

Tarea4

Nombre alumno o alumna: Podadera González, Andrés Samuel

Fecha de entrega: miércoles, 14 de diciembre de 2022, 16:29

Retroalimentación global de la actividad:

Hola, Andrés.

La tarea está excelentemente resuelta. ¡Enhorabuena! Tan solo debes revisar algunos detalles que te he indicado en las anotaciones de corrección, especialmente en el ejercicio 2, donde se te ha "escapado" una cosa importante.

Una vez visto eso, puedes descansar un poco y esperar a la semana que viene para cuando se abra la unidad 5 y puedas empezar a implementar tus propias clases.

Hasta pronto y enhorabuena de nuevo.

Puntuación orientativa de la actividad: **9,16**

RESULTADOS DE APRENDIZAJE EVALUADOS EN ESTA ACTIVIDAD

RA6 Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

		Punt. Máx.	Punt. Obt.	Punt. Final (sobre 10)
RA6.a	Se han escrito programas que utilicen arrays.	66	62,00	9,4
RA6.g	Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.	17	15,00	8,8

CORRECCIÓN POR EJERCICIO/ACTIVIDAD/ELEMENTO

Ejercicio 1. Ofuscación de contraseñas.

Puntos de control (elementos evaluables)		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Declaración del array de contraseñas de entrada.	2	2,0	RA6.a					
PC2	Reserva de espacio para el array de resultados, sin uso de literales.	2	2,0	RA6.a					
PC3	Recorrido de cada uno de los elementos del array de contraseñas.	3	3,0	RA6.a					
PC4	Generación de la contraseña ofuscada.	2	2,0	RA6.a	RA6.g				
PC5	Almacenamiento de contraseña ofuscada en el array de resultados.	3	3,0	RA6.a					
PC6	Recorrido del array de resultados para mostrarlo por pantalla con el formato solicitado.	3	3,0	RA6.a					
PC7	El programa funciona correctamente, no le falta ningún elemento necesario y no incluye elementos extraños, innecesarios o sin sentido.	2	2,0	RA6.a					

Retroalimentación:

Correcto. Muy bien.

Ejercicio 2. Validación de contraseñas.

Puntos de control (elementos evaluables)		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Declaración y reserva de memoria del array de resultados, sin uso de literales.	2	2,0	RA6.a					
PC2	Expresión regular que sirva como patrón para la detección, de al menos, un carácter especial.	1	1,0	RA6.g					
PC3	Expresión regular que sirva como patrón para la detección, de al menos, una letra mayúscula.	1	1,0	RA6.g					
PC4	Expresión regular que sirva como patrón para la detección de, al menos, una letra minúscula.	1	1,0	RA6.g					
PC5	Expresión regular que sirva como patrón para la detección de, al menos, un dígito.	1	1,0	RA6.g					
PC6	Expresión regular que sirva como patrón para la detección de los caracteres no permitidos.	1	0,0	RA6.g					
PC7	Utilización apropiada de mecanismos gestión de expresiones regulares en Java.	2	2,0	RA6.g					
PC8	Combinación apropiada de todos los patrones permitidos y no permitidos para decidir si se trata de una contraseña válida.	4	4,0	RA6.g					
PC9	Almacenamiento de los resultados en el array de resultados.	3	3,0	RA6.a					
PC10	Recorrido del array de resultados para mostrarlo por pantalla con el formato solicitado.	3	3,0	RA6.a					
PC11	El programa funciona correctamente, no le falta ningún elemento necesario y no incluye elementos extraños, innecesarios o sin sentido.	2	1,0	RA6.a	RA6.g				

Retroalimentación:

Has utilizado las “assertions” o “aserciones” de tipo “look ahead”, “look behind”, etc. usando ?= (búsqueda anticipada), que no llegamos a ver en este curso, pero que tú has buscado por tu cuenta y te funcionan bien, así que por mí perfecto.

El problema es que **no funciona correctamente para la parte de los caracteres no válidos. Si la contraseña contiene un carácter no válido** como por ejemplo ‘^’, ‘ñ’ o ‘ç’, **el programa la admite como válida siempre que cumpla el resto de condiciones** (letra mayúscula, letra minúscula, dígito, carácter especial y longitud). De hecho, deberías haberlo notado en el último ejemplo de password no válido (ejemplo 13: “[1234567890Aa^]”), que tu programa sí lo da como válido, cuando no lo debería ser porque contiene el carácter ‘^’, que no está en la lista de los permitidos. Imagino que habrá sido un despiste y no te has dado cuenta.

Ejercicio 3. Cifras gigantes.

Puntos de control (elementos evaluables)		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Declaración del array bidimensional con los modelos ASCII de las cifras.	1	1,0	RA6.a					
PC2	Entrada de datos con filtro de número entero no negativo de como máximo diez cifras mediante el uso de expresiones regulares.	2	2,0	RA6.g					
PC3	Implementación de un mecanismo apropiado para la manipulación y uso del array bidimensional con los modelos ASCII de las cifras.	5	5,0	RA6.a					
PC4	Implementación de un mecanismo apropiado para la generación de una cadena con el resultado final buscado basada en el uso fragmentario del array bidimensional con los modelos ASCII de las cifras.	5	5,0	RA6.a					
PC5	El programa funciona correctamente, no le falta ningún elemento necesario y no incluye elementos extraños, innecesarios o sin sentido.	2	2,0	RA6.a					

Retroalimentación:

Correcto.

Ejercicio 4. Tabla de días de la semana por mes (I): cantidades.**Puntos de control (elementos evaluables)**

		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Declaración y reserva de espacio de la tabla (array bidimensional) para almacenar el resultado (array de tamaño 12x7).	1	1,0	RA6.a					
PC2	Implementación de un mecanismo apropiado para el relleno de la tabla con la información solicitada (cuántos días de la semana hay en cada mes). Puede implicar el uso de bucles, arrays, objetos LocalDate, contadores, recorridos, etc.	10	8,0	RA6.a					
PC3	Recorrido del array de resultados para mostrarlo por pantalla con el formato solicitado, con sus cabeceras de fila y de columna y con la información correcta.	2	2,0	RA6.a					
PC4	El programa funciona correctamente, no le falta ningún elemento necesario y no incluye elementos extraños, innecesarios o sin sentido.	2	2,0	RA6.a					

Retroalimentación:

Funciona, pero es innecesario un switch para distinguir cada día de la semana. ¿Se te ocurre cómo podría evitarse ese switch para dejar un código más simple y compacto?

Ejercicio 5. Tabla de días de la semana por mes (II): lista de días.**Puntos de control (elementos evaluables)**

		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Declaración y reserva de espacio de la tabla (array tridimensional) para almacenar el resultado (array de tamaño 12x7xdesconocido).	1	1,0	RA6.a					
PC2	Reserva de espacio para los subarrays de la tercera dimensión basándose en los cálculos obtenidos en el ejercicio anterior (cuántos días de la semana hay en cada mes).	2	2,0	RA6.a					
PC3	Implementación de un mecanismo apropiado para recorrer cada día del año y, en función del mes y del día de la semana que se trate, incluir ese día en el subarray (tercera dimensión) correspondiente en la posición apropiada.	4	3,0	RA6.a					
PC4	Recorrido del array de resultados para mostrarlo por pantalla con el formato solicitado, apropiadamente tabulado, con sus cabeceras de fila y de columna y con la información correcta.	2	2,0	RA6.a					
PC5	El programa funciona correctamente, no le falta ningún elemento necesario y no incluye elementos extraños, innecesarios o sin sentido.	2	2,0	RA6.a					

Retroalimentación:

Nuevamente, ¿necesitas realmente esos switch?

¿Necesitas realmente las variables contadorLunes, contadorMartes, contadorMiercoles, etc.? ¿Se te ocurre alguna manera de intentar evitarlas?