

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web				
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14			Profesor Juan Antonio Jiménez Morales	
Curso lectivo 2018 / 2019	Grupo 1º DAW	Tipo de documento Examen	Trimestre Primero – Control 2	Modelo Único	Fecha 12/12/2018	Pág. 1/4

INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por “Ex” seguido del número de lista, seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregará una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- ➔ En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el turno. También debe indicar una breve descripción de lo que hace el programa.
- ➔ Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class**.

EJERCICIOS

1. [3 puntos] Implemente el juego del ahorcado. Para ello, deberá tener en cuenta lo siguiente:
 - a) El programa tendrá inicialmente almacenadas en memoria una batería de 10 palabras (que ud. mismo elegirá), las cuales estarán guardadas en un array de Strings. Las palabras estarán en mayúsculas.
 - b) El programa deberá elegir aleatoriamente una de esas palabras, la cual almacenará, carácter a carácter, en un array de caracteres. Ejemplo:

'H'	'O'	'L'	'A'
-----	-----	-----	-----

- c) Al usuario se le mostrará una “palabra” con las letras ocultas, presentando, en su lugar, el carácter ‘_’ (subrayado o “guión bajo”). Para el caso de la palabra “HOLA”, mostrará, inicialmente, algo similar a:

_ _ _ _

Esta “palabra” con letras ocultas estará almacenada en otro array, en el que se irá reflejando el estado “actual” de letras descubiertas de la palabra.

- d) El programa pedirá reiteradamente la introducción de una letra, y comprobará si la letra está en la palabra a ser adivinada (entenderemos esto como un intento). Si está, habrá que “hacer visibles” las ocurrencias de dicha letra. Si no está, contabilizará un fallo. En cualquier caso, tras cada intento, el programa informará al usuario del “estado” de letras adivinadas hasta el momento (en sus posiciones). El programa tendrá que asegurarse de que procesa la letra introducida en mayúsculas.
 - e) El programa parará cuando se de alguna de las siguientes circunstancias:
 - i. Se ha completado la palabra. El usuario habrá ganado, y habrá que informar de esto.
 - ii. Se ha alcanzado el tope de fallos permitidos, que será de 5. El usuario habrá perdido, y habrá que informar de esto.
 - iii. El usuario introduce el carácter ‘0’ (número cero). El programa interpretará esto como que el usuario abandona el juego. Ojo: En este caso, no se procesa como un intento: bastará con salir del programa.

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web				
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14			Profesor Juan Antonio Jiménez Morales	
Curso lectivo 2018 / 2019	Grupo 1º DAW	Tipo de documento Examen	Trimestre Primero – Control 2	Modelo Único	Fecha 12/12/2018	Pág. 2/4

Puede utilizar las siguientes funciones para implementar el juego del ahorcado:			
Función	Argumentos	Qué hace	Devuelve
eligePalabra	<ul style="list-style-type: none"> Array de Strings 	Pasa la palabra elegida, carácter a carácter, a un array de chars, reservando la cantidad de memoria “precisa” para ello	Array de chars con las letras de la palabra elegida
presentaPalabra	<ul style="list-style-type: none"> Array de chars 	Presenta por pantalla los caracteres del array de chars	Nada
longitudPalabra	<ul style="list-style-type: none"> Array de chars 	Calcula el n.º de caracteres (longitud) del array de chars	La longitud del array de chars
palabraEnBlanco	<ul style="list-style-type: none"> Número de letras 	Almacena en un array de chars tantos caracteres ‘_’ (subrayado o “guión bajo”) como se indique en el argumento n.º de letras, reservando la cantidad de memoria “precisa” para ello	Array de chars “en blanco”
palabraDescubierta	<ul style="list-style-type: none"> Array de chars 	Comprueba si en el array de chars quedan letras por descubrir (caracteres ‘_’)	Verdadero todas las letras han sido descubiertas; falso en caso contrario
destapaLetra	<ul style="list-style-type: none"> Array de chars A Array de chars B Una letra 	Uno de los arrays contendrá la palabra a ser averiguada. El otro, la palabra con las letras “averiguadas hasta el momento”. Si la letra está en el array de la palabra a ser averiguada, la “hará visible” en esa misma posición en el array de la palabra con las letras “averiguadas hasta el momento”	Verdadero si la letra estaba en la palabra; falso en caso contrario. Además, los cambios que se hayan realizado en el array de la palabra con las letras “averiguadas hasta el momento”

- Se pueden usar las funciones de String que considere oportunas para extraer los caracteres de los Strings, para calcular su longitud (n.º de caracteres) y para convertir a mayúsculas.

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web				
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14			Profesor Juan Antonio Jiménez Morales	
Curso lectivo 2018 / 2019	Grupo 1º DAW	Tipo de documento Examen	Trimestre Primero – Control 2	Modelo Único	Fecha 12/12/2018	Pág. 3/4

2. [3 puntos] Elija una de las opciones:

OPCIÓN A: Implemente un comprobador de números de un “sorteo de la lotería”. En este sorteo, se van a dar los siguientes premios:

Primer premio: 1 número	1000 €
Segundo premio: 1 número	500 €
Tercer premio: 1 número	250 €
Pedreas: 10 números	50 €
Reintegros: Números que acaben o empiecen con la última cifra-dígito del primer premio:	5 €

Los números que juegan oscilan entre el 1 y el 9999. El programa obtendrá aleatoriamente los números premiados (los cuales no podrán repetirse). A continuación, pedirá reiteradamente al usuario que introduzca números, informando para cada uno de ellos si tiene premio y su cuantía. El programa finalizará cuando se introduzca el número 0 (cero), mostrando entonces por pantalla la “tabla” completa con los números premiados (también se informará de la cifra-dígito de reintegros). Los premios no son acumulativos; sólo se cobra el premio de mayor cuantía. Y el reintegro se cobra sólo una vez.

→ Debe implementar, al menos, las siguientes funciones: (a) una función que genere los números premiados sin que se repitan, (b) otra que sirva para comprobar el premio (cuantía) que recibe un número que se le pase como argumento, y (c) otra función que presente por pantalla la lista de los números premiados, además de la última cifra-dígito del primer premio (correspondiente a los reintegros). Debe decidir los argumentos y los valores que, en su caso, cada función necesite.

OPCIÓN B. Implemente el juego del tres en raya para dos jugadores. El juego terminará cuando uno de los jugadores haya conseguido poner en línea tres de sus fichas, o cuando ya no sea posible realizar ninguna línea de tres fichas (por parte de ambos jugadores), informando convenientemente al usuario del motivo del fin del juego: gana uno de los jugadores -indicando quién- o empate. El programa irá solicitando reiterada y alternativamente a cada jugador la posición (fila, columna) donde quiere colocar la ficha. Si la posición está ocupada, el programa debe volver a solicitar al mismo jugador una nueva posición, advirtiéndole que la casilla estaba ocupada.

→ Para la implementación de este ejercicio, debe apoyarse en el uso de las siguientes funciones: (a) función para presentar por pantalla el estado del tablero de juego, enmarcado y con la numeración de filas y columnas, (b) función que chequee si un jugador ha ganado (ha hecho una línea), y (c) función que chequee si no es posible ganar por parte de un jugador. Debe decidir los argumentos y los valores que, en su caso, cada función necesite.

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web				
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14			Profesor Juan Antonio Jiménez Morales	
Curso lectivo 2018 / 2019	Grupo 1º DAW	Tipo de documento Examen	Trimestre Primero – Control 2	Modelo Único	Fecha 12/12/2018	Pág. 4/4

3. [2 puntos] Elija una de las opciones:

OPCIÓN A. Hacer un programa para visualizar los números perfectos que hay entre el número 2 y un número introducido desde teclado. Un número perfecto es un número natural que es igual a la suma de sus divisores propios positivos. Se denomina factor o divisor propio de un número entero n, a otro número también entero que es divisor de n, pero diferente de n.

Así, 6 es un número perfecto porque sus divisores propios son 1, 2 y 3; y $6 = 1 + 2 + 3$. Los siguientes números perfectos son 28, 496 y 8128.

→ Para la implementación de este ejercicio, debe apoyarse en el uso de funciones. Estas funciones deberán estar integradas en un paquete, y deben ser, al menos, las siguientes: (a) función para saber si un número es divisor propio de otro, (b) función que devuelva la suma de los divisores propios de un número, y (c) función que determine si un número es perfecto. Debe decidir los argumentos y los valores que, en su caso, cada función necesite.

Recomendación: utilice “enteros largos” para poder manejar valores enteros elevados.

OPCIÓN B. Hacer un programa para visualizar los números curiosos que hay entre el número 2 y un número introducido desde teclado. Un número es curioso si su cuadrado tiene al propio número como últimas cifras. Así, el número 25 es curioso ya que su cuadrado es 625, también es curioso 76 porque su cuadrado es 5776

→ Para la implementación de este ejercicio, debe apoyarse en el uso de funciones. Estas funciones deberán estar integradas en un paquete, y deben ser, al menos, las siguientes: (a) función que devuelva el último dígito de un número -las unidades-, (b) función que, dado un número, devuelva otro número idéntico, pero sin la cifra de las unidades, y (c) función para saber si un número es curioso. Debe decidir los argumentos y los valores que, en su caso, cada función necesite.

Recomendación: utilice “enteros largos” para poder manejar valores enteros elevados.

4. [2 puntos] Escriba un programa que solicite al usuario números enteros comprendidos entre los valores 1 y 100, y que genere un histograma con las frecuencias de cada número entero introducido. Si el valor introducido está fuera del rango [1,100], se informará al usuario que el valor no es válido y será ignorado, volviéndosele a solicitar un nuevo valor, salvo en el caso del valor -1, que servirá para finalizar la captura de enteros. Para representar las barras del histograma utilice secuencias del carácter '*'. Por ejemplo, la secuencia: 1, 1, 20, 1, 2, 20, 3, 3, 3, 4, 4, 4, 33, 3, -1 generaría la siguiente salida:

```

1:    ***
2:    *
3:    ****
4:    ***
20:   **
33:   *
```

Si lo considera oportuno, utilice funciones para implementar este programa.