

Resúmenes 14, 15 y 16

Data Structures and Algorithms in Java

Robert Lafore

Manuel Arturo Chinchilla Sánchez

2013009344

Una estructura de datos es un arreglo de datos en la memoria de una computadora (o algunas veces en un disco). Las estructuras de datos incluyen matrices, listas enlazadas, pilas, árboles binarios y tablas hash, entre otros. Algoritmos manipular los datos de estas estructuras de varias maneras, tales como la búsqueda de un elemento de datos particular y la clasificación de los datos.

Muchas de las estructuras y técnicas se refieren a cómo manejar el almacenamiento de datos del mundo real. Por datos del mundo real, nos referimos a los datos que describen las entidades físicas externos al ordenador. Como algunos ejemplos, un registro de personal describe un ser humano real, un registro de inventario describe un elemento pieza del coche o de la tienda de comestibles existente, y un registro de las transacciones financieras se describe, por ejemplo, un control real por escrito a pagar el recibo de la luz. Un ejemplo no informático de almacenamiento de datos del mundo real es una pila de 3-por-5 tarjetas. Estas tarjetas se pueden utilizar para una variedad de propósitos. Si cada tarjeta tiene el nombre, dirección y número de teléfono de una persona, el resultado es una libreta de direcciones. Si cada tarjeta contiene el nombre, la ubicación y valor de la posesión de la casa, el resultado es un inventario de su vivienda. Por supuesto, las fichas no son exactamente el estado de la técnica. Casi cualquier cosa que una vez que se hizo con tarjetas de índice ahora se puede hacer con un ordenador. Suponga que desea actualizar su sistema de índice de tarjetas de edad a un programa de ordenador.

Conceptualizando de un problema del mundo real utilizando lenguajes de procedimiento es difícil. Métodos llevan a cabo una tarea, mientras que los datos almacenan la información, pero la mayoría de los objetos del mundo real hacen ambas cosas. El termostato de su horno, por ejemplo, lleva a cabo tareas (que dan vuelta el horno encendido y apagado), pero también almacena la información (la temperatura actual y la temperatura deseada) .Si usted escribió un programa de control del termostato en una lengua de procedimiento, que podría terminar con dos métodos, horno encendido () y el horno apagado (), pero también dos variables globales, temperatura actual (suministrada por un termómetro) y la temperatura deseada (establecido por el usuario) .Sin embargo, estos métodos y variables no

formarían ningún tipo de unidad de programación; no habría ninguna unidad en el programa que se podría llamar termostato. El único tal concepto sería en la mente del programador. Para programas grandes, que podrían contener cientos de entidades como termostatos, este enfoque procesal hizo las cosas caóticas, propensas a errores, ya veces imposible de implementar en absoluto. Lo que se necesitaba era una mejor correspondencia entre las cosas en el programa y las cosas en el mundo exterior.

La herencia es la creación de una clase, llamada la clase extendedor derivada, de otra clase llamada la clase base. La clase extendida tiene todas las características de la clase base, además de algunas características adicionales. Por ejemplo, una clase secretario podría ser derivado de una clase más general empleado e incluyen un campo llamado velocidad de escritura que carecía de la clase empleado. En Java, la herencia se llama también subclases. La clase base puede ser llamado el súper-clase, y la clase extendida puede ser llamada la subclase. La herencia le permite agregar fácilmente las características de una clase existente y es una ayuda importante en el diseño de programas con muchas clases relacionadas. Por lo tanto la herencia hace que sea fácil de reutilizar clases para un propósito ligeramente diferente, un beneficio clave de la programación orientada a objetos. El polimorfismo implica el tratamiento de objetos de diferentes clases de la misma manera. Para polimorfismo funcione, estas diferentes clases deben derivarse de la misma clase base. En la práctica, el polimorfismo implica generalmente una llamada de método que realmente realiza diferentes métodos para objetos de diferentes clases.

En los últimos años, se ha puesto de moda para comenzar un libro sobre las estructuras de datos y algoritmos con un capítulo sobre la ingeniería de software. Nosotros no seguimos ese enfoque, pero vamos a examinar brevemente la ingeniería de software y ver cómo encaja en los temas que discutimos en este libro. Ingeniería de software es el estudio de formas de crear programas informáticos grandes y complejos, que implican muchos programadores. Se centra en el diseño general de los programas y en la creación de ese diseño de las necesidades de los usuarios finales. Ingeniería de software tiene que ver con el ciclo de vida de un

proyecto de software, que incluye la especificación, diseño, verificación, codificación, pruebas, producción y mantenimiento.