

УТВЕРЖДАЮ

Зав. кафедрой программной инженерии БГТУ

к.т.н., доцент _____ В.В. Смелов

протокол № 4 от 28 ноября 2024 г.

**Экзаменационные вопросы по дисциплине
«Конструирование программного обеспечения»,
2 курс ПОИТ, весенний семестр 2024
6-05-0612-01 «Программная инженерия»**

1. Лексический анализ: определение, назначение, задачи лексического анализа, применение.
2. Лексический анализатор: определение, место лексического анализатора в структуре транслятора, входная и выходная информация, способы взаимодействия с синтаксическим анализатором.
3. Лексический анализатор: функции, выполняемые лексическим анализатором. Принцип реализации лексического анализатора в вашем курсовом проекте.
4. Определение грамматики для описания лексики языка программирования. Регулярное выражение как способ задания лексики языка программирования. Пример использования.
5. Конечный автомат: формальное определение конечного автомата, схема работы лексического анализатора.
6. Конечный автомат: алгоритм работы простейшего лексического анализатора.
7. Схема работы лексического анализатора. Определение конечного автомата.
8. Конечный автомат: определение графа переходов конечного автомата и метод его построение по регулярному выражению.
9. Конечный автомат: алгоритм разбора цепочки, основанный на двух массивах.
10. Синтаксический анализ: определение, назначение, задачи синтаксического анализа применение.
11. Синтаксический анализатор: определение, место синтаксического анализатора в структуре транслятора, входная и выходная информация.
12. Синтаксический анализатор: задачи синтаксического анализатора, взаимодействие с лексическим анализатором, основные типы синтаксических анализаторов КС-грамматик.
13. Синтаксический анализатор: какой тип грамматик применяется при описании синтаксиса языков программирования. Формальное определение КС-грамматики в иерархии Хомского.
14. Приведение контекстно-свободной грамматики: основные цели преобразований КС-грамматик, порядок выполнения шагов преобразования КС-грамматики. Определение бесплодного символа, алгоритм удаления бесплодных символов.
15. Приведение контекстно-свободной грамматики: основные цели преобразований КС-грамматик, порядок выполнения шагов преобразования КС-грамматики. Определение недостижимого символа, алгоритм удаления недостижимых символов.
16. Приведение контекстно-свободной грамматики: основные цели преобразований КС-грамматик, порядок выполнения шагов преобразования КС-грамматики. Определение λ -правил, цепных правил. Алгоритмы удаления λ -правил, цепных правил.
17. Автомат с магазинной памятью (МП-автомат): определение, назначение, схема работы, пример,

18. Определение конфигурации МП-автомата (текущее состояние автомата), описание такта работы МП-автомата.
19. Формальное определение МП-автомата. Алгоритм работы МП-автомата, три возможных случая выполнения одного шага работы МП-автомата.
20. Автомат с магазинной памятью: построение МП-автомата.
21. Алгоритмы синтаксического разбора. Обратная польская запись форма записи выражений (ПОЛИЗ). Алгоритм построения обратной польской записи. Алгоритм вычисления выражения в обратной польской записи и возможность расширения алгоритма.
22. Приведенные КС-грамматики: нормальная форма Хомского контекстно-свободной грамматики. Алгоритм приведения контекстно-свободной грамматики к нормальной форме Хомского. Примеры.
23. Приведенные КС-грамматики: нормальная форма Грейбах контекстно-свободной грамматики. Алгоритм приведения контекстно-свободной грамматики к нормальной форме Грейбах. Примеры.
24. Создание нового языка программирования. Основные элементы языков программирования. Система типов языка программирования, парадигмы программирования, способы реализации языков программирования. Классы языков программирования. Общая схема работы компилятора. Основные принципы реализации.
25. Язык Ассемблера: структура программы, типы данных, директивы определения данных, регистры общего назначения центрального процессора: назначение, названия. Примеры.
26. Определения: язык Ассемблера, транслятор с языка Ассемблера, уровень языка Ассемблера. Структура программы на Ассемблере, модели памяти Ассемблера.
27. Язык Ассемблера: регистры общего назначения, регистр флагов. Примеры.
28. Язык Ассемблера: типы данных, директивы для определения данных, инициализация данных, синтаксис оператора определения данных, массивы и их инициализация. Примеры.
29. Язык Ассемблера: формат команды Ассемблера, основные типы операндов команды Ассемблера. Примеры.
30. Язык Ассемблера. Препроцессор: символы, директива присваивания, счетчик команд, директива EQU.
31. Язык Ассемблера: формат команды Ассемблера, команды пересылки данных, команды целочисленного сложения и вычитания, операторы OFFSET, PTR, TYPE, SIZEOF и LENGTHOF. Примеры.
32. Адресация в Ассемблере: прямая и косвенная адресация. Косвенная адресация: с косвенным операндом. Косвенная адресация: операнды с индексом. Указатели и их реализация на Ассемблере. Примеры.
33. Язык Ассемблера: операции работы со стеком, команды переходов (безусловные, условные), команда LOOP. Назначение, использование. Примеры.
34. Язык Ассемблера: создание статической библиотеки, взаимодействие вызов функций, написанных на разных языках программирования. Примеры.
35. Язык Ассемблера: особенности программирования на MASM для архитектуры x64, основные возможности и ограничения. Регистры общего назначения. Соглашение о передаче параметров в функцию для архитектуры x86-64.
36. Назначение семантического анализа, входные данные, этапы семантического анализа. Примеры соглашений языка, проверка которых обязательна.

37. Место семантического анализатора в процессе компиляции. Статические и динамические семантические проверки. Основные функции семантического анализатора.
38. Семантический анализ: определение, назначение, задачи семантического анализа применение.
39. Генерация промежуточного кода: определение, место в процессе компиляции, способы внутреннего представления программ.
40. Генерация промежуточного кода: многоадресный код с явно именуемым результатом (тетрады).
41. Генерация промежуточного кода: многоадресный код с неявно именуемым результатом (триады).
42. Генерация промежуточного кода: обратная (постфиксная) польская запись для внутреннего представления программы.
43. Генерация промежуточного кода: ассемблерный код и машинные команды.
44. Генерация кода: задачи, входная и выходная информация, принципы и подходы к генерации кода.
45. Подходы к разработке трансляторов: трансляция исходного кода в язык ассемблера для некоторой аппаратной платформы.
46. Подходы к разработке трансляторов: компиляторы-интерпретаторы.
47. Генерация и оптимизация кода: определение, назначение, критерии и методы оптимизации. Понятие машинно-независимой и машинно-независимой оптимизации кода.
48. Стандартизация информационных технологий: основные понятия, подходы к разработке стандартов, структура, поддерживающая процесс стандартизации ИТ (уровни стандартизации, примеры).

Лектор

Наркевич А.С.