

Estructuras de Datos (COM-11102)

agosto - diciembre de 2019

Objetivos:

- Emplear los conceptos básicos de la programación orientada a objetos.
- Implementar correctamente estructuras de datos lineales (con arreglos y estructuras enlazadas) utilizando interfaces y clases.
- Utilizar objetos de estas estructuras de datos para resolver problemas.
- Aplicar algoritmos recursivos para resolver problemas.

Temario:

Programación Orientada a Objetos

Software de calidad. Interfaces. Herencia, jerarquías de clases y polimorfismo. Métodos heredados de Object: toString, equals y hashCode. Sobreescritura de métodos. Sobrecarga de métodos. Visibilidad de miembros en Java: private, public, protected y la cuarta categoría. Diagramas UML. Miembros estáticos y no estáticos. Manejo de excepciones. Interfaces gráficas de usuario creadas (fácilmente) con un IDE. Clases envoltorio en Java: Integer, Double, Character y Boolean. Conversión/reinterpretación explícita de tipos ("casting"). Documentación (Javadoc). Pruebas unitarias (JUnit). Interfaz Serializable y archivos de objetos.

Pilas

Tipos de datos abstractos. Concepto de pila: característica LIFO. Implementación de pilas con arreglos. Aplicaciones: verificación de balanceo de paréntesis, conversión de expresiones infijas a postfijas y evaluación de expresiones postfijas.

Conjuntos

Concepto. Implementación usando arreglos. Operaciones. Ejemplos y aplicaciones. Interfaz Iterable y ciclo for-each. Interfaz Iterator (y sus métodos hasNext y next).

Recursividad

Definición. Escritura y seguimiento de código recursivo. Ejemplos y aplicaciones. Memoización.

Colas

Concepto de cola: característica FIFO. Implementación de colas con arreglos. Utilización de los arreglos "de forma circular". Ejemplos y aplicaciones.

Estructuras enlazadas

Concepto. Nodos enlazables. Estructuras enlazadas lineales. Operaciones: recorrer, obtener, agregar, quitar, buscar. Implementación de pilas, colas y conjuntos con estructuras enlazadas. Comparaciones de las dos implementaciones (arreglos vs. estructuras enlazadas). Estructuras doblemente enlazadas. Estructuras enlazadas circulares y ortogonales. Ejemplos y aplicaciones.

Listas

Concepto. Subtipos de listas: ordenadas, desordenadas e indexadas. Interfaces Comparable y Comparator y su uso en las listas ordenadas. Ejemplos y aplicaciones.

Evaluación:

Proyectos (15%), tareas y controles (10%), dos exámenes parciales (25% cada uno) y el examen final (25%). La entrega de los proyectos incluye el programa y un reporte escrito.

Para aprobar la materia es necesario tener un promedio de exámenes parciales aprobatorio y aprobar el examen final.

Bibliografía:

- John Lewis y Joseph Chase, *Java Software Structures – Designing and using Data Structures*, Fourth edition, Pearson, 2014.
- Silvia Guardati, *Estructuras de Datos Básicas –Programación Orientada a Objetos con Java*, Alfaomega, 2015.
- Eric Roberts, *Thinking Recursively with Java*, John Wiley & Sons, 2006.
- Osvaldo Cairó y Silvia Guardati, *Estructuras de datos*, Tercera edición, McGraw-Hill, 2006.
- Peter Drake, *Data Structures and Algorithms in Java*, Pearson, 2006.
- Mark Weiss, *Data Structures and Algorithm Analysis in Java*, Second Edition, Addison Wesley, 2007.
- Documentación oficial de Java proporcionada por Oracle:
<http://docs.oracle.com/javase/8/docs/> y <http://docs.oracle.com/javase/8/docs/api/> y
<https://docs.oracle.com/javase/tutorial/java/index.html>.

Para obtener el IDE NetBeans:

Software bundle: NetBeans (con JUnit) y JDK (JDK 8u141 with NetBeans 8.2).

<http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html> o

<https://netbeans.org/downloads/>.