

## УТВЕРЖДАЮ

Зав. кафедрой программной инженерии БГТУ

к.т.н., доцент \_\_\_\_\_ Н.В. Пацей

протокол № 9 от 06 мая 2022 г.

### Экзаменационные вопросы по дисциплине «Конструирование программного обеспечения», 1 курс ПОИТ, весенний семестр 2022

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

1. Современные подходы к разработке программного обеспечения (ПО): понятие конструирования ПО, основные этапы конструирования ПО.
2. Современные подходы к разработке программного обеспечения: понятие конструирования ПО; результат конструирования; решения, которые принимаются при конструировании.
3. Система программирования: определение, состав, назначение. Классический жизненный цикл разработки ПО. Текстовый редактор. Интегрированная среда разработки. Программный продукт. Общая схема преобразования исходного кода в процесс операционной системы.
4. Интегрированная среда разработки: определение, состав, назначение. Транслятор: виды трансляторов. Транслятор: определение, назначение, примеры. Интерпретатор: определение, назначение, примеры.
5. 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.
6. Компоновщик: определение, назначение. Загрузочный модуль. Загрузчик: определение, назначение. Общая схема преобразования исходного кода в процесс операционной системы.
7. Иерархическая структура компонентов в Visual C++: глобальный контейнер, компоненты, входящие в его состав. Структура проекта Visual C++.
8. Иерархическая структура компонентов в Visual C++: многофайловый проект, предварительно откомпилированные заголовки: назначение, создание, применения. Пример (C++).
9. Кодировка: определение, назначение, примеры. Кодировка ASCII. Структура кодировки Windows-1251. Отличие ASCII и Windows-1251. Символы времени выполнения: установка кодовой страницы языкового стандарта.
10. Кодировка: определение, назначение, примеры. Кодировка UNICODE: назначение, структура, UCS, UTF. Прямой (LE) и обратный (BE) порядок байт. BOM: определение, назначение, примеры.
11. Структура языка программирования: алфавит языка программирования, идентификаторы, зарезервированные идентификаторы, литералы, ключевые слова, определения. Правила составления идентификаторов в C++ и других языков программирования. Примеры.
12. Структура языка программирования: фундаментальные типы данных. Что определяет тип данных? Определения, примеры (C++ и другие языки).

13. Структура языка программирования: литералы, типы литералов, способы задания. Строки. Массивы данных фундаментального типа. Примеры (C++).
14. Структура языка программирования: фундаментальные типы данных. Типы данных для хранения вещественных значений. Стандарт IEEE 754. Что определяет стандарт IEEE 754.
15. Структура языка программирования: фундаментальные типы данных. Внутреннее представление величины целого типа, спецификаторы, диапазоны значений целого типа.
16. Структура языка программирования: система типов языка программирования, Вывод типов. Преобразование типов: автоматическое преобразование, явное преобразование. Определения и примеры (C++).
17. Структура языка программирования: система типов языка программирования, Вывод типов. Преобразование типов: расширяющее преобразование, сужающее преобразование. Назначение оператора sizeof языка программирования C++. Примеры.
18. Структура языка программирования: пользовательские типы данных (структуры, перечисления, объединения). Определение, примеры (C++). Массивы данных пользовательского типа. Перегрузка операторов для пользовательских типов.
19. Структура языка программирования: пользовательские типы данных. Определение структуры, примеры (C++). Перегрузка операторов для пользовательских типов.
20. Структура языка программирования: понятие конфликта имен, область видимости переменных, пространства имен, псевдонимы пространства имен.
21. Структура языка программирования: определение выражения, его состав, порядок вычисления выражения. Символ окончания последовательности. Примеры (C++).
22. Структура языка программирования: выражения (lvalue, rvalue, побочные эффекты, точка последовательности, унарные, бинарные и тернарные выражения), константные выражения, укороченное вычисление. Примеры (C++).
23. Структура языка программирования: инструкции языка программирования, объявление, определение, инициализация, простые и составные инструкции, инструкции выбора, инструкции циклов, инструкции переходов, примеры (C++).
24. Структура языка программирования: инструкции языка программирования, инструкции обработки исключений, примеры (C++).
25. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Функции: передача параметров по значению и по ссылке, передача переменного числа параметров в функции C++. Примеры.
26. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров в функцию. Передача параметров по значению и по ссылке, передача значений параметров по умолчанию в функции C++. Примеры.
27. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров в функцию. Реализация программных конструкций: лямбда-функции в C++. Примеры.
28. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров в функцию. Реализация программных конструкций в C++. Перегружаемые функции в C++. Inline-функции в C++. Шаблоны функций C++.
29. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров в функцию. Реализация программных конструкций в C++: соглашения о вызовах (\_cdecl).

30. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров в функцию. Реализация программных конструкций в C++: соглашения о вызовах (`_stdcall`).
31. Структура языка программирования: программные конструкции (блоки, функции, процедуры и пр.). Передача параметров в функцию. Реализация программных конструкций в C++: соглашения о вызовах (`_fastcall`).
32. Структура языка программирования: стандартная библиотека. Реализация стандартной библиотеки STL в C++. Понятие контейнера, итератора и алгоритма. Стандартные функции STL для работы со строками: копирование, сравнение, вычисление длины, поиск символа и подстроки.
33. Классы памяти: код, стек, статические данные, динамическая область памяти. Структура динамической памяти (Heap) C++. Фрагментация динамической памяти C++.
34. Механизм обработки исключений: определение, назначение, применение. Реализация обработки исключений в C++. Пример.
35. Препроцессор: определение, назначение, применение, директивы, выражения, макросы, директивы условной компиляции. Примеры на C++.
36. Статическая библиотека: определение, назначение, применение. Реализация статической библиотеки в Visual C++. Утилита LIB. Создание статической библиотеки. Применение директивы препроцессора `#pragma comment` для компоновки со статической библиотеки.
37. Теория формальных языков. Определения: алфавит языка, цепочка, пустая цепочка, длина цепочки, равенство (эквивалентность) цепочек, конкатенация цепочек, итерация цепочки, операторы  $^+$  и  $^*$ . Примеры.
38. Теория формальных языков. Определение формального языка, эквивалентность двух языков, способы задания формального языка. Лексика, синтаксис и семантика языка. Примеры.
39. Теория формальных языков. Формальная грамматика (определение, назначение), выводимость цепочки символов в грамматике, сентенциальная форма грамматики, язык, порождаемый грамматикой, способы задания грамматик. Примеры.
40. Теория формальных языков. Определение порождающей грамматики. Определение вывода, левосторонний и правосторонний выводы, дерево вывода. Примеры.
41. Теория формальных языков: способы задания грамматик. Общая характеристика формы Бэкуса-Наура. Расширенная БНФ. Примеры.
42. Классификация языков и грамматик по Хомскому: иерархия Хомского (неограниченные, контекстно-зависимые, контекстно-независимые, регулярные грамматики и вид правил этих грамматик), соотношения грамматик и порождаемых ими языков.
43. Лексический анализ: определение, назначение, применение. Лексический анализатор: место в структуре транслятора, входная и выходная информация, способы взаимодействия с синтаксическим анализатором (последовательное и параллельное).
44. Лексический анализ: определение, назначение, применение. Лексический анализатор: вид правил регулярной грамматики, нисходящий и восходящий разбор цепочек, дерево разбора.
45. Регулярные выражения: рекурсивное определение, назначение, применение, метасимволы, примеры. Пример применения в функциях стандартной библиотеки STL C++.
46. Конечный автомат (КА): определение, назначение, схема работы КА, примеры.

47. Конечный автомат (КА): недетерминированные и детерминированные КА. Мгновенное описание КА. Последовательность мгновенных описаний, диаграмма мгновенных описаний. Пример.
48. Конечный автомат: определение, схема работы, примеры. Соотношение регулярного языка, регулярной грамматики, регулярного языка и конечного автомата.
49. Конечный автомат: определение графа переходов конечного автомата и метод его построение по регулярному выражению. Примеры построения графов переходов.
50. Конечный автомат: алгоритм разбора цепочки символов, основанный на двух массивах.

Лектор

Наркевич А.С.