

PERFIL DCC_C13

Asignatura: 3010426 Teoría de lenguajes de programación
Planes relacionados: Ciencias De La Computación, Ingeniería De Sistemas E Informática, Ingeniería De Control
Contenido: 1. INTRODUCCIÓN A LOS PARADIGMAS DE PROGRAMACIÓN 1.1. Programación Imperativa 1.2. Programación Orientada a Objetos 1.3. Programación Dinámica 1.4. Programación dirigida por Eventos 1.5. Programación declarativa 1.6. Programación Lógica 1.7. Programación Funcional 1.8. Programación con restricciones 1.9. Programación multiparadigma 1.10. Lenguajes específicos del dominio 2. PRINCIPIOS DE DISEÑO DE LOS LENGUAJES DE PROGRAMACIÓN 2.1. Definición de compilador 2.2. Estructura de un compilador 2.3. Tipos de Compiladores 2.4. Fases del proceso de compilación 2.5. Herramientas y descripción del lenguaje 2.6. Patronaje 3. ANÁLISIS LÉXICO 3.1. Especificación y construcción de un analizador léxico 3.2. Expresiones Regulares 3.3. Diagramas de Transiciones 4. ANÁLISIS SINTÁCTICO 4.1. Diseño de Gramáticas 4.2. Recursividad, ambigüedad, asociatividad, precedencia y parentización. 4.3. Análisis Lineal 4.4. Diagramas de Sintaxis 4.5. Análisis Descendente 4.6. Tabla de Tipos y Símbolos 5. ANÁLISIS SEMÁNTICO 5.1. Atributos y Acciones semánticas 5.2. Notaciones para la especificación de un traductor 5.3. Comprobación semántica 6. GENERACIÓN DE CÓDIGO INTERMEDIO Y FINAL

- 6.1. Tipos de código intermedio.
- 6.2. Código intermedio para registros
- 6.3. Espacio de direcciones
- 6.4. Secuencias de Acciones

Asignatura: 3007744 Programación orientada a objetos
Planes relacionados: Ciencias De La Computación, Ingeniería De Sistemas E Informática, Ingeniería Eléctrica, Ingeniería de Control, Estadística
Contenido: EL MODELO OBJETUAL (MODELO CONCEPTUAL). 1. La complejidad del software y las metodologías. 2. Principios: - Modularidad. - Jerarquía. - Abstracción. - Encapsulamiento. - Concurrencia, Persistencia, Reusabilidad, Control de Acceso. 3. El manifiesto. 4. Elementos: -Objetos -Clases. -Atributos. *Eventos y estados. -Agrupaciones de Clases. 5. Relaciones: -Relaciones entre Clases. * Composición. *Relación de uso *Herencia. -Relaciones entre objetos y clases. -Relaciones entre Objetos. *Intercambio de mensajes. *Composición. PROGRAMACIÓN ORIENTADA POR OBJETOS. 1. Modelos Lingüísticos OO - C++. - Smalltalk. HERRAMIENTAS OO. 1. Análisis diseño y Construcción OO. 2. Modelos de especificación para análisis y diseño. -Diagramas de Clases. -Diagrama de transición de estado. -Diagramas de relaciones entre clases. -Diagramas de relaciones entre estados. DESARROLLO ORIENTADO A OBJETOS (MODELO METODOLÓGICO). 1. Metodología de Desarrollo. -Desarrollo iterativo incremental y prototípico. -Estudio de Metodologías. * OMT. * BOOCH. * JACOBSON. * OOMETHOD.

Asignatura: 3007741 Estructura de datos
Planes relacionados: Ciencias De La Computación, Ingeniería De Sistemas E Informática, Ingeniería Eléctrica, Ingeniería de Control, Ingeniería Biológica, Ingeniería Mecánica
Contenido: Agregados básicos. 1. Concepto de agregado. Constructores y Selectores. 2. Agregación Estructurada. Registros (secuencia). Uniones (selección, Repeticiones (iteración)). Tipos abstractos: Interfaz e implementación. 1. Especificaciones del agregado. Tipo de los elementos, estructura de agregación. Especificación de las Operaciones sobre el agregado. La lista como tipo abstracto.

2. Implementación de agregado. Representación de los componentes y de sus enlaces. Implementación de la lista como arreglo ordenado y como lista doblemente enlazada. Efecto de la implementación en la eficiencia.

Algoritmos y eficiencia.

1. Factores que determinan la eficiencia. Principio de invarianza. Justificación de su utilización. Complejidad.
2. Orden, Omega y Theta de una función. Definición y primeras consecuencias.
3. Medida de un algoritmo. Selección del Barómetro. Número de repeticiones. Reducción a una escala comparativa.
4. Tipos de algoritmo. Voraces. Divide y Vencerás preconditionamiento. Recursivo. Probabilístico.

Estructuras iteradas.

1. Justificación frente a la eficiencia de las operaciones sobre el agregado.
2. Estructura del agregado determinada por la aridez de las relaciones.
3. Listas. Ordenados por clave y por frecuencia. Implementaciones. Orden de sus operaciones.
4. Árboles de búsqueda. Balanceos. Btree. Orden de sus operaciones.
5. Estructuras Hash.
6. Funciones Hash.
7. Colisión.
8. Eficiencia.

Aplicaciones.

1. Ejemplos de programación que requieren de estructuras.
2. Uso del agregado como tipo abstracto. 3. Selección y refinamiento del tipo de estructura e implantación).