

Rec. 1ra Evaluación de Aprendizajes

Programación Avanzada (1113) 2do Cuatrimestre 2020

La presente evaluación tiene como objetivo desarrollarse dentro del período comprendido entre el Martes 3 de Noviembre a las 8:00 am y el Miércoles 4 de Noviembre a las 23:59. Se otorga una ventana que contempla los eventuales inconvenientes que podrían tener los estudiantes al momento de resolverla.

Para consultas públicas sobre consignas, utilizar el foro de discusión. No se atenderán consultas individuales a menos que se acuerde eso mediante foro.

Forma de entrega

Por favor, leer atentamente el README que acompaña al proyecto. Deberás entregar tanto el PDF que generarás como el código de respaldo.

1. Resolución de ejercicio OIA

Resuelva el ejercicio de la OIA “[Instalando aplicaciones](#)”, adjunto a este enunciado. La resolución debe ser completa, correcta, ordenada a objetos, prolija y cumplir con toda la metodología planteada por la cátedra.

El código, lote de pruebas, y la documentación de la resolución del ejercicio debe entregarla en un zip de respaldo y también subirlo a la plataforma MIEL según se ha especificado anteriormente.

2. Ejercicio básico Java

Dado un lienzo en blanco de tamaño $(n) \times (2n - 1)$, donde la primera dimensión es el alto y la segunda, el ancho, se desea pintar con el siguiente patrón de color negro, sabiendo que n es impar y positivo.

Ejemplo para $n = 5$:

	0	1	2	3	4	5	6	7	8
0	X				X				X
1		X		X		X		X	
2			X				X		
3		X		X		X		X	
4	X				X				X

Ejemplo para $n = 3$:

	0	1	2	3	4
0	X		X		X
1		X		X	
2	X		X		X

Deberá realizarse utilizando **dos estructuras de iteración como límite** (ya sean for, while o foreach). Si el parámetro de entrada no es válido, deberá arrojarse una **excepción tipo RuntimeException**.

Especificar la **complejidad computacional** del problema.

3. Ejercicio Complejidad Computacional.

Se realiza una modificación al algoritmo de ordenamiento por burbujeo para que al terminar una pasada, la siguiente la haga en el sentido contrario. Dicho algoritmo se llama Cocktail Sort.

- a. ¿Cuál es la complejidad de dicho algoritmo? Justificar.
- b. Si se sabe que el algoritmo tardó 30 segundos en ordenar 30.000 elementos:
 - 1. ¿Cuánto tardaría en ordenar el triple de elementos?
 - 2. ¿Cuántos elementos podría ordenar en el triple de tiempo?
 - 3. ¿Cuánto tardaría en ordenar el triple de elementos en una máquina 3 veces más rápida?

4. Ejercicio con objetos Java

Se desea programar un método que elimine repetidos de un arreglo de Strings.

El resultado debe proporcionarse en forma de arreglo de Strings.

Se busca una complejidad que sea menor a $O(n^2)$

Especificar la complejidad computacional del problema.

5. Multiple Choice

a. Si ejecuto múltiples y variadas pruebas sobre mi software y todas dan el resultado esperado (positivo):

- a. Podemos garantizar que el software no tiene errores
- b. Podemos garantizar que el software tiene errores
- c. No podemos garantizar que el software tenga o no errores

b. El método equals:

- a. Permite comparar variables
- b. Es necesario definirlo para cada clase
- c. El IDE lo define automáticamente
- d. Permite comparar objetos

c. Una clase abstracta:

- a. Debe tener al menos un método abstracto
- b. Debe tener atributos abstractos
- c. No puede heredar de ninguna otra clase
- d. Puede no tener ningún método abstracto
- e. Ninguna de las anteriores