

Приложение *(обязательное)*

Фонд оценочных средств по дисциплине "Языки и системы программирования. Технология разработки программного обеспечения "

ФОС оформлен в виде отдельного файла и является неотъемлемой и обязательной составляющей рабочей программы дисциплины.

1. Контроль и оценка результатов освоения учебной дисциплины

Контроль и оценка результатов освоения учебной дисциплины осуществляются преподавателем в процессе проведения устных опросов, докладов и дискуссий, а также при проверке рефератов, выполненных аспирантами.

Для итоговой оценки усвоения дисциплины учебным планом предусмотрен зачет. Необходимым, но не достаточным, условием допуска к зачету является выполнение рефератов.

2. Требования к устным докладам

К устным докладам аспирантов предъявляются следующие требования:

- содержание доклада должно соответствовать теме доклада;
- доклад должен иметь достаточную полноту и глубину охвата темы;
- выступление должно быть четким, аргументированным и выразительным.

3. Требования к оформлению и содержанию рефератов

К реферату предъявляются следующие требования:

- содержание реферата должно соответствовать теме;
- объем реферата должен быть в пределах от 3 до 8 листов;
- реферат должен быть выполнен в печатном виде, рукописном виде или в виде вложения в формате "DOC" по e-mail;
- реферат в печатном оформлении должен иметь шрифт Times New Roman 12, междустрочный интервал 1,25;

- титульный лист для рефератов выполняется стандартным способом, т.е. должен содержать наименование, образовательного учреждения, темы реферата, Ф.И.О. исполнителя, Ф.И.О. преподавателя, год.

При оценке реферата учитываются степень соответствия содержания теме реферата, полнота охвата и глубина знания, четкость ответа, уровень изложения материала аспирантами.

4. Требования к зачету

Для получения допуска к зачету аспиранту необходимо своевременно выполнить реферат. Если реферат не был выполнен вовремя, аспирант не допускается до зачета.

К зачетной работе предъявляются следующие требования:

- работа должна выполняться на базе пройденных тем письменно;
- работа должна быть выполнена в аудитории в течение 45 мин.;
- задания аспирантам должны быть произвольными в пределах списка вопросов по пройденным темам.

При оценке качества зачетной работы учитываются степень соответствия теме вопроса, полнота охвата и глубина знания, четкость ответа, уровень изложения материала аспирантами.

5. Перечень примерных вопросов для зачета и тем рефератов и докладов

1. Языки программирования, их классификация.
2. Трансляторы и их виды.
3. Синтаксис и семантика языков программирования.
4. Цепочки и языки.
5. Порождающие грамматики, распознаватели и преобразователи.
6. Классификация Хомского.
7. Лексемы и понятия, Бэкуса-Наура формы (БНФ) и синтаксические диаграммы.
8. Операционная, трансформационная и денотационная семантика.
9. Стандарты и версии языков программирования.

10. Языки программирования высокого уровня. Иерархия языковых конструкций.
11. Система типов языка высокого уровня, простые, составные, первичные, стандартные, библиотечные и конструируемые типы.
12. Статическая и динамическая типизация, свойства языков со строгой типизацией.
13. Процедуры и функции, простая структура вызовов-возвратов, задачи, сопрограммы и подпрограммы обработки исключительных ситуаций.
14. Блоки и правила локализации имен.
15. Механизм параметров, ключевые и позиционные параметры, способы подстановки параметров.
16. Рекурсия.
17. Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы.
18. Понятие об объектном окружении. Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).
19. Распределенное программирование, процессы и их синхронизация, семафоры, мониторы Хоара.
20. Объектно-ориентированное распределенное программирование. CORBA.
21. Параллельное программирование над общей памятью. Нити. Стандартный интерфейс Open MP.
22. Распараллеливание последовательных программ.
23. Параллельное программирование над распределенной памятью. Парадигмы SPMD и MIMD. Стандартный интерфейс MPI.
24. Этапы и схемы трансляции.
25. Построение абстрактной программы: лексический, синтаксический и контекстный анализы. Генерация.
26. Оптимизация и схема оптимизирующей трансляции.
27. Промежуточные представления программ.

- 28.Обнаружение ошибок при трансляции.
- 29.Конечные автоматы и их свойства.
- 30.Реализация лексического анализа.
- 31.Автоматы с магазинной памятью и их свойства.
- 32.Стратегии синтаксического анализа и обработки ошибок.
- 33.LL-грамматики и LL-анализаторы.
- 34.LR-грамматики LR-анализаторы.
- 35.Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик, системы LEX и YACC.
- 36.Контекстный анализ: идентификация и атрибутивная индукция.
- 37.Атрибутные грамматики и алгоритмы вычисления атрибутов.
- 38.Атрибутные системы построения трансляторов.
- 39.Смешанная стратегия при трансляции и оптимизирующие преобразования. Локальная, квазилокальная и глобальная оптимизация.
- 40.Схемы программ и промежуточные представления программ.
- 41.Анализ потока управления: представление и структуризация множеств исполнений, нахождение свойств операторов и переходов, выбор порядка обработки операторов.
- 42.Анализ потока данных: свойства исполнений и состояний.
- 43.Способы оптимизации программ: разгрузка участков повторяемости, упрощение действий, чистка программы, экономия памяти, сокращение программы.
- 44.Выбор последовательности применения оптимизирующих преобразований и контекстных условий.
- 45.Общая схема генерации объектного кода в компиляторах и внутреннее представление команд.
- 46.Представление структур данных. Генерация выражений, операторов и подпрограмм.
- 47.Распределение памяти. Динамическая поддержка.

48. Покадровая оптимизация.
49. Перенастраиваемые (retargetable) компиляторы.
50. Автоматическая генерация генераторов объектного кода (системы BEG, Iburg и др.).
51. Машинно-зависимые и машинно-ориентированные языки.
52. Язык ассемблера, представление машинных команд и констант, команды транслятору и их реализация.
53. Макросы и макропроцессоры.
54. Системы программирования и их типовые компоненты: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.
55. Применение макросредств при построении трансляторов. Методы обеспечения мобильности.
56. Метод раскрутки.
57. Модульный подход при построении трансляторов.
58. Пакеты прикладных программ (ППП). Системная часть и наполнение. Языки общения с ППП.
59. Машинная графика. Средства поддержки машинной графики. Графические пакеты.
60. Технология разработки и сопровождения программ. Жизненный цикл программы. Этапы разработки, степень и пути их автоматизации.
61. Обратная инженерия.
62. Декомпозиционные и сборочные технологии, механизмы наследования, инкапсуляции, задания типов.
63. Модули, взаимодействие между модулями, иерархические структуры программ.
64. Отладка, тестирование, верификация и оценивание сложности программ.
65. Генерация тестов. Системы генерации тестов.
66. Срезы программ (slicing) и их применение при отладке программ и для генерации тестов.

67. Методы спецификации программ. Методы проверки спецификации.
68. Схемное, структурное, визуальное программирование.
69. Языки функционального и логического программирования.
70. Разработка пользовательского интерфейса, стандарт CUA, мультимедийные среды интерфейсного взаимодействия.