



Examen #2, II-2020
Miércoles 25 de noviembre del 2020
Prof. Maureen Murillo

Indicaciones generales

Relacionadas con la entrega:

1. Cuenta con 2:45 horas para desarrollar la solución del examen.
2. Una vez transcurrido este tiempo tendrá 15 minutos adicionales para subir un archivo .jar (cuyo nombre de archivo sea el nombre y el apellido suyo) a mediación virtual. El enlace de mediación virtual se cerrará a las 10:00am en punto. Note que estos 15 minutos adicionales se otorgan para solventar cualquier problema técnico para la entrega del archivo. Queda a su criterio si utiliza parte de este tiempo para seguir programando, quedando bajo su responsabilidad cualquier problema que se le presente para subir el archivo (falla eléctrica, falla de internet, problemas con mediación, etc).
3. El proyecto entregado en mediación virtual debe compilar, tal como está establecido en la carta al estudiante. De no cumplir con esta característica se considerará como no entregado y tendrá un 0 en el examen.
4. El desarrollo completo del examen (desde el inicio hasta el fin) deberá realizarse utilizando el repositorio personal de Github creado para este curso, en la carpeta de exámenes. Este repositorio deberá estar compartido únicamente con el(la) asistente y la profesora. Deberá contener los commit-push que se le indiquen durante el examen, no importa si en ese momento no compila. Si el examen no lo desarrolla asociado con este repositorio y con los commit-push que la profesora indique durante el examen, se considerará no entregado el examen aunque haya subido el examen en mediación virtual. El objetivo de esta forma de trabajo es validar la autenticidad de su examen.

Relacionadas con el desarrollo:

1. Según los lineamientos de la Universidad de Costa Rica en cuanto a evaluaciones virtuales sincrónicas, para tener derecho a realizar el

examen el estudiante deberá permanecer todo el tiempo en la sesión de Zoom y deberá tener encendida la cámara en donde se vea claramente en todo momento su rostro. De ser necesario se le solicitará que encienda el micrófono. De tener algún inconveniente con estos requerimientos debe conversarlo previamente con la docente.

2. Descargue el enunciado de mediación virtual y trabaje en BlueJ. Recuerde que debe trabajar conectado con el repositorio de Github indicado anteriormente.
3. El examen es individual. Es prohibido interactuar con cualquier otra persona que no sea la profesora.
4. Puede utilizar cualquier material y código propio escrito por usted (prácticas, tareas, libros, apuntes) y material de Internet debidamente referenciado. Si no cumple con estas características se considerará fraude.
5. En el comentario principal de cada clase, como autor indique su número de carné y su nombre.
6. Cada cierto tiempo la profesora indicará que deben hacer un commit-push en el repositorio, no importa si en ese momento no compila el examen, para lo cual debe estar atento con el audio encendido en todo momento. Es obligatorio hacer estos commit-push, de lo contrario la entrega del examen se considerará inválida y tendrá un cero como nota.

Relacionadas con un imprevisto:

1. Cualquier imprevisto técnico durante el examen deberá comunicarlo inmediatamente a la profesora por medio de un mensaje privado en Telegram.
2. Deberá estar atento(a) al chat grupal y al personal de Telegram, en caso de ser necesario que la profesora dé indicaciones.
3. Si durante el período de entrega del examen, entre las 9:45am y las 10:00am, mediación virtual estuviera “caído”, envíe el archivo .jar al correo institucional de la profesora con copia al asistente.
4. Si posee una computadora portátil, es una buena alternativa que la utilice para realizar el examen, asegurándose de que la batería está cargada al 100%, para que en caso de una falla eléctrica pueda continuar el examen. Si realiza el examen en una computadora de escritorio que depende de energía eléctrica y sucede un fallo en el suministro, debe continuar su examen en papel. En este caso se calificará la versión subida hasta ese

momento a Github, así como la documentación que aporte en papel. Recuerde reportarlo a la profesora tan pronto suceda el imprevisto.

5. Si normalmente utiliza el internet de la casa con señal wifi y se va la electricidad, pero posee internet en el celular infórmelo inmediatamente a la profesora por un mensaje privado de Telegram. En estos casos, de ser posible conecte la computadora a la señal del celular para realizar al menos los commit-push y para subir el examen en mediación.

Enunciado

Descargue el proyecto jar que viene adjunto con el examen, ábralo en BlueJ y guárdelo con el siguiente nombre: *NombreApellido_examen2*. En este proyecto se implementa parcialmente un programa que maneja dos listas enlazadas de enteros:

Una lista simplemente enlazada **recursiva** (*ListaRecursiva*): en esta lista el atributo *siguiente* de la clase *NodoRecursivo* es otra lista. Esta lista ya tiene implementados los métodos de insertar al final y *toString*.

Una lista **doblemente** enlazada (*ListaDoble*): en esta lista la clase *NodoDoble* tiene un atributo *siguiente* y otro *anterior*, ambos de tipo *NodoDoble*. La clase *ListaDoble* tiene un atributo *primero* y otro *ultimo*. Esta clase ya tiene implementado el método *toStringDerechoReves*, que devuelve una hilera conteniendo los elementos de la lista tanto al derecho como al revés. Este método le permitirá probar si está insertando bien los elementos (pregunta #3 del examen).

Usando este proyecto programe los métodos que se explican a continuación y que permiten que funcionen correctamente las opciones #4, #5 y #6 del programa. Todas las demás opciones del programa ya funcionan correctamente, no debe programarlas ni modificarlas.

1. (33%) En la clase *ListaRecursiva* programe el método:
public int buscarCuantos(int elBuscado)

que recibe de parámetro un entero y devuelve cuántas veces está en la lista, ya que la lista puede tener números repetidos. El número buscado podría no estar en la lista.

2. (33%) En la clase *ListaRecursiva* programe el método:
public boolean tenerSoloPositivos()

el cual devuelve *true* si la lista está vacía o si tiene solo números positivos, en caso contrario devuelve *false*.

3. (34%) En la clase *ListaDoble* programe el método:
public void insertarPenultimo(int elEntero)

el cual inserta el número recibido en la penúltima posición de la lista. Considere los casos especiales: si la lista está vacía simplemente lo inserta de primero; si la lista tiene solo un elemento, al insertarlo de penúltimo se convertirá en el primero de la lista. No olvide actualizar todos los punteros (*primero*, *ultimo*, *siguiente*, *anterior*).

Lineamientos generales de evaluación

- a) Cumplimiento de buenas prácticas de programación (convenciones).
- b) Correcto diseño y programación según el paradigma orientado a objetos.
- c) Uso correcto de las estructuras de datos.
- d) Uso correcto de estructuras de control.
- e) Correctitud de algoritmos y funcionamiento del programa.