

УТВЕРЖДАЮ

Зав. Кафедры программной инженерии БГТУ

к.т.н., доцент _____ Н.В. Пацей

протокол № 8 от 05 мая 2021 г.

Экзаменационные вопросы

Языки программирования, 1 курс ПОИТ, весенний семестр 2019

1-40 01 01- Программное обеспечение информационных технологий (специализация «Программирование интернет-приложений»)

1. Система программирования: определение, состав, назначение. Классический жизненный цикл разработки программного обеспечения. Текстовый редактор. Интегрированная среда разработки. Программный продукт. Общая схема преобразования исходного кода в процесс операционной системы.
2. Система программирования: определение, состав, назначение. Интегрированная среда разработки: определение, состав, назначение. Транслятор: виды трансляторов. Транслятор: определение, назначение, примеры. Интерпретатор: определение, назначение, примеры.
3. Язык программирования: определение, назначение, примеры. Исходный код. Язык программирования. Алфавит языка. Транслятор: определение, назначение, примеры. Объектный код, объектный модуль. Общая схема преобразования исходного кода в процесс операционной системы.
4. Язык программирования: определение, назначение, примеры. Исходный код. Язык ассемблера. Ассемблер. Транслятор с ассемблера: определение, назначение, примеры. Объектный код, объектный модуль. Общая схема преобразования исходного кода с языка ассемблера в процесс операционной системы.
5. Компоновщик: определение, назначение. Загрузочный модуль. Загрузчик: определение, назначение. Общая схема преобразования исходного кода в процесс операционной системы.
6. Кодировка: определение, назначение, примеры. Кодировка ASCII. Структура кодировки Windows-1251. Отличие ASCII и Windows-1251. Кодировка UNICODE: назначение, структура, UCS, UTF.
7. Кодировка: определение, назначение, примеры. Кодировка UNICODE: назначение, структура, UCS, UTF. Прямой (LE) и обратный (BE) порядок байт. BOM: определение, назначение, примеры.
8. Структура языка программирования: идентификаторы, зарезервированные идентификаторы, литералы, ключевые слова. Определения, примеры (C++).
9. Структура языка программирования: фундаментальные типы данных. Определение, примеры (C++). Строки. Массивы данных фундаментального типа. Преобразование типов.
10. Структура языка программирования: фундаментальные типы данных. Преобразование типов: автоматическое преобразование, явное преобразование.
11. Структура языка программирования: пользовательские типы данных. Определение, примеры (C++). Массивы данных пользовательского типа. Перегрузка операторов для пользовательских типов.
12. Структура языка программирования: область видимости переменных, пространства имен, псевдонимы пространства имен.

13. Структура языка программирования: инициализация переменных, присвоение значений, выражения (lvalue, rvalue, побочные эффекты, точка последовательности, унарные, бинарные и тернарные выражения), константные выражения.
14. Структура языка программирования: инструкции языка программирования, объявление, простые и составные инструкции, инструкции выбора, инструкции циклов, инструкции переходов, инструкции обработки исключений, примеры.
15. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Передача параметров по значению и по ссылке, передача переменного числа параметров в функции C++.
16. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Передача параметров по значению и по ссылке, передача значений параметров по умолчанию в функции C++.
17. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Реализация программных конструкций в C++. Лямбда-функции в C++.
18. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Реализация программных конструкций в C++. Перегружаемые функции в C++. Inline-функции в C++. Шаблоны функций C++.
19. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Реализация программных конструкций в C++. Соглашения о вызовах (`_cdecl`) в C++.
20. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Реализация программных конструкций в C++. Соглашения о вызовах (`_stdcall`) в C++.
21. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров. Реализация программных конструкций в C++. Соглашения о вызовах (`_fastcall`) в C++.
22. Структура языка программирования: стандартная библиотека. Реализация стандартной библиотеки STL в C++. Понятие контейнера, итератора и алгоритма. Стандартные функции STL для работы со строками: копирования, сравнения, вычисления длины, поиска символа и подстроки.
23. Структура языка программирования: стандартная библиотека. Реализация стандартной библиотеки STL в C++. Понятие контейнера, итератора и алгоритма. Стандартные функции STL для работы с широкими строками: копирования, сравнения, вычисления длины, поиска символа и подстроки.
24. Структура языка программирования: стандартная библиотека. Реализация стандартной библиотеки STL в C++. Понятие контейнера, итератора и алгоритма. Стандартные функции для работы с потоками данных. Стандартные функции для получения текущей даты, даты и времени.
25. Классы памяти: код, стек, статические данные, динамическая область. Структура динамической памяти (Heap) C++. Фрагментация динамической памяти C++.
26. Механизм обработки исключений: определение, назначение, применение. Реализация обработки исключений в C++.
27. Препроцессор: определение, назначение, применение, директивы, выражения, макросы. Примеры на C++.

28. Статическая библиотека: определение, назначение, применение. Реализация статической библиотеки в Windows C++. Утилита LIB. Компоновка статической библиотеки. Применение директивы препроцессора `#pragma comment` для компоновки статической библиотеки.
29. Теория формальных языков: алфавит языка, цепочка, пустая цепочка, длина цепочки, равенство (эквивалентность) цепочек, конкатенация цепочек, итерация цепочки, операторы $^+$ и * .
30. Теория формальных языков: определение формального языка, эквивалентность двух языков, способы задания формального языка. Лексика, синтаксис и семантика языка.
31. Теория формальных языков: грамматика (определение, назначение, примеры), выводимость цепочки символов в грамматике, сентенциальная форма грамматики, язык, порождаемый грамматикой, способы задания грамматик.
32. Теория формальных языков: грамматика (определение, назначение, примеры), способы задания грамматик. Общая характеристика формы Бэкуса-Наура.
33. Теория формальных языков: иерархия Хомского (неограниченные, контекстно-зависимые, контекстно-независимые и регулярные грамматики), соотношения грамматик и порождаемых ими языков.
34. Лексический анализ: определение, назначение, применение. Лексический анализатор: место в структуре транслятора, входная и выходная информация, способы взаимодействия с синтаксическим анализатором (последовательное и параллельное).
35. Лексический анализ: определение, назначение, применение. Лексический анализатор: нисходящий и восходящий разбор, дерево разбора.
36. Регулярные выражения: рекурсивное определение, назначение, применение, метасимволы, примеры. Применение в функциях стандартной библиотеке STL C++.
37. Конечный автомат: определение, назначение, схема работы, примеры. Недетерминированные и детерминированные конечные автоматы. Мгновенное описание конечного автомата. Последовательность мгновенных описаний, диаграмма мгновенных описаний.
38. Конечный автомат: определение, назначение, схема работы, примеры. Соотношение регулярного языка, регулярной грамматики, регулярного языка и конечного автомата.
39. Конечный автомат: построение графа переходов конечного автомата по регулярному выражению. Примеры построения графов переходов. Алгоритм разбора цепочки символов.
40. Конечный автомат: построение графа переходов конечного автомата по регулярному выражению. Примеры построения графов переходов. Алгоритм разбора цепочки символов. Реализация алгоритма разбора с помощью двух массивов.
41. Синтаксический анализ: определение, назначение, применение. Синтаксический анализатор: определение, входные данные, выполняемые задачи, результат работы. Соотношение между лексическим и синтаксическим анализом.
42. Приведение контекстно-свободной грамматики: удаление бесплодных символов, недостижимых символов, λ -правил, цепных правил.
43. Автомат с магазинной памятью: определение, схема работы, пример, конфигурация (текущее состояние автомата).
44. Нормальная форма Хомского контекстно-свободной грамматики. Алгоритм приведения контекстно-свободной грамматики к нормальной форме Хомского. Примеры.

45. Нормальная форма Грейбах контекстно-свободной грамматики. Алгоритм приведения контекстно-свободной грамматики к нормальной форме Грейбах. Примеры.
46. Язык ассемблера: структура программы, типы данных, директивы для определения данных, регистры общего назначения центрального процессора: назначение, названия. Примеры.
47. Язык ассемблера: типы данных, инициализация данных, синтаксис оператора определения данных, массивы и их инициализация. Примеры.
48. Язык ассемблера: формат команды ассемблера, команды пересылки данных, команды целочисленного сложения и вычитания, операторы OFFSET (понятие), PTR, TYPE, SIZEOF и LENGTHOF. Примеры.
49. Язык ассемблера: адресация – прямая и косвенная. Косвенная адресация: с косвенным операндом. Косвенная адресация: операнды с индексом. Примеры косвенной адресации.
50. Язык ассемблера: операции работы со стеком, команды переходов (безусловные, условные), команда LOOP. Назначение, использование. Примеры.

Лектор

Наркевич А.С.