

ESPECIFICACIONES TÉCNICAS PARA LA CONSTRUCCIÓN DEL PROYECTO FINAL DEL CURSO

A continuación se presenta cada uno de los elementos que debe tener el proyecto final del curso, recuerde que el tema es libre y usted deberá ajustar su proyecto a los requisitos aquí presentados. Cada ítem tiene una asignación de puntos, los cuales son un total de 100 puntos para una nota de 5.0.

General:

- Hecho
- No Hecho

Código:

- Hecho
- No Hecho

1. (5 Puntos) El proyecto debe estar montado y trabajado en GitHub, para aprender a usar esta herramienta se le propone realizar el siguiente curso: <https://www.codecademy.com/learn/learn-git>.
2. Toda la información del programa debe ser persistente.
 - a. (5 Puntos) Una parte de la persistencia debe ser manejada a través de archivos de texto plano (escritura y lectura de los archivos, tiene una asignación de 2.5 puntos cada uno).
 - b. (5 Puntos) Otra parte de la persistencia debe ser manejada a través de serialización de clases (escritura y lectura de los archivos, tiene una asignación de 2.5 puntos cada uno).
3. Se deben identificar los principales casos de excepción y manejarlos pertinentemente en cada caso (pueden ser lanzadas en un método, propagados por otros y capturados posteriormente)
 - a. (4 Puntos) Unas excepciones se manejarán con las excepciones propias del api de java (al menos 4).
 - b. (6 Puntos) Se deben identificar al menos 3 excepciones personalizadas (con su propia clase).
4. El proyecto debe incluir el diseño y la implementación de Pruebas Unitarias Automáticas para TODOS los métodos de las clases del mundo.
 - a. (5 Puntos) Diseño de los Casos de Prueba.
 - b. (7 Puntos) Implementación de las Pruebas Unitarias Automáticas.
5. (6 Puntos) Debe incluir al menos una implementación propia de cada algoritmo de ordenamiento de objetos visto en el curso, en los cuales, implemente al menos dos criterios diferentes por cada método de ordenamiento. (Owner, Client)
6. (6 Puntos) Debe incluir al menos dos implementación propia de un algoritmo de búsqueda binaria sobre una colección de objetos del mundo, usando un criterio diferentes para cada implementación. (Client->Vehicles, Owner->Parkings)
7. (8 Puntos) Debe incluir al menos dos implementaciones de listas enlazadas. (Bill, Slot)

8. (4 Puntos) Su modelamiento e implementación de las clases deben incluir (4) herencias diferentes (que no sean del mismo padre) (User, Record, Vehicle, ControlGlobal)
9. (4 Puntos) Su modelamiento e implementación de las clases deben incluir (3) Desacoplamiento a través del uso de interfaces que sean coherentes en el problema. (Generator, FileLoader, Encryptor)
10. (10 Puntos) Debe incluir al menos dos implementación propia de árboles binarios.(User, Report)
11. (10 Puntos) En su implementación debe incluir al menos cuatro métodos recursivos en estructuras naturalmente recursivas.(User, Report)
12. (6 Puntos) Su programa debe incluir al menos un panel donde usted dibuje un gráficos con al menos 2 primitivas por cada gráfico
13. (9 Puntos) Su proyecto debe utilizar coherentemente al menos 3 hilos adicionales al hilo principal.
14. Requerimientos funcionales y no funcionales
15. Diagrama de clases, diagrama de objetos y casos de pruebas
16. JavaDoc
17. Trazabilidad

Cada uno de los puntos antes mencionados debe tener su respectivo análisis (requerimientos), diseño (diagrama de clases, diagrama de objetos y casos de pruebas), implementación (código Java y documentación del código - JavaDoc) y pruebas para obtener todos los puntos asignados. Cualquier duda al respecto con los temas, aplicación e inclusión de los temas aquí presentados, por favor, aclarar los con el docente de su asignatura. Recuerde seguir la buenas prácticas recomendadas en el curso.

Grupos de trabajo

- Dependerá de la propuesta realizada, según su dimensión y complejidad.
- Las definiciones realizadas anteriormente pueden ser para un grupo de una o dos personas, para un grupo mayor a dos personas los requerimientos deben ser aprobados por el profesor del curso para ser aceptados (si los requerimientos no son aprobados por el profesor, el proyecto tendrá como nota 0.0).

Entrega y sustentación

- La fecha de entrega del taller será la establecida en el campus virtual.
- Para la entrega debe generar un archivo .zip o .tar con el contenido del taller (código fuente, documentación del proyecto, ayudas y ejecutables) el cual lo debe nombrar con los códigos de los estudiantes de cada grupo, separado por el carácter “-”.
- Deben incluir un archivo llamado readme.html, que contenga los aspectos desarrollados, el contenido del archivo .zip y cualquier información adicional sobre el proyecto (ejemplo instalación).
- La sustentación será realizada por cada integrante del grupo de trabajo.

Calificación

- El desarrollo completo de todas las funcionalidades será equivalente a una nota de 5.0 en el proyecto y un total de 100 puntos.
- En todos los casos la sustentación será pilar fundamental de la nota asignada. Cada persona, después de la sustentación tendrá asignado un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de su sustentación. Su nota definitiva será la nota del proyecto, multiplicada por ese valor. Si su asignación es 1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0. La idea es que lo que no sea debidamente sustentado no vale así funcione muy bien!!!