1. Классификация ПО На какие классы подразделяют ПО?

[\*] а) Системное программное обеспечение

[\_] б) Программы управления языком интерфейса информационных систем [\*] в) Прикладное программное обеспечение

[\_] г) Системные обрабатывающие программы [\_] д) Базовое программное обеспечение

1. Классификация СПО

На какие классы подразделяют системное ПО по первой классификации? [\*] а) Системные управляющие программы

[\_] б) Программы управления языком интерфейса информационных систем [\_] в) Прикладное программное обеспечение

[\*] г) Системные обрабатывающие программы [\_] д) Базовое программное обеспечение

1. Классификация СПО [1]

На какие классы подразделяют системное ПО по второй классификации? [\_] а) Системные управляющие программы

[\_] б) Программы управления языком интерфейса информационных систем [\*] в) Сервисное программное обеспечение

[\_] г) Системные обрабатывающие программы [\*] д) Базовое программное обеспечение

1. Назначение СПО Системное программное обеспечение предназначено для:

[\*] а) создания операционной среды функционирования других программ [\_] б) решения функциональных задач

[\*] в) обеспечения надежной и эффективной работы самого компьютера и вычислительной сети

[\*] г) выполнения вспомогательных технологических процессов

[\*] д) проведения диагностики и профилактики аппаратуры компьютера и вычислительных сетей

[\_] е) обработку информации различных предметных областей

1. Сервисное ПО

В сервисное ПО входят:

[\*] а) антивирусные программы, обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов

[\*] б) драйверы специфических и специальных устройств

[\_] в) интерфейсные оболочки для взаимодействия пользователя с ОС (операционные оболочки) и программные среды

[\*] г) программы архивирования данных, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения

[\_] д) системы управления файлами

1. Базовое ПО

В базовое ПО входят:

[\_] а) антивирусные программы, обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов

[\*] б) операционные системы и драйверы в составе ОС

[\*] в) интерфейсные оболочки для взаимодействия пользователя с ОС (операционные оболочки) и программные среды

[\_] г) программы архивирования данных, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения

[\*] д) системы управления файлами

1. Система программирования

Система программирования относится к системному программному обеспечению и включает следующие программные компоненты:

[\*] а) редактор текста

[\_] б) система управления файлами

[\_] в) система управления базами данных [\*] г) транслятор с соответствующего языка [\*] д) отладчик

1. Функции отладчика

Отладчик позволяет управлять процессом исполнения программы. Базовый набор функций отладчика включает:

[\_] а) ввода и модификации текста

[\*] б) остановка в заранее определенных точках

[\_] в) преобразование программ, написанных на языках программирования, в программы на машинном языке

[\*] г) пошаговое выполнение программы (режим трассировки) с отображением результатов

[\*] д) возможность остановки в некотором месте программы при выполнении некоторого условия

[\_] е) редактировать и объединять объектные (ранее оттранслированные) модули в единые загрузочные

[\*] ж) изображение и изменение значений переменных.

1. Функции редактора

Редактор системы программирования содержит следующий набор функций: [\*] а) ввода и модификации текста

[\_] б) остановка в заранее определенных точках

[\_] в) преобразование программ, написанных на языках программирования, в программы на машинном языке

[\_] г) пошаговое выполнение программы (режим трассировки) с отображением результатов

[?] д) вывод интерактивных подсказок о структуре классов и наборе методов

[\_] е) редактировать и объединять объектные (ранее оттранслированные) модули в единые загрузочные

[\_] ж) изображение и изменение значений переменных.

1. Требования к СПО

Системные программное обеспечение должно удовлетворять следующим требованиям: [\_] а) работа с большинством пакетов прикладного программного обеспечения

[\*] б) гарантированная надежность выполнения в соответствии со спецификациями [\*] в) минимальные затраты на хранение машинных кодов;

[\_] г) минимальная скорость выполнения

[\*] д) поддержка стандартных средств связи с прикладными программами

1. Файловая система

Файловая система - это набор спецификаций и соответствующее им программное обеспечение, которые отвечают за

[\*] а) создание и уничтожение файловой информации [\_] б) архивирование файловой информации

[\*] в) чтение и запись файловой информации [\_] г) проверка логической целостности файлов

1. Определение файлов\_системы “Файловая система”

[\*] а) определяет принципы доступа к данным, организованным в файлы

[\_] б) комплекс программных модулей, обеспечивающих работу с файлами в конкретной ОС

[\_] в) набор спецификаций и соответствующее им программное обеспечение

1. Система управления файлами опр. “Система управления файлами”

[\_] а) определяет принципы доступа к данным, организованным в файлы

[\*] б) комплекс программных модулей, обеспечивающих работу с файлами в конкретной ОС

[\_] в) набор спецификаций и соответствующее им программное обеспечение

1. Разделы диска Какие типы разделов диска существуют?

[\*] а) Первичный [\_] б) Вторичный [\*] в) Расширенный [\_] г) Укороченный

1. Физический адрес Физический адрес сектора на диске определяется:

[\*] а) триадой [c-h-s], где c - номер цилиндра (cylinder), h - номер рабочей поверхности диска (магнитной головки, head), s - номер сектора на дорожке (sector).

[\_] б) парой [c-h], где c - номер цилиндра (cylinder), h - номер рабочей поверхности диска (магнитной головки, head)

[\_] в) тетрадой [c-h-s -p], где c - номер цилиндра (cylinder), h - номер рабочей поверхности диска (магнитной головки, head), s - номер сектора на дорожке (sector), p - смещение относительно начала сектора.

1. Главная загрузочная запись Главная загрузочная запись - MBR - состоит из

[\_] а) таблица файловой системы

[\*] б) программы начальной загрузки (внесистемный загрузчик) [\_] в) основной каталог

[\*] г) таблица разбиения диска если ответ один, то это, если два то оба

1. Таблица разделов

Таблица разделов содержит

[\*] а) сведения о положении и характеристиках имеющихся на винчестере разделов [\_] б) данные о размере диска и файловых системах

[?] в) располагается в MBR по смещению Оx1ВЕ и занимает 64 байта [\_] г) располагается в secondary boot record

1. FAT\_1

В файловой системе FAT дисковое пространство любого логического диска делится на следующие области:

[\_] а) файловую область [\*] б) системную область [\*] в) область данных

[\_] г) главный каталог

1. Системная область FAT

Системная область создается и инициализируется при форматировании. Системная область состоит из следующих компонентов:

[\*] а) загрузочного сектора, содержащего загрузочную запись (boot record) [\_] б) главной загрузочной записи

[\*] в) зарезервированных секторов (их может и не быть) [\*] г) корневого каталога (Root directory, ROOT)

[\_] д) таблицы разделов

[\*] е) таблицы размещения файлов (FAT, File Allocation Table)

1. Таблица размещения файлов FAT

Разбиение области данных на кластеры вместо использования секторов позволяет: [\*] а) уменьшить размер таблицы FAT

[\_] б) уменьшить размер файлов

[\*] в) сокращается длина цепочек файла, а, следовательно, ускоряется доступ к файлу [\_] г) увеличить системную область диска

[\*] д) уменьшить фрагментацию файлов

1. Основные недостатки FAT

Основные недостатки FAT:

[\*] а) медленная работа с файлами

[\*] б) фрагментация диска

[\_] в) отсутствие резервной копии

[\*] г) сложные цепочки файлов

[?] д) ограничения на размер файлов

1. Корневой каталог FAT Элементом Корневого каталога FAT является:

[\_] а) логическое имя файла ПЛИСЕНКО СКАЗАЛА ЭТО

[\_] б) физический адрес файла на диске

[\*] в) дескриптор (описатель) файла В ИНЕТЕ ЭТО

[\_] г) цепочка кластеров файла

23

Дескриптор каждого файла и каталога в FAT содержит:

[\*] а) имя (8 байт)

[\*] б) время создания или последней модификации (2) [\_] в) права доступа для групп пользователей

[\*] г) длину файла (для каталога - 0) (4) [\_] д) количество кластеров для файла

24

Перечислите основные особенности HPFS:

[\_] а) Таблица размещения файлов имеет структуру простого списка

[\_] б) Для поиска информации на диске используется метод рехеширования

[\*] в) Дисковое пространство в HPFS выделяется не кластерами (как в FAT), а блоками. [\*] г) Используется метод сбалансированных двоичных деревьев для хранения и поиска информации о местонахождении файлов

[\_] д) Основной каталог файлов имеет ограничение на размер

[\*] е) Система HPFS стремится расположить файл в смежных блоках, или, если такой возможности нет, разместить его на диске таким образом, чтобы экстенты (фрагменты) файла физически были как можно ближе друг к другу.

25

Для сокращения времени позиционирования головок чтения/записи жесткого диска система HPFS стремится

[\*] а) расположить файл в смежных блоках [\_] б) сжать файлы в один блок

[\_] в) архивировать блоки, относящиеся к разным экстентам

[\*] г) разместить экстенты фрагментированного файла как можно ближе друг к другу

26

Проблемы, возникающие при работе с HPFS

[\_] а) Автоматическое удаление малоиспользуемых файлов

[\*] б) При переименовании файлов может возникнуть так называемая перебалансировка дерева

[\_] в) Разрыв цепочек при каскадном переименовании файлов

[\*] г) Создание файла, переименование или стирание может приводить к каскадированию блоков каталогов

27

Главный служебный метафайл файловой системы NTFS занимает: [\*] а) Первые 12 % диска

[\_] б) Первые 25 % диска [\_] в) Первые 88 % диска

28

Каждый файл в NTFS представлен с помощью:

[\_] а) кластеров служебной информации и данных [\*] б) потоков (streams)

[\_] в) логического имени набора данных [\_] г) набора атрибутов файла

29

Атрибуты файлов в системе NTFS могут включать следующие значения: [\_] а) Зашифрованное имя файла в символах Unicode.

[\*] б) Структура данных защиты (ACL), предохраняющая файл от несанкционированного доступа. Атрибут “дескриптор защиты” определяет, кто владелец файла и, кто имеет доступ к нему

[\*] в) Традиционные атрибуты Read Only, Hidden, Archive, System, отметки времени, включая время создания или последней модификации, число каталогов, ссылающихся на файл

[\_] г) Параметры доступа через глобальную сеть

30

Интерпретаторы отличаются от компиляторов отсутствием следующих фаз: [\_] а) Лексический анализ

[?] б) Синтаксический анализ [\*] в) Генерация кода

[\*] г) Оптимизация кода

31

Часть компилятора, проверяющая правильность текста исходной программы с точки зрения смысла программы на входном языке, называется:

[\_] а) Синтаксический анализ [\_] б) Лексический анализ

[\_] в) Генерация кода

[\*] г) Семантический анализ

32

В фазу подготовки к генерации кода компилятор выполняет следующие действия: [\_] а) выделение лексем

[\*] б) идентификацией элементов языка [\*] в) распределением памяти

[\_] г) генерация кода

33

После выполнения лексического анализа, на вход синтаксического анализа в качестве входных данных подаются:

[\_] а) текст входной программы

[\_] б) текст входной программы с исключенными комментариями [\_] в) таблица идентификаторов и таблица лексем

[\*] г) таблица лексем

1. Компиляторы\_5

Таблицы идентификаторов — это специальным образом организованные наборы данных, служащие для хранения информации об элементах исходной программы, которые затем используются для порождения текста результирующей программы. В таблице идентификаторов хранится информация о следующих элементах программы:

[\*] а) идентификаторах

[\*] б) константах [\_] в) типах данных

[\_] г) значимых лексемах

[\_] д) зарезервированных словах [\_] е) комментариях

1. Компиляторы\_6

Проход компилятора - это:

[\_] а) процесс чтения компилятором таблицы лексем, и удаления комментариев [\_] б) процесс последовательной обработки синтаксических конструкций языка и параллельной генерации кода

[\_] в) процесс оптимизации заранее сгенерированного кода

[\*] г) процесс последовательного чтения компилятором данных из внешней памяти, их обработки и помещения результата работы во внешнюю память

1. Комп\_проход (1)] Выберите достоинства однопроходных компиляторов:

[\_] а) хорошо справляется со сложной грамматикой входного языка [\*] б) увеличивается скорость работы компилятора

[\*] в) сокращается объем необходимой компилятору памяти

[\_] г) позволяет использовать простые алгоритмы синтаксического разбора

1. Комп\_ассембл (2)

Какие особенности имеет язык ассемблер, которые упрощают реализацию компилятора для него?

[\*] а) допускает использование в качестве операндов определенных ограниченных сочетаний обозначений регистров, идентификаторов и констант, которые объединены некоторыми знаками операций

[\_] б) содержит конструкции для описания сложных структур данных [\_] в) не содержит операторов безусловного перехода

[\*] г) ряд идентификаторов в нем выделяется специально для обозначения регистров процессора

1. Таблицы идентификаторов

В таблице идентификаторов содержится информация об идентификаторах программы (переменных, константах, именах функций и т.д.), включающая:

[\*] а) имя переменной

[\_] б) длина и состав комментариев

[\*] в) область памяти, связанная с переменной

[\*] г) имя функции, количество и типы формальных аргументов функции, тип возвращаемого результата

[\_] д) зарезервированные слова

1. Таблицы идентификаторов\_1

Основные требования к методу построения таблиц идентификаторов содержат следующее:

[\_] а) таблица идентификаторов должна иметь фиксированную длину

[ ] б) таблица идентификаторов не должна содержать не описанные идентификаторы [\*] в) метод поиска по таблице идентификаторов должен быть максимально быстрым

[\*] г) размер таблицы идентификаторов должен изменяться динамически в зависимости от количества идентификаторов в программе

[\_] д) идентификаторы должны быть занесены в таблицу последовательно согласно порядку поступления

1. Таблицы идентификаторов\_2

:

(Тз - время добавления элемента в таблицу; Тп - время поиска элемента в таблице) [\*] а) Тз не зависит от числа элементов в таблице, Тп = О(N).

[\_] б) Tз = O(N\*log2N) +k\*O(N2), Тп = О(log2N). [\_] в) Тз = N\*O(log2 N), Tп = O(log2 N)

[\_] г) среднее время на размещение одного элемента и на поиск элемента в таблице зависит только от среднего числа коллизий

1. Таблицы идентификаторов\_3

Построение таблицы идентификаторов по методу цепочек имеет следующую сложность: (Тз - время добавления элемента в таблицу;

Тп - время поиска элемента в таблице)

[\_] а) Тз не зависит от числа элементов в таблице, Тп = О(N). [\_] б) Tз = O(N\*log2N) +k\*O(N2), Тп = О(log2N).

[\_] в) Тз = N\*O(log2 N), Tп = O(log2 N)

[\*] г) среднее время на размещение одного элемента и на поиск элемента в таблице зависит только от среднего числа коллизий

1. Таблицы идентификаторов\_4

Построение таблицы идентификаторов по методу бинарного дерева имеет следующую сложность:

(Тз - время добавления элемента в таблицу; Тп - время поиска элемента в таблице)

[\_] а) Тз не зависит от числа элементов в таблице, Тп = О(N). [\_] б) Tз = O(N\*log2N) +k\*O(N2), Тп = О(log2N).

[\_] в) Тз = N\*O(log2 N), Tп = O(log2 N)

[\*] г) среднее время на размещение одного элемента и на поиск элемента в таблице зависит только от среднего числа коллизий

1. Таблицы идентификаторов\_5

Метод построения таблицы идентификаторов с использованием хэш-функций имеет следующие недостатки:

[\_] а) необходимость компактного хранения информации об идентификаторе [\*] б) неэффективное использование объема памяти

[\_] в) время записи нового идентификатора всегда намного больше времени поиска [\*] г) необходимость соответствующего разумного выбора хэш-функции

[\_] д) при каждой записи нового идентификатора требуется полная перестройка таблицы

1. Цепоч\_символов\_1

Операция конкатенации цепочек символов обладает следующими свойствами:

1. коммутативности
2. (αβ)R= βR αR
3. ассоциативности
4. αn = ααααα….. [\_] а) 1,2,4

[\*] б) 2,3,4

[\_] в) 2,3

1. Алфавит – это:

[\_] а) заданный набор символов и правил, устанавливающих способы комбинации этих символов между собой для записи осмысленных текстов

[\*] б) счетное множество допустимых символов языка

[\_] в) множество всех печатных символов

1. Язык – это:

[\*] а) заданный набор символов и правил, устанавливающих способы комбинации этих символов между собой для записи осмысленных текстов

[\_] б) счетное множество допустимых символов языка

[\_] в) множество всех печатных символов

1. Язык\_1 Способами задания языка являются:

[\*] а) перечисление всех допустимых цепочек языка

[\_] б) определение смысла каждой устоявшейся конструкции

[\*] в) построение распознавателя цепочек языка

[\*] г) указание способа порождения цепочек

[\_] д) указание допустимых операций над цепочками языка

1. Язык\_2 Для задания языка программирования необходимо решить следующие вопросы:

[\*] а) задать смысл для каждой правильной программы

[\_] б) определить набор операций над элементами языка

[\*] в) определить множество правильных программ языка

[\_] г) определить структуру таблиц идентификаторов и лексем

[\*] д) определить множество допустимых символов языка

1. Грамматика Грамматика языка может быть задана:

[\*] а) в формальном виде, в правилах по форме Бэкуса-Наура

[\*] б) в виде правил в графическом виде

[\*] в) через задание алфавита и правильно построенных цепочек языка

[\_] г) через определение правильно построенных фраз

[\_] д) через определение правил с использованием метасимволов

1. Классификация грамматик К какому классу языков относятся языки, поддерживающие грамматику со следующей структурой правил: α ->β, где α,β є V+ , |β|≥|α|

[\_] а) грамматики с фразовой структурой

[\*] б) контекстно-зависимые и неукорачивающие

[\_] в) контекстно-свободные

[\_] г) регулярные

1. Классификация грамматик1 К какому классу языков относятся языки, поддерживающие грамматику со следующей структурой правил: α ->β, где α є V+ , β є V\*

[\*] а) грамматики с фразовой структурой

[\_] б) контекстно-зависимые и неукорачивающие

[\_] в) контекстно-свободные

[\_] г) регулярные

1. Классификация грамматик2 К какому классу языков относятся языки, поддерживающие грамматику со следующей структурой правил: A ->β, где A є VN, β є V\*

[\_] а) грамматики с фразовой структурой

[\_] б) контекстно-зависимые и неукорачивающие

[\*] в) контекстно-свободные (укорачивающие)

[\_] г) регулярные

1. Классификация грамматик3 К какому классу языков относятся языки, поддерживающие грамматику со следующей структурой правил: A ->γB или A->γ, где A,B є VN, γ є VT\*

[\_] а) грамматики с фразовой структурой

[\_] б) контекстно-зависимые и неукорачивающие

[\_] в) контекстно-свободные

[\*] г) регулярные

1. Классификация грамматик4 К регулярному типу языков принадлежат:

[\_] а) языки словарей

[\_] б) естественные языки

[\*] в) простейшие конструкции языков программирования

[\*] г) языки ассемблеров

[\_] д) языки командных процессоров

1. Лексические анализаторы Основные функции лексических анализаторов:

[\_] а) добавление неявных описаний типов идентификаторов

[\_] б) дописывание неявных определений функций

[\*] в) определение типов лексем: констант, знаков операций, зарезервированных слов, разделителей....

[\*] г) исключение из текста незначащих пробелов, символов табуляции и перевода строки

[\_] д) выявление синтаксических ошибок

1. Лексические анализаторы1

Недостатками параллельной работы лексического и синтаксического анализаторов являются

[\_] а) неясное разграничение функций лексического и синтаксического анализа [\_] б) неудобство в использовании таблицы лексем и таблицы идентификаторов [\*] в) сложность алгоритма реализации

[\*] г) возможность вернуться к фазе лексического анализа при нечетком определении синтаксической конструкции

[\*] д) сложный механизм обмена данными и повторное прочтение уже разобранных лексем

1. Лексические анализаторы2

Для построения лексического анализатора используется математический аппарат:

[?] а) двусторонних недетерминированных автоматов с линейно ограниченной внешней памятью

[\_] б) недетерминированных двусторонних автоматов, имеющих неограниченную внешнюю память

[\_] в) детерминированных автоматов без магазинной памяти [\_] г) недетерминированных автоматов без магазинной памяти

[\*] д) недетерминированных автоматов с магазинной внешней памятью

1. Синтаксические анализаторы НАШЛИ Г – ОСТАЛЬНЫЕ ХЗ

Для построения синтаксических анализаторов контекстно –свободных языков используется математический аппарат:

[\_] а) двусторонних недетерминированных автоматов с линейно ограниченной внешней памятью

[\_] б) недетерминированных двусторонних автоматов, имеющих неограниченную внешнюю память

[\_] в) детерминированных автоматов без магазинной памяти

[\*] г) односторонние недетерминированные автоматы с магазинной внешней памятью [\_] д) недетерминированных автоматов с магазинной внешней памятью

1. Синтаксические анализаторы\_1 В задачу синтаксического анализа входит:

[\*] а) поиск и выделение синтаксических конструкции в тексте исходной программы

[\_] б) сохранение информации об обнаруженной лексеме (или выдача сообщения об ошибке, если лексема неверна)

[\_] в) проведение проверки соблюдения во входной программе семантических соглашений входного языка

[\*] г) установка типа и проверка правильности каждой синтаксической конструкции [\*] д) представление синтаксических конструкций в виде, удобном для дальнейшей генерации текста результирующей программы

[\_] е) распределение памяти для переменных исходной программы

1. МП автомат (1)

В чем отличие МП-автомата от обычного КА

[\*] а) конфигурация автомата определяется тремя параметрами: состоянием автомата, входной цепочкой символов и положением указателя на символ во входной цепочке [\_] б) функция переходов зависит не только от текущего состояния, но и от символа из алфавита V, поданного на вход автомата

[\_] в) имеет стек (магазин), в который можно помещать специальные «магазинные» символы

[\_] г) переход из одного состояния в другое зависит не только от входного символа, но и от символа на верхушке стека

1. МП автомат (2)

В чем отличие расширенного МП-автомата от обычного МП - автомата?

[\_] а) допускает цепочку символов с опустошением магазина, если при окончании разбора цепочки автомат находится в одном из конечных состояний

[\*] б) может заменять цепочку символов конечной длины в верхней части стека на другую цепочку символов конечной длины

[\_] в) переход из одного состояния в другое зависит не только от входного символа, но и от символа на верхушке стека

[\_] г) при выполнении перехода из стека удаляется верхний символ, соответствующий условию перехода, и добавляется цепочка, соответствующая правилу перехода

1. МП автомат (3)

Какие из утверждений истины для детерминированного МП автомата?

[\_] а) допускаются переходы, при которых входной символ игнорируется (и, тем самым, он будет входным символом при следующем переходе)

[\*] б) все языки, принадлежащие к классу детерминированных КС-языков и описываемых с помощью детерминированного МП автомата, могут быть построены с помощью однозначных КС-грамматик

[\_] в) конфигурация МП-автомата описывается в виде тройки которая определяет: текущее состояние автомата, цепочку еще не прочитанных символов на входе автомата и содержимое магазина

[\_] г) из каждой его конфигурации возможно не более одного перехода в следующую конфигурацию

1. Распознаватели (1)

Для каких групп распознавателей используется алгоритм с подбором альтернатив? [\_] а) для линейных распознавателей

[\_] б) для универсальных распознавателей [\_] в) для восходящих распознавателей [\*] г) для нисходящих распознавателей

1. Распознаватели (2)

Для каких групп распознавателей используется алгоритм "сдвиг - свертка"? [\_] а) для линейных распознавателей

[\_] б) для универсальных распознавателей [\*] в) для восходящих распознавателей [\_] г) для нисходящих распознавателей

(65)

Каковы цели преобразования КС - грамматик?

[\_] а) запись правил грамматики в форме Бэкуса - Наура [\_] б) удаление всех нетерминальных символов

[\*] в) упрощение правил грамматики

[\*] г) облегчение создания распознавателя языка

66

Какие из преобразований грамматики используются для упрощения правил? [\*] а) удаление бесплодных символов

[\_] б) устранения левой рекурсии [\*] в) устранение цепных правил [\_] г) удаление лямбда правил

67

Какие КС грамматики можно отнести к одному из классов грамматик предшествования? [\_] а) левосторонние неукорачивающие грамматики

[\_] б) все приведенные КС грамматики

[\*] в) грамматики для которых возможно определить отношения предшествования между символами грамматики

[\_] г) LR - грамматики

68

Для какого класса грамматик предшествования справедливо следующее утверждение: "Различные правила в грамматике имеют разные правые части (то есть в грамматике не должно быть двух различных правил с одной и той же правой частью)" ?

[\*] а) простого предшествования

[\_] б) расширенного предшествования [\_] в) слабого предшествования

[\_] г) смешанной стратегии предшествования [\_] д) операторного предшествования

69

Для какого класса грамматик предшествования справедливо следующее утверждение: "КС-грамматика без лямбда-правил, в которой правые части всех правил не содержат смежных нетерминальных символов" ?

[\_] а) простого предшествования

[\_] б) расширенного предшествования [\_] в) слабого предшествования

[\_] г) смешанной стратегии предшествования [\*] д) операторного предшествования

70

Для чего служат отношения предшествования?

[\_] а) для определения алгоритма распознавания

[\_] б) для построения множества терминальных символов [\*] в) для определения типа действия (сдвиг или свертка) [\_] г) для определения текущего состояния автомата

[\_] д) однозначного выбора цепочки для свертки

71

В чем заключаются основные свойства отношений предшествования?

[\*] а) отношение является единственным для каждой упорядоченной пары символов [\*] б) зависят от порядка, в котором стоят символы

[\_] в) включают все знаки арифметических операций

[\_] г) между какими-либо двумя символами может и не быть отношения предшествования [\_] д) аналогичны математическим символам

72

Какие из перечисленных действий выполняет семантический анализатор? [\_] а) построение дерева синтаксического разбора

[\*] б) дополнение внутреннего представления программы в компиляторе операторами и действиями, неявно предусмотренными семантикой входного языка

[\_] в) распределение памяти для сложных структур данных [\_] г) правильность синтаксических конструкций

[\*] д) проверку элементарных семантических (смысловых) норм языков программирования, напрямую не связанных со входным языком

73

Какие из перечисленных действий выполняет семантический анализатор? [\_] а) построение дерева синтаксического разбора

[\*] б) дополнение внутреннего представления программы в компиляторе операторами и действиями, неявно предусмотренными семантикой входного языка

[\_] в) распределение памяти для сложных структур данных

[\*] г) проверка соблюдения во входной программе семантических соглашений входного языка

[\_] д) выделение лексем входного языка

74

Выберите правильное определение этапа генерации кода.

[\_] а) часть компилятора, которая читает литеры программы на исходном языке и строит из них слова исходного языка

[\_] б) часть компилятора, в которую входят действия, связанные с идентификацией элементов языка, распределением памяти и т. п.

[\*] в) перевод компилятором внутреннего представления исходной программы в цепочку символов выходного языка

[\_] г) часть компилятора, проверяющая правильность текста исходной программы с точки зрения семантики входного языка

75

Что такое синтаксически управляемый перевод и для чего он нужен? [\_] а) метод идентификации лексем входного языка

[\_] б) основной метод порождения кода результирующей программы на основании результатов синтаксического анализа

[\_] в) метод распределения памяти для сложных структур исходной программы

[\*] г) алгоритм, воспроизводящий работу МП автомата