

Proyecto #3 - Jar Analyzer

Instituto Tecnológico de Costa Rica
Área de Ingeniería en Computadores
Algoritmos y Estructuras de Datos 1
I Semestre 2018
Valor 15%



Objetivo general

→ Implementar una aplicación de escritorio que permita analizar un Jar a nivel de dependencias.

Objetivos específicos

- Aplicar conocimientos de grafos
- Aplicar OO en Java

Descripción del problema

El proyecto consiste en la implementación de una aplicación de escritorio que recibe como entrada un archivo Jar. La aplicación toma dicho Jar, obtiene la lista de Jars definidos en el classpath y visualiza gráficamente un grafo en la que se ve el Jar, sus dependencias y las dependencias de las dependencias...

Cuando una dependencia no puede encontrarse localmente, la aplicación utilizará Maven y alguno de los repositorios comunes para descargarla. Debe visualizarse en la aplicación, cuáles dependencias no se encontraron y el progreso de la descarga desde el repositorio en Maven.

A este nivel, se pueden calcular las siguientes métricas de cada jar:

- Grado saliente
- Grado entrante
- Determinar si el grafo es conexo
- Generar un ranking que indique cuales jars son los que tienen más dependencias
- Generar un ranking que indique cuales jars son más referenciados por otros.

El usuario puede hacer zoom in a cada uno de los jars. De este forma se muestra un nuevo grafo pero a nivel de clases. Se mostrará cada una de las clases dentro del jar y las relaciones con otras clases dentro del mismo jar. De igual forma se calculará

- Grado saliente
- Grado entrante
- Determinar si el grafo es conexo
- Generar un ranking que indique cuáles clases son los que tienen más dependencias
- Generar un ranking que indique cuáles clases son más referenciados por otros.

Documentación requerida

1. Se deberá documentar el código fuente utilizando JavaDoc.

2. Se deberá entregar un documento que contenga:
 - a. Todas las partes estándar: Portada, índice, introducción, conclusión, bibliografía y anexos.
 - b. El cuerpo del documento debe contener:
 - Breve descripción del problema
 - **Planificación y administración del proyecto**
 - ◆ Lista de features e historias de usuario identificados de la especificación.
 - ◆ Distribución de historias de usuario por criticalidad y secuencia de uso.
 - ◆ Minimal system span.
 - ◆ Asignación de historias de usuario entre los miembros del equipo.
 - ◆ Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental y especificar el responsable de cada una.
 - ◆ Bitácora.
 - **Diseño**
 - ◆ Diagrama de componentes.
 - ◆ Diagrama de arquitectura.
 - ◆ Diagrama de secuencia (Seleccionar al menos 4 Historias de Usuario).
 - ◆ Diagrama de clases inicial.
 - ◆ Diagrama de clases final (generado utilizando el IDE).
 - **Implementación**
 - ◆ Descripción de las estructuras de datos desarrolladas.
 - ◆ Descripción detallada de los algoritmos desarrollados.
 - ◆ Problemas encontrados: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo. Incluye descripción detallada, intentos de solución sin éxito, solución encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.

Aspectos operativos y evaluación:

1. **Fecha de entrega: De acuerdo al cronograma del curso.**
2. El proyecto tiene un valor de 15% de la nota del curso.
3. El trabajo es **en grupos de 4 personas.**
4. Es obligatorio utilizar un manejador de versiones del código. Puede ser git o svn. Se revisará que cada commit lleve comentarios relevantes relacionados con alguna tarea identificada en la sección de planificación.
5. **Los proyectos que no estén debidamente integrados, no serán revisados.**
6. El código tendrá un valor total de 65%, la documentación (externa e interna) 30% y la defensa 5%.
7. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
8. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado, se recomienda que realicen la documentación conforme se implementa el código.
9. La nota del código NO podrá exceder en 35 puntos la nota de la documentación, por lo cual se recomienda documentar conforme se programa.

10. Se debe enviar el código (preliminar) y la documentación a más tardar a las 23:59 del día de la fecha de entrega al email del profesor. **Se debe seguir el formato del Subject descrito en el Programa del Curso.**
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial, momento en el cual deben enviar un email con el código fuente, si no lo hacen se asumirá que la versión oficial del código fue la enviada el día de la fecha de entrega junto con la documentación. **Se debe enviar en el mismo correo del punto 8.**
13. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones:
 - a. Si no se entrega documentación externa, automáticamente se obtiene una nota de 0.
 - b. Si no se utiliza un manejador de código se obtiene una nota de 0.
 - c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - d. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. El código debe ser desarrollado en Java, en caso contrario se obtendrá una nota de 0.
 - f. Si no se siguen las reglas del formato de email se obtendrá una nota de 0.
 - g. La nota de la documentación debe ser acorde a la completitud del proyecto.
 - h. Los estudiantes que no realicen la defensa del proyecto obtienen una nota de 0.
14. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto. El único requerimiento que se consultará durante la defensa del proyecto es el diagrama de clases, documentación interna y la documentación en el manejador de código.
15. Cada estudiante tendrá como máximo 15 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
16. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
17. Cada estudiante es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
18. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.