

Учебная практика профессионального модуля

ПМ 04 Сопровождение и обслуживание программного обеспечения компьютерных систем

Задания для подготовки к квалификационному экзамену ПМ04

Часть №1

1. Решить тест.
2. Гугл Формы. В сервисе Google Forms Оформить тест по 15 вопросов в соответствии с таблицей.
3. Установить разрешение редактирования формы для аккаунта ifizmat@gmail.com
4. Яндекс Формы. В сервисе Яндекс Формы Оформить тест по 15 вопросов в соответствии с таблицей.
5. Расчитать на отдельной странице электронной таблицы оценку за прохождение теста. Пример формулы расчета оценки теста

```
=IF(test2!C2="Это оператор присваивания, он помещает значение, расположенное справа от него, в переменную, стоящую слева", 1, 0)
+IF(test2!D2="Это ключевое слово для определения типа данных как целое число", 1, 0)
+IF(test2!E2="транзистор", 1, 0)
+IF(test2!F2="Потенциометр можно рассматривать как два резистора с переменным сопротивлением и использовать для регулировки напряжения", 1, 0)
+IF(test2!G2="Для включения на входе встроенного подтягивающего к напряжению питания резистора", 1, 0)
+IF(AND(test2!H2="Организация программного последовательного интерфейса передачи данных", test2!I2="Управление сервомотором", test2!J2="Управление шаговым двигателем", test2!K2="Работа с сетевым интерфейсом"), 1, 0)
+IF(AND(test2!L2="Передатчик аппаратного последовательного интерфейса", test2!M2="Приемник аппаратного последовательного интерфейса", test2!N2="Подача входного питания платы Arduino", test2!O2="Первый аналоговый вход аппаратного АЦП"), 1, 0)
```

6. Документ Гугл Таблиц с оценками сохранить в папке ПМ04DevOps/test/.

Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Тест 6	Тест 7
1	2	3	4	5	6	7
7	8	9	10	11	12	13
13	14	15	16	17	18	19
19	20	21	22	23	24	25
25	26	27	28	29	30	31
31	32	33	34	35	36	37
37	38	39	40	41	42	91
43	44	45	46	47	48	92
49	50	51	52	53	54	93
55	56	57	58	59	60	94
61	62	63	64	65	66	95
67	68	69	70	71	72	96
73	74	75	76	77	78	97
79	80	81	82	83	84	98
85	86	87	88	89	90	99

Тема: Основные методы внедрения и анализа функционирования программного обеспечения

Контрольные вопросы для проведения тестирования

1. Программная инженерия:

- software engineering
- Инструменты создания программного обеспечения
- Коллектив инженеров-программистов, разрабатывающих программное обеспечение для компьютеров
- Дисциплина, изучающая применение строгого систематического количественного подхода к разработке, эксплуатации и сопровождению программного обеспечения
- Комплекс программ, предназначенный для решения инженерных задач, связанных с большим количеством расчетов
- Инженерная индустрия применения прикладного программного обеспечения
- Совокупность инженерных методов и средств создания программного обеспечения
- Прикладное программное обеспечение для решения офисных задач

2. Построение SADT-модели включает в себя выполнение следующих действий:

- Написание программного обеспечения для разрабатываемой системы по требованиям заказчика
 - Сбор информации об объекте, определение его границ
 - Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм
 - Представление исследуемой системы в графическом виде
 - Представление исследуемого объекта средствами системного моделирования
 - Критическая оценка, рецензирование и комментирование
 - Разработка, отладка и тестирование программного обеспечения
 - Использование графических пакетов для представления системы в виде модели
- 3. Моделирование основывается на принципах:
 - Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение

- Декомпозиции системы на отдельные подзадачи
 - Инкапсуляции и полиморфизма
 - Децентрализации управления системой
 - Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности
 - Открытой трансформируемой системы
 - Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей,
- почти независимых друг от друга
- Анализа и синтеза проектирования систем
4. В бизнес-процессах выделяют классы процессов:
- Решающие бизнес-процессы
 - Регламентирующие бизнес-процессы
 - Основные бизнес-процессы
 - Бизнес-процессы поведения системы¹³
 - Программируемые бизнес-процессы
 - Экономические бизнес-процессы
 - Обеспечивающие бизнес-процессы
 - Бизнес-процессы управления
5. CASE-средства классифицируются по следующим признакам:
- По применяемым методологиям и моделям систем и БД
 - По используемому программному обеспечению
 - По этапам жизненного цикла программного обеспечения
 - По степени интегрированности с СУБД
 - По уровням детализации и декомпозиции проектируемой системы
 - По доступным платформам
 - По используемым языкам программирования
 - По степени сложности моделируемой системы
6. К малым интегрированным средствам моделирования относятся:
- ARIS Toolset
 - Design/IDEF
 - ERwin
 - BPwin
 - Designer/2000
 - Paradigm Plus
 - Model Mart
 - Rational Rose
7. К средним интегрированным средствам моделирования относятся:
- Rational Rose
 - Design/IDEF
 - BPwin
 - Designer/2000
 - ARIS Toolset
 - Model Mart
 - Paradigm Plus
 - ERwin
8. Объектно-ориентированная методология (ООМ) включает в себя составные части:
- Объектно-ориентированный анализ
 - Объектно-ориентированный подкласс
 - Объектно-ориентированное проектирование
 - Объектно-ориентированная парадигма
 - Объектно-ориентированная экспозиция
 - Объектно-ориентированное моделирование
 - Объектно-ориентированное программирование
 - Объектно-ориентированная декомпозиция
9. К основным понятиям объектно-ориентированного подхода относятся:
- Обобщение
 - Полиморфизм
 - Инкапсуляция
 - Реализация
 - Агрегирование
 - Наследование¹⁴

- Ассоциация
- Композиция

10. Главные принципы объектного подхода:

- Абстрагирование
- Наследование
- Ограничение доступа или инкапсуляция
- Безграничный доступ или инкапсуляция
- Модульность и иерархия
- Агрегирование
- Композиция
- Обобщение и специализация

11. Дополнительные принципы объектного подхода:

- Реализация
- Типизация
- Параллелизм
- Внедрение
- Перпендикулярность
- Сохраняемость или устойчивость
- Несохраняемость или неустойчивость
- Динамичность

12. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся:

- Rational Rose
- Model Mart
- MS Visio
- ARIS
- IDEF1X
- Erwin
- BPwin
- JAM

13. К инструментальным средствам представления функциональных моделей относятся:

- JAM
- Model Mart
- MS Visio
- ARIS
- IDEF0
- Erwin
- BPwin
- Rational Rose

14. Методологии, поддерживаемые в BPwin:

- IDEF1X
- IDEF0
- IDEF1
- IDEF3
- IDEFX
- IDEF5
- DFD15
- DFD1X

15. Диаграмма IDEF0 может содержать следующие типы диаграмм:

- Диаграмму классов
- Контекстную диаграмму, диаграмму декомпозиции
- Диаграмму компонентов
- Диаграмму дерева узлов
- Диаграмму взаимодействий
- Диаграмму только для экспозиции (FEO)
- Диаграмму последовательности, диаграмму кооперации
- Диаграмму узлов

16. Уровни логической модели:

- Диаграмма сущность
- Диаграмма связь
- Диаграмма пакетов
- Диаграмма сущность-связь
- Модель данных, основанная на классах

- Модель данных, основанная на ключах
- Полная операционная модель
- Полная атрибутивная модель

17. Внутренние стрелки не входящие в состав диаграммы IDEF0:

- mechanism- output
- output-input
- mechanism- input
- output-control
- output-input feedback
- output-control feedback
- output-mechanism
- control feedback- mechanism

18. Типы стрелок не входящие в состав диаграммы IDEF0:

- Input
- Editor
- Control
- Properties
- Output
- Mechanism
- Call
- Dictionary

19. Quick Reports – создание простейших отчетов – позволяет создавать отчеты:

- Group/Totals. Табличный отчет с автоматической группировкой и сортировкой данных
- Report Header. Печатается единожды в начале отчета
- Columnar. Простой табличный отчет
- Page Header. Печатается в верхней части каждой страницы
- Vertical. Простой вертикальный отчет
- Group Header. Печатается в начале каждой группы
- Blank Report. Бланк. Создается пустой бланк отчета, в который не включаются данные
- Detail. Печатается для каждой строчки набора данных

20. BPwin допускает следующие переходы с одной нотации на другую:

- IDEF3 → DFD
- DFD → IDEF0
- IDEF0 → DFD
- DFD → DFD
- IDEF3 → IDEF0
- IDEF0 → IDEF3
- IDEF3 → IDEF3
- DFD → IDEF3

21. DFD описывает:

- Функции обработки стрелок (arrow)
- Функции обработки информации (работы)
- Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации
- Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации
- Функции обработки внешних ссылок
- Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data stor- E)
- Функции обработки документов
- Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке внешних стрелок

22. BPwin позволяет создавать на диаграмме DFD типы граничных стрелок:

- Обычная граничная стрелка
- Специальная стрелка
- Внутренняя ссылка
- Межстраничная ссылка и тоннельная стрелка
- Внешняя ссылка
- Страницная ссылка и теневая стрелка

- Контрольная стрелка
- Стрелка механизм

23. Создать отчет в BPwin возможно с помощью:

- Встроенных шаблонов
- Программных модулей, создаваемых разработчиком на языке Visual Basic
- Создать отчет в BPwin не возможно
- Report Template Builder
- Отчет создается разработчиком
- Отдельно поставляемых программ
- Встроенных мастер-функций
- RPTwin

24. В BPwin 4.0 отчеты могут быть экспортаны в распространенные форматы:

- Текстовый
- Символьный
- MS Office
- Графический
- HTML
- Internet Explorer
- Acrobat17
- IBM Rational

25. Поддерживаемые в RPTwin типы операторов:

- Текстовый оператор конкатенации (&)
- Символ
- Текст
- Дата
- Арифметические
- Графический оператор конкатенации (&)
- Логические
- Номер

26. Инstrumentальное средство ERwin позволяет:

- Редактировать и отлаживать программы
- Проектировать на физическом и логическом уровне модели данных
- Управлять процессом конструирования ПО
- Проектировать диаграммы вариантов использования и взаимодействий
- Проводить процессы прямого и обратного проектирования баз данных
- Управлять процессом трансляции и отладки программ
- Выравнивать модель и содержимое системного каталога после редактирования
- Проектировать контекстные диаграммы и диаграммы декомпозиции

27. ERwin позволяет создавать модели следующих типов:

- Модель, имеющую только логический уровень
- Модель, имеющую абстрактный уровень
- Модель, имеющую абстрактный и физический уровни
- Модель, имеющую только физический уровень
- Модель, имеющую абстрактный и логический уровни
- Модель, имеющую как логический уровень, так и физический уровень
- Модель, имеющую концептуальный уровень
- Модель, имеющую контекстный уровень

28. Для создания моделей ERwin используют международно признанные системы обозначений (нотации):

- IDEF0
- IDEF1X
- IDEF3
- DFD
- IE
- DM
- IDEFDFD
- IDEF3

29. К основным компонентам диаграммы ERwin относятся:

- Сущности
- Переходы
- Атрибуты
- Классы
- Слияния
- Разветвления

- Использования

- Связи¹⁸

30. Точки зрения организации в ARIS:

- Структура внедрения и структура потоков
- Организационная структура
- Управленческая структура
- Поведенческая структура
- Функциональная структура
- Коммуникационная структура
- Структура данных и структура процессов
- Обобщенная структура

31. Уровни точки зрения в ARIS:

- Описание структуры
- Описание требований
- Описание поведения
- Описание разработки
- Описание спецификации
- Описание внедрения
- Описание процессов
- Описание классов

32. Методы описания, используемые в ARIS:

- EPT – метод описания потоков
- EPC – метод описания процессов
- ERM – модель сущность-связь для описания структуры объектов
- ERM – модель сущность-связь для описания структуры данных
- EPP – метод описания пакетов
- EPC – метод описания компонентов
- UML – унифицированный язык моделирования
- EPT – метод описания нитей

33. К основным компонентам инструментов ARIS Toolset относятся:

- Internet (интернет)
- WordPad (ввод текстовых данных)
- Media (средство для медиа описания моделей)
- Explorer (проводник)
- Acrobat (чтение текстовых данных)
- Designer (средство для графического описания моделей)
- Document (для ввода различных параметров и атрибутов) и выноски
- Таблица (для ввода различных параметров и атрибутов) и мастер (Wizards)

34. ARIS Business Optimizer позволяет:

- Определять целевые затраты и рассчитывать стоимость продукта: во что компании обходится предоставление отдельных продуктов
- Принимать решения о времени начала и окончания работы над проектом
- Принимать решения по аутсорсингу: стоит ли поручить выполнение бизнес-процессов внешнему поставщику услуг
- Определять последовательность работ, выполняемых в ходе работы над проектом
- Определять требования к персоналу компании, которая в дальнейшем будет эксплуатировать программное обеспечение
- Рассчитывать заработную плату сотрудников компании после внедрения программного обеспечения¹⁹
- Планировать требования к обслуживающему персоналу, сопровождающему программное обеспечение
- Планировать требования к персоналу: сколько необходимо сотрудников для оптимального выполнения работ

35. «Взгляды» ARIS:

- Процессы
- Потоки
- Функции (с целями)
- Данные и организация
- Процедуры
- Управление и внедрение
- Нити

- Память

36. Уровни анализа ARIS для каждого «взгляда»:

- Поведение
- Требования
- Спецификации
- Функции
- Процедуры
- Проверка
- Внедрение
- Тестирование

37. MS Visio позволяет создавать схемы, чертежи, диаграммы с помощью:

- Встроенных шаблонов
- Панели инструментов
- Трафаретов
- Графических редакторов
- Дополнительного программного обеспечения
- Панели рисования
- Стандартных модулей
- Панели автофигур

38. Язык UML – это:

- Язык программирования высокого уровня
- Унифицированный язык моделирования
- Язык для разработки систем искусственного интеллекта
- Unified Modeling Language
- Язык управления базами данных
- Язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем
- Язык создания запросов в базах данных
- Язык программирования низкого уровня

39. Моделирование в UML позволяет решать задачи:

- Анализа и синтеза систем управления
- Разработать и отладить программное обеспечение
- Визуализировать систему в ее текущем или желательном для нас состоянии
- Провести тестирование разработанного программного обеспечения²⁰
- Описать структуру или поведение системы; получить шаблон, позволяющий сконструировать систему
- Смоделировать разрабатываемую информационную систему
- Документировать принимаемые решения, используя полученные модели
- Рассчитать экономическую эффективность от внедрения программного обеспечения

40. Словарь UML включает строительные блоки:

- Зависимости
- Сущности
- Слияния
- Разветвления
- Связи
- Группировки
- Диаграммы
- Декомпозиции

41. UML, как язык документирования, помимо исполняемого кода производит и другие

продукты, включающие:

- Требования, архитектуру, проектные решения
- Спецификацию технических средств
- Дизайн, исходный код, проектные планы,
- Требования к уровню квалификации разработчиков
- Набор заданий для тестирования программного обеспечения
- Требования к уровню квалификации персонала сопровождения
- Тесты, прототипы, релизы (версии)
- Требования к выбору языка программирования

42. UML включает синтаксические и семантические правила для:

- Агрегации
- Тестирования
- Имен, областей действия

- Сборки
- Сопровождения
- Видимости, целостности
- Вывода из эксплуатации
- Исполнения

43. Применение языка UML существенно упрощает последовательное использование механизмов:

- Спецификации, дополнения
- Принятые разделения
- Выработки требований
- Создания плана работ
- Механизмы расширения
- Тестирования программного обеспечения
- Конструирования ПО
- Сопровождения ПО

44. Механизмы расширения UML включают:

- Исключения
- Стереотипы
- Дополнения
- Управления
- Помеченные значения
- Слияния
- Ограничения
- Объединения

45. Язык UML предназначен для:

- Визуализации
- Тестирования
- Сопровождения
- Специфирования
- Снятия с эксплуатации
- Конструирования, документирования
- Анализа требований
- Обучения персонала

46. В объектно-ориентированном моделировании между классами существуют типы связей:

- Слияние
- Линейность
- Зависимость
- Разветвление
- Цикличность
- Обобщение
- Ассоциация
- Агрегация

47. В состав графического представления класса в языке UML входят части:

- Отношения
- Имя
- Связи
- Атрибуты
- Описание
- Сущности
- Операции
- Механизмы

48. Программное обеспечение делится на классы:

- Системное ПО и прикладное ПО
- Системное ПО, прикладное ПО и инструментальные средства разработки программ
- Операционные системы, прикладное ПО, утилиты и драйверы
- Прикладное ПО и инструментальные средства разработки программ
- Системное ПО и инструментальные средства разработки программ
- Системное ПО, прикладное ПО и системы программирования
- Операционные оболочки, операционные системы, офисные программы
- Системное ПО, прикладное ПО и инструментальное ПО

49. Инструментальные средства разработки программ - это:

- Средства создания новых программ
- Сервисные средства разработки ПО

- Аналитические средства разработки ПО
 - Программное обеспечение, предназначенное для разработки и отладки новых программ²²
 - Средства отладки ПО
 - Средства тестирования ПО
 - Аппаратные и программные инструменты разработки нового ПО
 - Технические инструментальные средства разработки ПО
50. Аппаратные инструментальные средства разработки ПО – это:
- Система для разработки новых программ на конкретном языке программирования
 - Средства создания и редактирования текстов программ
 - Микропроцессор и подключаемые (внешние) устройства
 - Устройства вычислительной системы, специально предназначенные для поддержки разработки ПО
 - Периферийные устройства, микропроцессор вычислительного комплекса, предназначенные для разработки нового ПО
 - Программное обеспечение, написанное на языках программирования низкого уровня
 - Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ
 - Программы, используемые для корректировки и тестирования других прикладных или системных программ
51. Программные инструментальные средства разработки ПО – это:
- Программы, позволяющие выполнить все работы, определенные методологией проектирования ПО
 - Системное программное обеспечение, позволяющее сопровождать офисные программные пакеты
 - Средства создания текстовых документов
 - Программное обеспечение, используемое на всех стадиях разработки нового ПО
 - Программное обеспечение для настройки офисных приложений на условия конкретного применения
 - Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ
 - Устройство компьютера, специально предназначенное для поддержки разработки программных средств
 - Средства создания и редактирования текстовых документов
52. Транслятор – это:
- Программа, выполняющая перевод программы с одного языка программирования на другой
 - Комплекс программ мультимедийных технологий
 - Программа, которая выполняет перевод программы с одного языка программирования на машинные коды
 - Программа-переводчик с одного иностранного языка на другой
 - Техническое устройство передачи и преобразования аудио и видеосигналов
 - Техническое устройство для кодирования и декодирования информации
 - Программное обеспечение для обеспечения защиты информации на компьютере
 - Одно из основных средств автоматизации программирования для преобразования программы, написанный на машинно-независимом языке, в программу на машинном языке конкретной ЭВМ
53. Компилятор – это:
- Один из видов трансляторов
 - Прикладное программное обеспечение
 - Специальная утилита системного ПО²³
 - Операционная оболочка
 - Переводит в коды сразу всю программу и создает независимый исполняемый файл
 - Программное обеспечение, используемое в издательских системах
 - Программа, которая переводит программу, написанную на языке программирования высокого уровня в программу на машинном языке не участвуя в ее исполнении
 - Переводит в машинные коды 1 строчку программы и сразу ее выполняет

54. Интерпретатор:

- Программа для создания и редактирования электронных таблиц
- Программа, анализирующая команды или операторы исходной программы и немедленно выполняющая их
- Переводит в коды сразу всю программу и создает независимый исполняемый файл
- Переводит в машинные коды 1 строчку программы и сразу ее выполняет
- Программа для создания и редактирования текстовых документов
- Один из видов трансляторов
- Программа создания и управления базами данных
- Программа создания файлов мультимедиа

55. Компоновщик – это:

- Программа для компоновки и оформления тестовых документов
- Редактор связей
- Комплекс программ, для создания и ведения баз данных
- Программа, которая из одного или нескольких объектных модулей с привлечением библиотечных программ и стандартных подпрограмм формирует загрузочный модуль
- Программное обеспечение для создания презентаций
- Программа сборки загрузочного модуля из полученных в результате раздельной компиляции объектных модулей с автоматическим поиском и присоединением библиотечных подпрограмм и процедур
- Программа для поиска синтаксических и семантических ошибок в программе
- Программа

56. Отладчик:

- Программа, облегчающая программисту выполнение отладки разрабатываемых им программ
- Программа для создания системы защиты файла
- Программа создания системы защиты от вирусных атак
- Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку
- Операционная оболочка для создания и управления файловыми структурами
- Системное программное обеспечение для настройки операционной системы
- Программа создания и редактирования графических файлов
- Программа, позволяющая выполнять остановы в заданных точках, просмотреть текущие значения переменных и изменять их значения

57. К этапам развития технологии разработки программного обеспечения относятся:

- «Процедурное» программирование
- Программирование на алгоритмических языках высокого уровня
- Структурный подход к программированию
- Программирование на языках низкого уровня
- Компонентный подход и CASE-технологии
- Машинно-ориентированное программирование
- Машинно-независимое программирование²⁴
- Подход к разработке ПО, основанный на стратегии поиска

58. «Стихийное» программирование:

- Разработка программного обеспечения без предварительного составления план-графики работ
- Первый этап в истории развития технологии разработки программного обеспечения, когда программирование фактически было искусством
- Период в истории разработки программного обеспечения, когда программа создавалась одним программистом, способным отслеживать последовательность выполняемых операций и местонахождения данных в программе
- Разработка программ с использованием различных языков программирования низкого и высокого уровня
- Разработка программ с элементами случайного выбора алгоритмов решения задачи
- Характеризуется тем, что типичная программа этого периода состояла из основной программы, области глобальных данных и набора подпрограмм (в основном библиотечных), выполняющих обработку всех данных или их части

- Разработка программного обеспечения для решения задач теории вероятностей и математической статистики
 - Разработка программного обеспечения для решения задач, построенных на алгоритмах случайного поиска
59. Структурный подход к программированию – это:
- Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
 - Создание программного обеспечения на основе структурной схемы решаемой задачи
 - Подход, требующий разработки структурной схемы алгоритма и программы решения задачи
 - Подход, в основе которого лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40–50 операторов) подпрограмм
 - Подход к решению задачи, требующий создание структурной схемы этапов работ по разработке программного обеспечения
 - Процесс создания программного обеспечения на основе структурной схемы исследуемого объекта или процесса
 - Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
 - Подход, требующий представления задачи в виде иерархии подзадач простейшей структуры
60. Объектный подход к программированию – это:
- Технология создания сложного программного обеспечения, основанная на представлении задачи исследования как объекта
 - Технология создания сложного программного обеспечения, предназначенного для автоматизации технологических объектов
 - Технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств
 - Технология создания сложного программного обеспечения, основанная на представлении программы как единого объекта
 - Технология создания сложного программного обеспечения, позволяющая вести практическую независимую разработку отдельных частей (объектов) программы
 - Технология создания сложного программного обеспечения, основанная на объектном представлении кода программы²⁵
 - Технология создания сложного программного обеспечения, в основе которой лежат новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции, наполнения
 - Технология создания сложного программного обеспечения, основанная на объектноориентированном программировании
61. Компонентный подход:
- Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
 - Предполагает взаимодействие между компонентами через стандартизованные двоичные интерфейсы и позволяет использовать исполняемые файлы в любом языке программирования, поддерживающем соответствующую технологию
 - Позволяет рассматривать объект исследования, как структуру, состоящую из отдельных компонент
 - способ написания исходного кода программного обеспечения
 - Позволяет собрать объекты-компоненты в динамически вызываемые библиотеки или исполняемые файлы, и распространять в двоичном виде (без исходных текстов)
 - Способ отладки и тестирования программного обеспечения

- Способ внедрения и опытной эксплуатации программного обеспечения.

- Метод выработки требований к разработке программного обеспечения

62. Управление требованиями:

- Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям

- Процесс систематического выявления, организации и документирования требований к сложной системе

- Выявление требований заказчика и управление ими

- Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности

- Процесс создания программного обеспечения и адаптация его под требования заказчика

- Разработка требований к программному обеспечению и создание ПО на основе этих требований

- Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе

- Разработка программного обеспечения и выработка требований к изменению работы системы заказчика

63. К методам выявления требований относятся:

- Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение

- Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения

- Личные встречи и беседы со всеми сотрудниками предприятия

- Анализ технической документации и на основе нее разработка требований к системе

- На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения

- Интервьюирование и анкетирование, мозговой штурм и отбор идей

- Совещания, посвященные требованиям, создание прототипов

- Раскадровки, прецеденты, обыгрывание ролей

64. Требования к разрабатываемой системе должны включать:²⁶

- Разработку программного обеспечения и выработка требований к изменению работы системы заказчика

- Совокупность условий, при которых предполагается эксплуатировать будущую систему (аппаратные и программные ресурсы, предоставляемые системе; внешние условия

ее функционирования; состав людей и работ, имеющих к ней отношение)

- Построение программного обеспечения из отдельных компонентов физически отдельно

существующих частей программного обеспечения

- Описание выполняемых системой функций

- Технологию создания сложного программного обеспечения, основанную а объектном

представлении кода программы

- Ограничения в процессе разработки (директивные сроки завершения отдельных этапов,

имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)

- Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения

- Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования

65. Типы средств, иллюстрирующие цели моделирования системы:

- Функции, которые система должна выполнять

- Отношения между данными

- Зависящее от времени поведение системы (аспекты реального времени)

- Способы отладки и тестирования программного обеспечения

- Создание программного обеспечения на основе структурной схемы исследуемого объекта или процесса
 - Выявление требований заказчика и управление ими
 - Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
 - Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
66. Преимущества объектно-ориентированного подхода:
- Быстрота написания программного кода
 - Статичность конфигурации системы
 - Возможность многократного использования
 - Низкая стоимость проекта
 - Восприимчивость к изменениям
 - Отсутствие необходимости документирования
 - Простота реализуемых моделей
 - Реалистичное моделирование
67. Требования – это:
- Документ, регулирующий отношения между заказчиком информационной системы и проектировщиком
 - Некоторое свойство программного обеспечения, необходимое пользователю для решения проблемы при достижении поставленной цели
 - Оформленное заказчиком в виде документа задание на проектирование программного обеспечения
 - Возможность, которую должна обеспечивать система
 - Характеристика проектируемого программного обеспечения с точки зрения разработчика²⁷
 - Некоторое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить требования формальной документации
 - Оформленное разработчиком в виде документа задание на проектирование программного обеспечения
 - Характеристика проектируемого программного обеспечения с точки зрения заказчика
68. Типичная схема процесса анализа С-требований включает в себя:
- Идентификацию заказчика и проведение интервью с представителями заказчика
 - Разработку программного обеспечения в соответствии с требованиями заказчика
 - Изложение заказчику требований к системе на основе разработанного программного обеспечения
 - Написание С-Требований в форме стандартного документа
 - Верификацию разработанного программного обеспечения в соответствии с требованиями заказчика
 - Составление плана мероприятий по анализу С-требований
 - Проверку С-Требований и согласование их с заказчиком
 - Адаптацию разработанного программного обеспечения в соответствии с требованиями заказчика
69. В классификацию требований к программной системе входят:
- Требования заказчика
 - Требования, накладываемые условиями эксплуатации
 - Функциональные требования
 - Требования, накладываемые аппаратными средствами
 - Нефункциональные требования
 - Требования предметной области
 - Экономические требования
 - Требования разработчиков
70. Процесс определения и анализа требований включает в себя:
- Анализ работы систем с аналогичной предметной областью
 - Анализ предметной области, сбор и классификацию требований
 - Проведение совместных совещаний с представителями заказчика
 - Разрешение противоречий и определение приоритетов

- Адаптацию требований к разрабатываемому программному обеспечению
- Декомпозицию общей задачи на подзадачи
- Проверку, спецификация и документирование требований
- Верификацию требований в соответствии с разработанным программным обеспечением

71. Опорные точки зрения конечных пользователей системы программного обеспечения

можно трактовать как:

- Источник информации о системных данных
- Структуру требований
- Источник событий
- Структуру событий
- Структуру представлений
- Получателей требований
- Источник сценариев
- Получателей системных сервисов

72. При аттестации требований выполняются следующие типы проверок документации требований:²⁸

- Проверка на совместимость
- Проверка на управляемость
- Проверка правильности требований
- Проверка на непротиворечивость
- Проверка на соответствие
- Проверка на обратимость
- Проверка на полноту и на выполнимость
- Проверка на заменяемость

73. К методам аттестации требований относится:

- Тестирование
- Обзор требований
- Верификация
- Сравнительный анализ
- Прототипирование
- Генерация случайных данных
- Генерация тестовых сценариев
- Декомпозиция

74. Уровни организационного управления при планировании разработки системы:

- Стратегический
- Тактический
- Оперативный
- Основной
- Вспомогательный
- Дополнительный
- Системный
- Аналитический

75. Для различных представлений проектируемой системы используют типы моделей:

- Статическая модель
- Динамическая модель
- Модель классов
- Модель декомпозиции
- Модель размещения
- Модель состояний
- Модель взаимодействия
- Модель агрегации

76. Классификация бизнес-процессов включает следующие классы процессов:

- Вспомогательные бизнес-процессы
- Основные бизнес-процессы
- Дополнительные бизнес-процессы
- Обеспечивающие бизнес-процессы
- Обслуживающие бизнес-процессы
- Бизнес-процессы согласования
- Бизнес-процессы управления
- Руководящие бизнес-процессы

77. Типы D-требований:

- Функциональные требования
- Интерфейсные требования²⁹
- Нефункциональные требования
- Программные требования
- Обратные требования
- Ограниченные требования
- Производительные требования
- Надежность

78. Возможные способы организации D-требований:

- По атрибутам, по компонентам
- По взаимоотношениям сущности
- По пакетам и по иерархии компонентов
- По свойствам, по классам
- По вариантам использования
- По узлам и по использованным процессам
- По состояниям и по иерархии функции
- По прецедентам, по кооперациям

79. К моделированию относится:

- Система обозначений
- Система атрибутов
- Синтаксис языка моделирования
- Система свойств
- Совокупность поведении объектов
- Совокупность графических объектов
- Семантика языка моделирования
- Совокупность текстовых объектов

80. Классификация имитационных моделей:

- Статистическая
- Адаптивная
- Статическая или динамическая
- Структурная
- Сетерминированная или стохастическая
- Непрерывная или дискретная
- Объединенная
- Декомпозиционная

81. Принципы разработки эффективного пользовательского интерфейса:

- Сложность, графика
- Структура, простота
- Связь, обработка
- Видимость, обратная связь
- Невидимость, сложность
- Толерантность, повторное использование
- Первое использование, итерация
- Интеграция, повторение

82. Принципы разработки программного обеспечения:

- Коллективный процесс разработки
- Индивидуальный процесс разработки
- Параллельный процесс разработки
- Командный процесс разработки³⁰
- Промежуточный процесс разработки
- Модель зрелости возможностей
- Модель законченности возможностей
- Модель готовности процессов

83. Типы интерфейсных требований:

- Пользовательские требования
- Аппаратные требования
- Административные требования
- Требования к производительности
- Программные и коммуникационные требования
- Требования к надежности
- Требования к устойчивости
- Атрибуты программной системы и другие требования

84. Технология проектирования определяется как совокупность составляющих:

- Поэтапная процедура
- Пошаговая процедура
- Модели и правила
- Критерий и правила
- Тестирование
- Нотации
- Прецеденты
- Классы

85. Разработка и сопровождение ИС в конкретной организации и конкретном проекте

должна поддерживаться стандартами:

- Стандарт организации
- Стандарт конкретного проекта
- Стандарт проектирования
- Стандарт оценки
- Стандарт оформления проектной документации
- Стандарт аудита
- Стандарт оформления разработки
- Стандарт пользовательского интерфейса

86. Результатами проектирования архитектуры являются:

- Модель административного интерфейса
- Модель процессов
- Модель потоков
- Модель классов
- Модель данных
- Модель пользовательского интерфейса
- Модель компонентов
- Модель узлов

87. Какие работы включает процесс разработки программного обеспечения:

- Документирование, управление конфигурацией
- Управление, создание инфраструктуры
- Структура из процессов, работ, задач
- Обеспечение качества, верификация
- Анализ требований, проектирование
- Программирование, сборка, тестирование
- Ввод в действие, приемка
- Совместный анализ, аудит

88. Какие технологии разработки программ используются в современном программировании:

- Визуальные
- Событийные
- Структурные
- Объектно-ориентированные
- Модульные
- Текстуальные
- Графические
- Машинно-ориентированное

89. Объектно-ориентированное проектирование использует инструментальные средства:

- Model mart
- Rational Rose
- Bpwin
- ARIS
- Idef1X
- Erwin
- MS Visio
- Jam

90. Проектирование функциональных моделей поддерживается инструментальными средствами:

- Jam
- Model Mart
- MS visio
- ERwin
- Idef0

- Aris
- Rational rose
- BPwin

91. IEEE - это:

- Коммерческая организация ученых и исследователей
- Просто принятое обозначение, расшифровки не имеет
- Обозначение всемирной компьютерной сети
- Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей
- Такая аббревиатура нигде не используется
- Institute Of Electrical and Electronic Engineers, Inc
- Американская организация ученых-экономистов
- Институт инженеров радиоэлектроники и электротехники

92. Ядро знаний SWEBOK - это:

- ГОСТ на разработку программного обеспечения
- Нормативный документ, разработанный IEEE
- ГОСТ на разработку информационных систем32
- Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения
- Основополагающий научно-технический документ, который отображает мнение специалистов в области программной инженерии
- Документ, устанавливающий методику тестирования и испытания программного обеспечения
- Документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207
- ГОСТ на разработку и комплектацию сопровождающей документации

93. Каждая область ядра знаний SWEBOK представляется:

- Структурной схемой
- Общей схемой описания
- Диаграммой UML
- Описанием и комментариями
- Определением понятийного аппарата, методов и средств инженерной деятельности
- Определением языка программирования
- Определением инструментов поддержки инженерной деятельности
- Иерархической диаграммой

94. К основным областям знаний SWEBOK относятся:

- Инженерия требований, проектирование ПО
- Анализ деятельности системы
- Управление проектами
- Конструирование ПО
- Управление персоналом
- Тестирование ПО, сопровождение ПО
- Управление конфигурацией
- Инженерия качества программных средств

95. К организационным областям знаний SWEBOK относятся:

- Инженерия требований
- Управление конфигурацией, управление проектами
- Конструирование ПО
- Процесс инженерии программных средств, методы и средства программной инженерии

- Проектирование ПО
- Сопровождение ПО
- Тестирование ПО

- Инженерия качества программных средств

96. В рамках Rational Unified Process (RUP) набор действий по разработке программ включает этапы:

- Создание структурных схем
- Определения входных, выходных данных
- Согласование стоимости проекта
- Согласования требований с заказчиком
- Создания бизнес-моделей

- Определение требований

- Проектирование, программирование

- Тестирование, внедрение

97. Этапы разработки консалтинговых проектов включают в себя:33

- Анализ первичных требований и планирование работ

- Снятие программного продукта с эксплуатации

- Декомпозицию задачи на подзадачи

- Разработку спецификации и документации

- Проведение обследования деятельности предприятия

- Тестирование и сопровождение программного обеспечения

- Построение моделей деятельности предприятия (модели AS - IS - "как есть" и модели

TO - BE - "как должно быть")

- Разработку программного обеспечения

98. Концепции, лежащие в основе модульного программирования:

- Объем реализации и время исполнения (реакции)

- Мера автоматизма в работе реализации и инструментах разработки

- Визуальность и тестируемость разработки

- Функциональная декомпозиция, пространственная и временная группировка информации (модульность)

- Упрощение связей

- Комментируемость функций и данных

- Надежность, устойчивость

- Безопасность

99. Инструмент разработки программ выбирается на основе:

- Визуальности, набора реализуемых технологий

- Мощности множества элементов разработки

- Системного подхода к анализу, проектированию и реализации ПО

- Функциональной декомпозиции, пространственной и временной группировка информации (модульность)

- Упрощения связей, комментируемости функций и данных

- Объема реализации и времени исполнения (реакции), надежности, устойчивости, безопасности

- Меры автоматизма в работе реализации и инструментах разработки

- Визуальности и тестируемости разработки