

## УТВЕРЖДАЮ

Зав. кафедры программной инженерии БГТУ

к.т.н., доцент \_\_\_\_\_ В.В. Смелов

протокол № 4 от 28.11.2024 г.

### Экзаменационные вопросы

#### дисциплина «Основы программной инженерии»

6-05-0612-01 «Программная инженерия»

профилизация: программная обеспечение информационных технологий

1. Цели и задачи технологий разработки ПО. Особенности современных проектов разработки ПО.
2. Основные определения: программа, программный продукт, программное обеспечение, программная инженерия, жизненный цикл ПО. Стандарты программной инженерии.
3. Классификация программного обеспечения. Определение и состав системы программирования. Примеры.
4. Интегрированная среда разработки Visual Studio. Представление символьной информации в кодировке Windows-1251. Примеры.
5. Компоненты классической системы программирования. Трансляторы, ассемблеры, интерпретаторы. Схема работы транслятора.
6. Структура классической системы программирования. Этапы обработки исходного кода программы. отладчик. Основные элементы языка программирования. Алфавит языка программирования, символы времени трансляции, символы времени выполнения.
7. Структура системы программирования. Модели памяти (классы памяти): код, статическая память, стек, динамическая память.
8. Парадигмы программирования. Примеры.
9. Язык программирования: определение, назначение, примеры. Исходный код. Объектный код. Объектный модуль. Загрузочный модуль.
10. Кодирование информации: определение, назначение, данные, представление данных, кодировки. Примеры.
11. Спецификация системы программирования. Кодировка ASCII, кодировка Windows-1251, стандарт кодирования символов Unicode.
12. Кодирование информации: определение, назначение, примеры. Стандарт кодирования UNICODE: назначение, структура, UCS, UTF. Прямой (LE) и обратный (BE) порядок байт. BOM: определение, назначение, примеры.
13. Этапы и цели разработки программы, трудоемкость этапов разработки программ.
14. Алгоритм программы. Назначение и свойства алгоритмов. Способы описания алгоритмов.
15. Системы программирования Microsoft, Linux, Unix, IBM. Стандарты языков программирования. Парадигмы (стили) программирования.
16. Интегрированная среда разработки: определение. Примеры IDE. Назначение, основные возможности. Понятие отладки кода на языке программирования.
17. Интегрированная среда разработки MS Visual Studio 20XX. Назначение, возможности. Страница свойств проекта.
18. Среда разработки: назначение и основные возможности отладчика. Точки останова. Отображение и модификация локальных данных. Пошаговая отладка. Понятие и назначение дизассемблера.
19. Методология разработки программного обеспечения. Структурный подход к проектированию ПО. Сущность структурного подхода. Методы структурного программирования

20. Методология разработки программного обеспечения. Модульное программирование. Сущность модульного подхода. Методы модульного программирования
21. Системы контроля версий. Классификация. Назначение, разновидности систем контроля версий.
22. Три состояния файлов в Git. Сохранение изменений в репозитории. Основные определения и основные команды Git.
23. Ветвления в Git. Слияния веток. Конфликты при слиянии веток. Ветвление проектов. Распределенная разработка.
24. Понятие веб-сервиса Github. Отличие Git и GitHub. Назначение и основные возможности GitHub. Совместная работа над проектом.
25. Этапы создания программного продукта. Понятие жизненного цикла разработки программного обеспечения. Назначение модели жизненного цикла ПО. Структура процессов жизненного цикла программного обеспечения.
26. Каскадная модель жизненного цикла ПС: содержание этапов, область применения, достоинства и недостатки.
27. Эволюционная модель жизненного цикла ПС: последовательность действий, область применения, достоинства и недостатки.
28. Спиральная модель разработки ПО: содержание этапов создания ПС, область применения, достоинства и недостатки.
29. Инкрементальная модель разработки ПО. Развитие инкрементального подхода. XP-процессы.
30. Методологии разработки программного обеспечения. Инструментарий технологии программирования. Управление требованиями. Техническое задание на разработку программного продукта.
31. Методологии быстрой разработки ПО. Жизненный цикл ПО по методологии RAD. Преимущества, недостатки, область применения.
32. Инкрементальная модель разработки ПО. Итерационная модель разработки ПО. Отличие итерационной модели от инкрементной модели.
33. Методологии гибкой разработки ПО. Различия между Agile и традиционным подходом к разработке ПО. Преимущества и недостатки технологий быстрой разработки программного обеспечения.
34. Модель компетентного разработчика (Personal Software Process).
35. Этапы конструирования. Подходы к конструированию программных средств.
36. Технологии разработки ПО. Управление требованиями. Понятие требования к ПО. Виды и уровни требований, классификация требований.
37. Формализация функциональных требований: диаграммы вариантов использования. Назначение и компоненты диаграмм вариантов использования. Примеры.
38. Тестирование ПО: основные понятия и определения. Классификация видов тестирования. Цели, задачи и принципы тестирования.
39. Цикл тестирования ПО. Тестирование по позитивному и негативному признаку.
40. Тестирование ПО: методы тестирования. Ручное тестирование.
41. Разработка программной документации. Назначение документирования программного обеспечения. Стандарты документирования.
42. Управление командой проекта. Ролевая модель команды. Функции каждого из участников.
43. Международные организации стандартизации. Объекты стандартизации.
44. Принципы Agile. Манифест Agile. Управление проектами.
45. Схема процесса работы по методологии Scrum. Планирование и длительность спринта в Scrum. Бэклог продукта и пользовательские истории. Основные Scrum встречи.
46. Разработка проекта по методологии Kanban. Назначение доски Kanban. Отличие методологии Scrum от Kanban.

- 47. Язык программирования: определение, назначение, примеры. Схема создания исполняемого файла для компилируемых языков программирования. Стилль оформления кода.
- 48. Кодирование информации. Кодировка UNICODE: назначение, структура. Маркер BOM: определение, назначение, примеры.
- 49. Система контроля версий Git: основные возможности. Отличия Git от других систем контроля версий.
- 50. Технологии разработки ПО. Управление требованиями. Функциональные требования и нефункциональные требования. Разработка требований,

Лектор

Наркевич А.С.