**СПИСОК ВОПРОСОВ**

«Программная инженерия»

для студентов очной формы обучения

1. ПОНЯТИЕ ПРОГРАММНОЙ ИНЖЕНЕРИИ. История. Модульное программирование.
2. Рост сложности программ. Статистика успешности. Причин неудач.
3. Структурное программирование. Основные принципы.
4. Процедурно функциональное программирование. Основные принципы.
5. Объектно ориентированное программирование. Основные принципы.
6. Требования к ПО (функциональные, нефункциональные)
7. Кризис программирования
8. Программная инженерия и программный продукт
9. Отличие информатики от программной инженерии
10. Стоимость ПО. Распределение стоимости по основным этапам.
11. Коробочное и индивидуальное ПО
12. Разные виды ПО и подходы программной инженерии к ним (web, системное ПО, прикладное ПО, приложения)
13. ОС (webOS, Windows, Linux, MacOS и др). Сравнение и анализ применительно к ПО.
14. Машинный код. Языки программирования и среды разработки IDE. История, развитие, перспективы. Низкий уровень и высокий, уровни абстракций.
15. Базы данных.
16. Синхронная и асинхронная программные модели.
17. Однопоточность и многопоточность. Параллелизм
18. библиотека, API, фрэймворк. Разница
19. Фронтенд, бэкенд и фулстэк.
20. Техники программирования (парное, пинг понг, удаленное и тд)
21. Принципы для разработки: KISS, DRY, YAGNI, BDUF, SOLID, APO
22. Шаблон проектирования (паттерны) в разработке программного обеспечения
23. архитектура ПО ()
24. Виды архитектур приложений.
25. Методы программной инженерии. Компоненты метода. Case технологии и Case средства.
26. Понятие программного обеспечения и программного продукта
27. Профессиональные и этические требования к специалистам.
28. Стандарты программной инженерии (корпоративные, отраслевые, государственные, международные.)
29. Организации международной стандартизации
30. Жизненный цикл программного продукта. История. Проблемы.
31. Основные и вспомогательные процессы жизненного цикла.
32. Языки Программирования. Среды разработки.
33. Каскадная (водопадная) модель жизненного цикла ПО
34. Спиральная модель жизненного цикла ПО
35. V-образная модель жизненного цикла ПО
36. Инкрементальная модель жизненного цикла ПО.
37. Экстремальное программирование модель жизненного цикла ПО.
38. гибкие методологии разработки по agile, scrum, kanban
39. GitHUB. Основные понятия и возможности. Аналоги.
40. Автоматизация написания кода, рутинные задачи.
41. Управление проектами. Основные понятия, характеристики и ограничения.
42. Ролевая модель команды. Основные роли, функции и компетенции. Ролевые группы
43. Градация разработчика: преджуниор, джуниор, джуниор+, мидл, сеньор.
44. Модели организации команды (административная модель, модель хаоса, открытая архитектура)
45. Коммуникации в команде как основной фактор разработки ПО
46. Планирование и управление рисками. Примеры рисков.
47. Управление качеством проекта. Качество ПО и качество процесса. ISO9000
48. Техническая документация. Техническое задание и технический проект. Автодокументация и другие средства.
49. Тестирование ПО. Цели. Классификации видов и методов тестирования. Уровни тестирования.
50. Верификация и валидация
51. Отладка и дебаггинг. Логирование и профилирование
52. Юнит тесты и пирамида тестирования, программные средства для тестирования
53. Фаза интеграционного тестирования
54. Модели белого, черного и серого ящиков в тестировании. Понятие покрытие кода.
55. Статическое и динамическое тестирование.
56. Рефакторинг. Понятие, цели и причины. Понятие плохого кода и технического долга.
57. Реинжиниринг. Понятие, цели и причины. Сложности реинжиниринга.
58. Профилирование. Функциональные и аппаратные характеристики. Анализа покрытия.
59. инструмент управления проектами Trello/jira
60. Docker, Kubernetes (образы, контейнеры, виртуализация и тд)
61. Управление командой разработки ПО. Показатели контроля качества: план повышения производительности PIP, итоги Performance Review, SMP, UEIP и тд
62. Свободный вопрос по курсу.