a 255.*

*Por ejemplo: **"\033[48;2;201;180;88"** pondría un color de fondo amarillo. Una vez puesto hay que cambiar el color de fondo o hacer un "reset"*

Un ejemplo de salida por consola de Windows cambiando el color de fondo:

El estándar ANSI lo admiten muchos terminales (incluso el terminal de Windows:



No así la consola de órdenes del sistema (cmd.exe), para poder emplear colores se precisa instalar ANSICON:

http://adoxa.altervista.org/ansicon/

Si usas Windows precisas descargar la versión de 64 bits (o 32), descomprimirla e instalarla desde línea de órdenes:

```
C:\directorioDescarga\ansi140\x64> ansicon -i
```

Una vez hecho, debes abrir de nuevo una consola y podrás ver los colores.

Referencias:

http://adoxa.altervista.org/ansicon/
https://en.wikipedia.org/wiki/ANSI_escape_code#8-bit
https://www.baeldung.com/java-log-console-in-color

GENERACIÓN DE NÚMEROS ALEATORIOS

La generación de números aleatorios es muy útil en programación, tanto para simulaciones, generar casos de prueba, juegos, solución de problemas, etc. Aunque en realidad no son números aleatorios, son números pseudo-aleatorios basados en una "semilla" que suele ser el número de milisegundos en el instante de la ejecución.

En Java, el generador de números aleatorios, que también está en el mismo paquete, *java.util*, por lo que puede importarse todo el paquete java.util.* o sólo la clase *Random*.

```
Random r = new Random();
int numero = r.nextInt(100);
```

Que genera un **número entero aleatorio** entre 0 y 99, por ejemplo.

Para **generar números *double* aleatorios entre 0 y 1.0**, que es lo que haremos en el primer ejercicio, tenemos el método ***nextDouble()*** de Random:

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Random.html#nextDouble()

Además, en la clase Math existe un método estático similar para generar números ***double*** aleatorios entre 0 y 1.0, similar al de la clase *Random*, pero que no es más que un "alias" (o casi) del método anterior:

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Math.html#random()

Pues en la implementación de Math podréis ver algo como esto:

```
private static final class RandomNumberGeneratorHolder {
        static final Random randomNumberGenerator = new Random();
}

public static double random() {
  return RandomNumberGeneratorHolder.randomNumberGenerator.nextDouble();
}
```

WordleColor (enumeración)

WordleColor es una enumeración con los posibles colores válidos del Wordle: GRAY, GREEN, RESET, VOID, YELLOW.

Atributos

Tres atributos: **r, g, b** de tipo entero con los niveles de rojo, verde y azul.

Constantes de la enumeración

GRAY
Fondo gris con valores r, g, b: 120, 124, 126.
GREEN
Fondo verde, con niveles r, g, b: 0, 170, 100.
RESET
Fondo negro, con niveles r, g, b: 0, 0, 0.
VOID
Fondo claro (vacío), con niveles r, g, b: 180, 180, 180.
YELLOW
Fondo amarillo, con niveles r, g, b: 201, 180, 88.

Constructor

Constructor que recoge los tres atributos: r, g, b.

Métodos de instancia

getAnsiCodeFromRGB():

Devuelve una cadena con el valor del código escape de color de fondo ANSI para ese color: el valor devuelto es ***"\033[48;2;r;g;b"***, siendo r, g, b el valor de los atributos.

getRed(), getGreen(), ***getBlue()***: devuelven, respectivamente, los niveles de color r, g, b.

WordleLetter (clase)

public class ***WordleLetter*** *extends* *Object*

WordleLetter es la clase que representa **la letra palabra que se muestra en el juego** de Wordle y que puede tener 4 posibles colores: GREEN, YELLOW, GRAY, VOID (RESET se usa para reiniciar el color).

Atributos

- ***caracter***: representa la letra de la palabra, que puede ser 0 (valor por defecto) si no tiene valor.
- ***cor***: de tipo *WordleColor*, representa el color de fondo con el que se muestra ese carácter. El color de la letra puede ser: GRAY, si no aparece en la palabra; YELLOW si aparece, pero en otra posición; GREEN si aparece en esa posición; VOID si no tiene carácter o si no se ha comprobado.

Constructores

Constructor por defecto:

Crea una *WordleLetter* por defecto con el carácter por defecto (0) y el color como *WordleColor*.*VOID*.

Constructor que recoge el carácter:

Construye una WordleLetter vacía (color WordleColor.VOID) y con el carácter recogido como argumento. Debe comprobar que es una letra. Si no es una letra debe dejar el carácter por defecto: 0. Si puedes, invoca al constructor que recoge dos argumentos: this(caracter, WordleColor.VOID);.

Constructor que recoge el carácter y el color:

Construye una WordleLetter que recoge el color (WordleCOlor) y con el carácter recogido como argumento. Debe comprobar que es una letra. Si no es una letra debe dejar el carácter por defecto: 0.

Métodos

Get/set para cada atributo. Haz las comprobaciones que consideres oportunas.

isVoid(): indica si la letra está vacía. Devuelve verdadero cuando la letra tiene un valor vacío (es el carácter por defecto (0)).

toString(): devuelve la cadena que representa a la letra en mayúscula con el color de fondo que tiene asignado. Esto es: la concatenación del carácter de escape ANSI para ese color (método getAnsiCodeFromRGB()), la letra es mayúscula y el carácter de escape ANSI para el RESET (debe reiniciarse el color de la consola).

IMPORTANTE: si no sabes cómo hacerlo, haz que si el color es GREEN ponga la letra en mayúscula, si es amarillo en minúscula y si no está debe poner un asterisco.

AppGWordle (aplicación)

Clase de la aplicación **que pida una palabra y muestre las letras de esa palabra por consola, dándole un color aleatorio entre los colores válidos**.

Haz que pida la palabra cada vez hasta que se escriba una línea vacía.

Además, crea un archivo ***jwordle.jar*** que permita ejecutarlo.

ENTREGA

Genérese un único documento **PDF con el código del programa y las capturas de compilación y ejecución.** Además, debe proporcionar el **código fuente, *WordleColor.java*, *WordleLetter.java* y archivo *jwordle.jar*** para poder ser ejecutado, tal y como se muestra en el ejemplo anterior.

Entréguese un archivo comprimido de extensión ZIP o 7Z con:

- PDF con formato *Apellido1Apellido2NombreTarea0301PROG.pdf*
- Código fuente: ***WordleColor.java*, *WordleLetter.java***
- Archivo comprimido para ejecutar: ***jwordle.jar***.

El formato de documento debe ser:

*Apellido1Apellido2NombreTarea0301PROG.zip (o *.7z)*

Por ejemplo:

WittgensteinLudwigJosefTarea0301PROG.zip (sólo tiene un apellido)