

АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «Конструирование компиляторов»

по основной профессиональной образовательной программе по направлению подготовки
09.04.04 «Программная инженерия» (уровень магистратуры)

Направленность (профиль): Разработка программно-информационных систем

Общий объем дисциплины – 5 з.е. (180 часов)

Форма промежуточной аттестации – Экзамен.

В результате освоения дисциплины у обучающихся должны быть сформированы компетенции с соответствующими индикаторами их достижения:

- ПК-6.1: Анализирует и выбирает методы проектирования трансляторов и интерпретаторов языков программирования;
- ПК-6.2: Проектирует трансляторы и интерпретаторы языков программирования;

Содержание дисциплины:

Дисциплина «Конструирование компиляторов» включает в себя следующие разделы:

Форма обучения очная. Семестр 2.

1. Структура компилятора. Понятие компиляции и интерпретации. Методы проектирования трансляторов и интерпретаторов языков программирования. Лексический, синтаксический и семантический анализ.

КС-грамматики языков программирования.

Контекстные условия.

Реализация лексического уровня..

2. Синтаксический анализ. Нисходящий и восходящий синтаксический анализ.

Рекурсивные и магазинные методы синтаксического анализа.

Метод рекурсивного спуска.

LL(k)-анализаторы: определение, алгоритм построения управляющей таблицы, конфликты и их устранение.

Программирование LL(k)-анализатора.

S-анализаторы.

Анализаторы предшествования: определение, алгоритм построения управляющей таблицы, конфликты и их устранение.

Программирование анализатора предшествования.

LR(k)-анализаторы: определение, алгоритм построения управляющей таблицы.

Программирование LR(k)-анализатора..

3. Синтаксически управляемый перевод. Определение синтаксически управляемого перевода.

Операционные символы и семантические подпрограммы.

Понятие промежуточного кода. Способы представления дерева разбора:

древовидная структура, ПОЛИЗ, тетрады и триады.

Примеры представления в промежуточном коде конструкций языка программирования..

4. Оптимизация кода. Граф управления программой (Flow Diagram).

Выделение линейных участков, циклов, ветвлений.

Оптимизация выражений: вычисление, совпадающие фрагменты, Агрегаты аддитивных и мультипликативных операций.

Оптимизация ветвлений и циклов.

Оптимизация функций и их вызовов. Оптимизация хвостовой рекурсии.

Удаление недостижимого кода..

5. Генерация кода. Структура генерируемого кода, назначение и структура сегментов ассемблерного кода.

Назначение регистров.

Распределение памяти, выделение памяти для локальных и глобальных данных.

Пролог и эпилог функции.

Генерация уникальных имен.

Команды с вещественными и целыми данными.

Ассемблерный код для функций и их вызовов. Передача параметров через регистры и стек..

6. Автоматизация проектирования трансляторов. Метода автоматизации построения трансляторов.

Ранние генераторы Lex и Yacc.

Система ANTLR: язык описания грамматики, использование средств отладки, генерация кода, работа с семантическими функциями, методы listener и visitor..

Разработал:
профессор
кафедры ПМ

Е.Н. Крючкова

Проверил:
Декан ФИТ

А.С. Авдеев