*PROGRAMACIÓN II: EDA Lineales de Colas*.

Tarea Virtual #3

Tema: Colecciones de datos del tipo cola.

Objetivo: Resolución de ejercicios mediante la aplicación de colecciones de datos de tipo la estructura de cola.

Actividad: Implemente un proyecto de consola en Java para simular un servidor de gestión de impresión de una impresora conectada a la red. La impresora puede recibir varias peticiones desde diferentes ordenadores. Las peticiones serán impresas por orden de llegada. Cada petición incluye la siguiente información:

* id (String) de la máquina que solicita la impresión (por ejemplo, “I3493”)
* el nombre del documento a imprimir (por ejemplo, file1.pdf).
* número de páginas del documento.
* número de copias.

Implemente el siguiente diseño de clases:

1. La clase componente ***RequestImpresion*** con los atributos anteriormente descritos y los métodos de comportamiento necesarios como: *constructor parametrizable*, métodos *getter* y *setter* y método *toString*.
2. La clase compuesta ***ServerRequest*** que tiene por atributo una cola de objetos de la clase componente ***RequestImpresion*** y además debe implementar los siguientes métodos de comportamiento:
   1. ***addRequest***: toma una petición como entrada y la añade a la cola de peticiones. Dicho elemento insertado se situará al final de la cola en orden correspondiente a los objetos de la clase ***RequestImpresion.***
   2. ***printWork***: coge la primera petición y muestra sus datos (id, nombre del fichero, cantidad de páginas y número de copias a imprimir) por consola (únicamente simula la impresión de la petición). La petición, al ser mostrada, debe ser eliminada del conjunto de peticiones de la cola.
   3. ***getNumRequest()***: devuelve el número total de peticiones recibidas hasta el momento para imprimir.
   4. ***showAll()***: muestra todas las peticiones y sus datos (id, nombre del fichero, cantidad de páginas y número de copias a imprimir) que no han sido impresas aún.
   5. ***printAll()***: imprime (mostrar los datos) de todas las peticiones. Después de mostrar las peticiones, estas deben ser eliminadas.
3. Implemente en la clase de consola el ***método main*** que incluya un menú de opciones para las llamadas necesarias de todos los métodos, descritos anteriormente, desde un objeto de la clase compuesta ***ServerRequest***.

Enlaces (Materiales de Consulta):

* Lectura de las ***ideas claves***.
* ***Charla Magistral #3***.
* Lectura del ***capítulo 6 del libro Mark Allen Weiss***: Estructuras de datos en Java, 4ta edición. ***Págs. 225-236, págs. 244-257***. (Sección LO MAS RECOMENDADO).
* Lectura del ***capítulo 16 del libro Mark Allen Weiss***: Estructuras de datos en Java, 4ta edición. ***Págs. págs. 583-604***. (Sección LO MAS RECOMENDADO).
* Lectura del ***capítulo 17 del libro Mark Allen Weiss***: Estructuras de datos en Java, 4ta edición. ***Págs. 607-634***. (Sección LO MAS RECOMENDADO).

Orientaciones metodológicas:

* Se valorará la estética, claridad y la correlación con los materiales a consultar.
* Crear un proyecto de consola desde el ***IDE Netbeans 8.2*** (descargar [aquí](https://drive.google.com/file/d/1kHC83WGovky04ipCBO5LFeTd4wbkztR6/view?usp=sharing)) o compilador online de JAVA <https://www.onlinegdb.com/online_java_compiler>
* Implementar las opciones anteriormente descritas teniendo en cuenta el esquema general de ejecución de algoritmos y el paradigma de programación orientado a objetos.
* Comprobar la solución implementada anteriormente.
* Compactar la carpeta de proyecto con todo su directorio de subcarpetas y archivos del proyecto y, posteriormente, subir el compactado a la plataforma de Eva online.