Tecnología de la programación

Práctica del examen final de Junio.



Introducción de Threads en el algoritmo Ant Colony Optimization: En este documento se explica en que consiste la práctica que se ha realizado, su alcance, sus objetivos, las soluciones dadas, etcétera…

2013

Juan Luis Pérez Valbuena, Daniel Serrano Torres, Juan Carlos Marcos González, Emilio Álvarez Piñeiro y Álvaro Quesada Pimentel

14/06/2013

Índice

[**1.** **Requisitos** 2](#_Toc346703472)

[**2.** **Objetivos** 2](#_Toc346703473)

[**3.** **Resumen** 2](#_Toc346703474)

[**4.** **Abstract** 3](#_Toc346703475)

[**6.** **Sugerencias** 3](#_Toc346703476)

[**7.** **Introducción** 3](#_Toc346703477)

[**8.** **Objetivos que se alcanzarán** 4](#_Toc346703478)

[**9.** **Relación con la docencia cursada** 5](#_Toc346703479)

[**10.** **Viabilidad** 5](#_Toc346703480)

[**11.** **Estado del arte y fundamentación teórica** 5](#_Toc346703481)

[**12.** **Propuestas de solución** 6](#_Toc346703482)

[**13.** **Conclusión** 7](#_Toc346703483)

[**15.** **Bibliografía** 7](#_Toc346703484)

**PRÁCTICA DOS**

Buenas Prácticas de Programación

**Fecha de entrega:** 24/1/2013

1. **Requisitos**

* Para realizar esta práctica es necesario conocer los factores de calidad del software.
* Es necesario conocer el lenguaje Java así como tener conocimientos de la herramienta eclipse.
* No es estrictamente necesario pero si ayuda conocer lo que hace el código para así poder corregirlo con seguridad.
* Es fundamental tener conocimientos básicos sobre Threads: cómo funcionan, cómo se declaran y cómo se ejecutan. También es recomendable conocer métodos de sincronización en Java.

1. **Objetivos**

**General**:

El objetivo de la práctica es introducir el uso de Threads o hilos de ejecución en la práctica sobre el SimpleACS(Ant Colony Optimization). Para ello se trabajará sobre la versión de la práctica entregada en Febrero, en la que se intentó aplicar los factores de calidad del software para facilitar la comprensión del código. Sobre ese código ya simplificado se usarán hilos de ejecución sobre el algoritmo de optimización para agilizar, aún más, la obtención de la solución.

**Particulares**:

* Conseguir que compile y que funcione mediante el uso de hilos.

1. **Resumen**

Empezando con el código de la entrega de Febrero se ha intentado encapsular el comportamiento propio de las “hormigas”, los agentes que recorren el circuito intentando resolver el problema, para poder separarlos de la “colonia” y que sólo se comuniquen con ésta en el momento de su creación y cuando ha finalizado su ejecución. Así mediante el uso de hilos conseguimos que las hormigas se ejecuten independientemente de la colonia y que se eliminen los tiempos de espera entre ambas.

1. **Abstract**

Starting with the code presented in February we have tried to encapsulate “ants´” behavior, the ones marching through the track trying to solve the problem, and separate it from the “colony” so they only interwine with the latter in their creation and when they´re finished their execution. Using Threads we achieve independence between the “colony” and it’s “ants” therefore avoiding periods of delay.

1. **Observaciones**

La práctica requiere mucho tiempo para poder aislar el comportamiento de unos y otros procesos. Es una tarea complicada dada la complejidad intrínseca del problema en cuestión, de marcado carácter matemático y de amplia complejidad, y dado que el código con el que se trabaja es resultado de un proceso, con mayor o menor grado de éxito, de depuración y adecuación del código original entregado como material de la primera práctica. Por eso, si el trabajo desempeñado en la aplicación de los factores de calidad del software en la práctica de Febrero fue insuficiente, para la introducción de los hilos los estudiantes se verán obligados a realizar ese trabajo pendiente debido a que es necesario comprender en un grado medio/alto cual es el flujo del programa y donde se definen los límites del comportamiento de cada uno de los actores en la aplicación.

1. **Sugerencias**

Sería recomendable incluir temario sobre sincronización en Java para el desarrollo de prácticas con concurrencias. A parte de esta sencilla petición no se encuentra ningún problema para realizar la práctica

1. **Introducción**

Dado un código depurado en prácticas anteriores por los alumnos se pide que se introduzca el uso de threads o hilos en la aplicación. Los hilos permiten que la aplicación principal sea capaz de liberar procesos que no tiene que controlar, de los que sólo le importa los resultados que obtenga, reduciendo los tiempos de espera de la aplicación. Mientras que en una implementación sin hilos sería la aplicación la que tendría que ejecutar todos los procesos uno a uno, quedando bloqueada mientras estos acaban, en la implementación con hilos la aplicación solo tiene que encargarse de declarar los procesos a ejecutarse y, en este caso, de obtener la respuesta de éstos cuando finalicen su ejecución, sin detener en ningún momento el flujo del programa. Así, además de reducir los tiempos “muertos” en la ejecución de la aplicación, conseguimos aprovechar al máximo la capacidad de procesamiento del hardware, pudiendo realizar una mayor carga de trabajo en un tiempo menor.

Pero el principal problema de hilos es la concurrencia, cuando los procesos son asíncronos y tienen que realizar operaciones con **datos** **comunes** puede ocurrir que se produzcan fallos en la manipulación de estos datos, lo que puede dar lugar a un resultado final erróneo o incluso errores que paren la ejecución. Por eso también se usarán funciones de **sincronización**. La sincronización permite evitar este acceso simultáneo por parte de los procesos a partes de la aplicación que son sensibles de producir errores si no se manipulan correctamente.

1. **Objetivos que se alcanzarán**

* Ahondar aún más en la aplicación de los factores de calidad del software para alcanzar la comprensión necesaria del funcionamiento de la aplicación que permita el encapsulamiento de los diferentes actores y su diferenciación en hilos de ejecución.
* Aprender sobre el uso general de hilos:
  + La clase Thread de Java.
  + El interfaz Runnable de Java.
  + Inicio y finalización de procesos.
  + Cómo evitar la concurrencia mediante la sincronización.

1. **Relación con la docencia cursada**

* Esta práctica pone a prueba nuestra capacidad para aplicar los factores de calidad del software.
* Sirve para poner en práctica los conocimientos teóricos obtenidos en clase sobre hilos de ejecución. Cómo utilizarlos y entender cuando y donde son necesarios.

1. **Viabilidad**

Dado que los conocimientos de los participantes en la práctica sobre el funcionamiento exacto del algoritmo Ant Colony Optimization, y por tanto, de los actores que tienen lugar en la aplicación, son nulos y dado que el código original obtenido para la realización de la primera entrega era una implementación basada en el paradigma de programación estructurada, lo que dificulta aún más su análisis, se espera que la versión obtenida al final de la práctica sea una implementación basada en la programación orientada a objetos, con uso amplio y adecuado de los factores de calidad del software y que incorpore el uso de hilos para la ejecución de las hormigas, independientemente de que el algoritmo matemático en sí no sea el correcto o que el resultado del algoritmo tampoco lo sea.

1. **Estado del arte y fundamentación teórica**

Esta aplicación fue publicada por la universidad Roskildensis, universidad que se encuentra en Dinamarca. En la aplicación se da solución al problema del viajero, que consiste en recorrer una serie de puntos pasando por todos ellos en el menor tiempo posible. Para ello se fijan en las hormigas.

Las hormigas particularmente eficientes en un cosa, encontrar la ruta más corta entre dos puntos. Esto lo consiguen de la siguiente manera: Las hormigas van moviéndose al azar hasta que encuentran un punto de interés, supongamos comida, cuando vuelven al hormiguero van dejando una sustancia química a la que se le conoce como feromona, lo que les ayudara a encontrar el camino de vuelta hacia la comida. Cuando otras hormigas encuentran el camino de feromonas, estas lo seguirán, cada vez de una manera menos errática. Debido a que la feromona es una sustancia muy volátil, se necesita un flujo constante de hormigas para mantener el camino. Esto significa que si un camino más corto existe, el poder de las feromonas será más fuerte, ya que las hormigas lo atravesaran en un periodo más corto de tiempo. Después de un periodo corto de tiempo la mayoría de las hormigas seguirán el camino más corto, ya que las feromonas serán más fuertes en el.

Alberto Colorni, Maniezzo Vittorio y Marco Dorigo fueron los primeros en llegar con un algoritmo que simula el comportamiento de las hormigas, llamándolo Ant Colony System (Sistema de la colonia de hormigas) ACS, conocido comúnmente ACO, la O es por las optimizaciones que se hacen diariamente a la ACS original. Como ACO contiene métodos generales para la resolución de distintos problemas, puede ser visto como una colección de métodos heurísticos que se pueden aplicar a un gran número de problemas. ACO es parte de la ciencia “Inteligencia de enjambre”, que se basa en la utilización de pequeñas unidades independientes (por ejemplo: abejas, hormigas, aves) para resolver problemas. Solas estas unidades son tontas, pero cuando se les permite comunicarse entre sí y reaccionar a sus respuestas, son capaces de resolver problemas complejos.

1. **Propuestas de solución**

* Soluciones
  1. Primera solución:

Aislar el comportamiento de las hormigas en una clase con el mismo nombre. En la clase estaría recogido lo que tiene que hacer una hormiga por sí sola, así que desde que es creada, la hormiga puede realizar su trabajo independientemente de la aplicación y solo comunicarse con ésta cuando haya terminado su labor. La clase implementaría el interfaz Runnable que le permitiría ejecutarse en hilos.

* 1. Debido a los problemas provocados por la concurrencia en la lectura/escritura de los datos comunes de las hormigas, se ha decidido utilizar el prefijo **synchronized**(palabra reservada en Java) en la declaración de las funciones de finalización de las hormigas. Las funciones declaradas con **synchronized,** si son invocadas simultáneamente por dos procesos, dan prioridad al proceso que la invoca primero mientras deja al segundo en espera hasta que termine.

1. **Conclusión**

Al realizar esta práctica hemos comprendido mejor la importancia de los factores de calidad del software, y él porque es necesario llevarlos a cabo correctamente.

También hemos puesto en práctica nuestros conocimientos sobre hilos y hemos aprendido a identificar procesos liberables y cómo evitar la concurrencia mediante la sincronización

1. **Líneas Futuras**

De cara al futuro se tendrá muy en cuenta los factores de calidad del software, y la importancia de ellos, así como la gran importancia que tiene legibilidad del código.

1. **Bibliografía**

**Estructura del documento actual:**

* Transparencias sobre la estructura de las prácticas, del profesor Francisco Javier Crespo.

**Programación orientada a objetos:**

* <http://www.elguille.info/colabora/NET2005/Percynet_ConstruyendoSoftCalidad.htm>
* PDF sobre la ACS de la universidad de Roskildensis.
* Uso de métodos sincronizados y ejemplos:

<http://docs.oracle.com/javase/tutorial/essential/concurrency/syncmeth.html>