=Луд А.С.= Вопросы к экзамену =las=

Дисциплина: Надежность программного обеспечения

1. Свойства компьютера, определяющие его значимость в системах обработки данных. Человеческий фактор как главный признак ненадежной системы.
2. **Быстро вычисляет**
3. **Хранит большие объёмы данных**
4. **Алгоримически обрабатывает данные**
5. **Не подпускать идиота ближе чем радиус твой руки.**

2) Принципы обеспечения безопасности функционирования компьютерных систем и значимость их применения.

1. **Копия - основа восстановления**
2. **Ограничение прав доступа**

3) Принципы обеспечения надежности программного обеспечения в компьютерных системах и значимость их применения.

1. **Строгая типизация**
2. **Минимальная и достаточная документация**
3. **Жёсткий контроль и сопровождение**
4. **Не подпускать идиота ближе чем радиус твой руки.**

4) Инструментальные средства, используемые при разработке систем, и их сравнинельный анализ с точки зрения повышения надежности.

1. **Ассемблер – работа с аппаратурой и драйверами (недостатки: отсутствует мобильность, ненадежность, неэффективен)**
2. **ЯП (высокого уровня) – эффективно, понятный интерфейс, независимость от процессора (эффективная реализация)**
3. **СУБД – гарантия работы генераторов экранов и отчетов. (+суперэффективная реализация)**
4. **Библиотеки (классический язык с мощными библиотеками) – высокая эффективность. (недостатки: сложный интерфейс)**

5) Основные критерии выбора компьютерных систем с точки зрения надежности, их относительная значимость.

**Надёжность системы с точки зрения обеспечения гарантии сопровождения.**

**Наличие исходных текстов у разработчика.**

6) Понятие подсистемы, его значимость и роль в системах обработки данных с точки зрения повышения надежности.

**Подсистема - локальный участок работ, связанный с одним локальным информационным пространством.**

**За счет разделения подсистем проще осуществлять сопровождение.**

7) Типизация процедур работы с картотеками и ее роль с точки зрения повышения надежности разработки.

1. **Информационная область (id объекта)**
2. **Область данных**
3. **Типовые операции – изменение количественные**
4. **Отображение на внешних носителях**

8) Основные алгоритмы, используемые в системах обработки данных, и их типизация с точки зрения повышения надежности системы.

1. **Выборка (родной селект) из справочника**
2. **Формирование печатных форм**
3. **Сортировка – при формировании печатных форм для формирования уровня отчета**
4. **Типовой функционал работы с картотекой**
5. **Формирование тейбла на основе других тейблов**

9) Минимальный набор документирования системы.

**Оглавление лабы**

1. **Меню**
2. **Картотеки**
3. **Первичные документы**

**См описание системы**

10) Методы проектирования систем обработки данных. Типизация функциональной схемы обработки данных с точки зрения повышения надежности разработки.

**См описание системы (тейблы, линейки и стрелки тд)**

11) Методы проектирования систем обработки данных. Типизация определение картотек с точки зрения надежности.

**См описание системы (тейблы, линейки и стрелки тд)**

12) Методы проектирования систем обработки данных. Типизация описание работ с точки зрения надежности.

**См описание системы**

1. **Определить объекты.**
2. **Определить связи между объектами.**
3. **Типизация объектов**

**Вопросы 13-32 – показать на практике**

13) Типизация и специфика определения картотек в конкретных инструментальных средах с точки зрения надежности (C).

14) Типизация и специфика определения картотек в конкретных инструментальных средах с точки зрения надежности(MS Access)

15) Типизация и специфика определения картотек в конкретных

инструментальных средах с точки зрения надежности(CBuilder).

16) Типизация и специфика определения картотек в конкретных

инструментальных средах с точки зрения надежности (DBase).

17) Типизация и специфика определения картотек в конкретных

инструментальных средах с точки зрения надежности (QT).

18) Реализация типизированного двухуровневого меню (С).

19) Реализация типизированного двухуровневого меню (MS Access).

20) Реализация типизированного двухуровневого меню (CBuilder).

21) Реализация типизированного двухуровневого меню (DBase).

22) Реализация типизированного двухуровневого меню (QT).

23) Типизированное использование картотеки "настройка"

на примере общесистемных параметров (C).

24) Типизированное использование картотеки "настройка"

на примере общесистемных параметров (CBuilder).

25) Типизированное использование картотеки "настройка" на примере общесистемных параметров (MS Access).

26) Типизированное использование картотеки "настройка" на примере общесистемных параметров (DBase).

27) Типизированное использование картотеки "настройка" на примере общесистемных параметров (QT).

28) Типизированное использование картотеки "настройка" на примере указания настройки выдаваемых форм (С).

29) Типизированное использование картотеки "настройка" на примере указания настройки выдаваемых форм (CBuilder).

30) Типизированное использование картотеки "настройка" на примере указания настройки выдаваемых форм (DBase).

31) Типизированное использование картотеки "настройка" на примере указания настройки выдаваемых форм (MS Access).

32) Типизированное использование картотеки "настройка" на примере указания настройки выдаваемых форм (QT).

33) Типизация установки системы (С).

**Установка инструментальной среды (В нашем случае Geany). Установка компилятора gcc.**

34) Типизация установки системы (DBase).

**Инструкция**

35) Типизация установки системы (CBuilder).

**Инструкция**

36) Организационные особенности и методы контроля перехода к новой версии программного продукта.

1. **Интерфейс между производственными отделами и разработчиками**
2. **Старое сохраняется в специальных архивах, новые закидываются в новые версии**
3. **В случае снятия программ происходит ручное вытаскивание старых модулей**

**Сначала копия, потом изменение.**

37) Критерии грамотного сопровождения компьютерных систем.

1. **Определение поставленной задачи в группе разработчик – пользователь на расширение системы, изменения функционала или исправления ошибок**
2. **В случае аварийного завершения четкое восстановление передачи разработчикам всех сведений**

38) Критерии грамотной эксплуатации компьютерных систем.

1. **Наличие инструкций уровня персонала**
2. **Запрещение работ вне рамок инструкций**
3. **В случае аварийного завершения четкое восстановление передачи разработчикам всех сведений**

39) Разделение полномочий и ответственности при эксплуатации компьютерных систем.

**Полномочия делятся между программистами и теми, кто эксплуатирует программами. Программисты - при изменении внесение изменение в основную систему, обеспечение сохранение рабочей копии, копирование своих программ, в случае заявки от эксплуататора модернизация. Эксплуататоры - запуск программ, информирование об ошибках, переход на копию в случае поломки.**

**Вопросы 40 – 49 - продемонстрировать на схеме.**

40) Типовая модель обработки данных в проекте реализованном в конретной инструментальной среде (С).

41) Типовая модель обработки данных в проекте реализованном в конретной инструментальной среде (СBuilder).

42) Типовая модель обработки данных в проекте реализованном в конретной инструментальной среде (DBase).

43) Типовая модель обработки данных в проекте реализованном в конретной инструментальной среде (MS Access).

44) Типовая модель обработки данных в проекте реализованном в конретной инструментальной среде (QT).

45) Типизация сопровождения картотек первичного ввода с точки

зрения надежности (C).

46) Типизация сопровождения картотек первичного ввода с точки

зрения надежности (DBase).

47) Типизация сопровождения картотек первичного ввода с точки

зрения надежности (MS Access).

48) Типизация сопровождения картотек первичного ввода с точки

зрения надежности (CBuilder).

49) Типизация сопровождения картотек первичного ввода с точки

зрения надежности (QT).

50) Заполнение реквизитов с "подсказкой" из картотеки на примере картотеки первичных документов (MS Access). Минусы такого подхода с точки зрения надежности.

**Минусы – пользователь должен пользоваться только таблицами, не свойствами.**

**Вопрос 51 – 54 - Инструкция (возможно «случайное» удаление кнопки преподавателем, нужно будет восстановить)**

51) Заполнение реквизитов из справочника на примере картотеки

первичных документов (DBase).

52) Заполнение реквизитов из справочника на примере картотеки

первичных документов (С).

53) Заполнение реквизитов из справочника на примере картотеки

первичных документов (CBuilder).

54) Заполнение реквизитов из справочника на примере картотеки

первичных документов (MS Access). Минусы реализации с точки

зрения обеспечения надежности сопровождения.

**Вопрос 55 – 59**

55) Типизация сопровождения картотеки на примере картотеки

типовые операции (C).

56) Типизация сопровождения картотеки на примере картотеки

типовые операции (CBuilder).

57) Типизация сопровождения картотеки на примере картотеки

типовые операции (DBase).

58) Типизация сопровождения картотеки на примере картотеки

типовые операции (MS Access).

59) Типизация сопровождения картотеки на примере картотеки

типовые операции (QT).

**Вопрос 60-64 - если есть типизированная группа документов, то выгодно использовать типизированную форму регистрационного журнала (с каждого документа разноска в журнал и потом одна ветка)**

60) Место и роль регистрационных журналов в компьютерныx системах (С).

61) Место и роль регистрационных журналов в компьютерныx системах (DBase).

62) Место и роль регистрационных журналов в компьютерныx системах (MS Access).

63) Место и роль регистрационных журналов в компьютерныx системах (CBuilder).

64) Место и роль регистрационных журналов в компьютерныx системах (QT).

**Вопрос 65-69 – Главное правило - readonly**

65) Экранная форма для просмотра картотек (C).

66) Экранная форма для просмотра картотек (DBase).

67) Экранная форма для просмотра картотек (MS Access).

68) Экранная форма для просмотра картотек (CBuilder).

69) Экранная форма для просмотра картотек (QT).

**Вопрос 70-74 - выносить поля таким образом, чтобы потом использовать селект, сортировку и сумму.**

70) Реализация процедуры "двойная запись". Ее смысл с точки зрения обработки данных (C).

71) Реализация процедуры "двойная запись". Ее смысл с точки зрения обработки данных (DBase).

72) Реализация процедуры "двойная запись". Ее смысл с точки зрения обработки данных (MS Access).

73) Реализация процедуры "двойная запись". Ее смысл с точки зрения обработки данных (CBuilder).

74) Реализация процедуры "двойная запись". Ее смысл с точки зрения обработки данных (QT).

**Вопрос 75-79 – программа?**

75) Реализация просмотра книги счетов. Ее смысл с точки зрения программиста (C).

76) Реализация просмотра книги счетов. Ее смысл с точки зрения программиста (DBase).

77) Реализация просмотра книги счетов. Ее смысл с точки зрения программиста (MS Access).

78) Реализация просмотра книги счетов. Ее смысл с точки зрения программиста (CBuilder).

79) Реализация просмотра книги счетов. Ее смысл с точки зрения программиста (QT).

80) Формирование списковых форм на примере формирования оборотных ведомостей (C).

**Показать: сначала программа в промежуточный файл на уровне выбрать, потом сортировать, потом программа генерации отчета**

81) Формирование списковых форм на примере формирования оборотных ведомостей (DBase)

**Показать query**

82) Формирование списковых форм на примере формирования оборотных ведомостей (MS Access).

**Показать query**

83) Формирование списковых форм на примере формирования оборотных ведомостей (CBuilder).

**Показать без сортировки, только списковая печатная форма**

84) Формирование списковых форм на примере формирования оборотных ведомостей (QT).

**Вопрос 85 – 88 – показать балансовую ведомость и журнал ордера**

85) Формирование итоговых форм на примере формирования журнала-ордера (DBase).

86) Формирование итоговых форм на примере формирования журнала-ордера (MS Access).

87) Формирование итоговых форм на примере формирования балансовой ведомости (MS Access).

88) Формирование итоговых форм на примере формирования балансовой ведомости (DBase).

89) Операционные системы: функциональное назначение и особенности обеспечения надежности.

1. **Обеспечение физического в/в**
2. **Обеспечение файловой системы**
3. **Отслеживание ресурсов, запуск программ и снятие в случае аварийного завершения**

90) Кодирование информации в ЭВМ. Кодовые таблицы. Шрифты. Проблемы обеспечения надежного интерфейса пользователя.

**Кодировки показать в geany**

1. **Связка с ОС (дос, вин)**
2. **Графическое отображение на мониторе (пропорционально или нет)**
3. **Поддержка знаковой клавиатуры**

**Аварийное завершение**

91) Представление информации в ЭВМ: структурные переменные, адресация, понятие типа данных. Обозначение переменных и надежность.

**Структурные переменные – операции на уровне рекода, надежность позиционирования.**

**Адресация – самый слабый момент в надежности.**

**Тип данных – минимизация типов данных (цифр, строк, даты) – позволяет обеспечивать надежное обеспечение системы. Правильная система именования переменных (привязка к тейблу, привязка к модулю).**

92) Классификация программ по целевому назначению. Специфика копирования программ исходя из их целевого назначения.

1. **ос**
2. **прикладные системы общего назначения**
3. **специализированные системы с мин изменчивости (копия)**
4. **специализированные системы с высокой степенью изменчивости (копия и версии сопровождения)**
5. **инструментальные среды**

93) Компоненты ПЭВМ, их характеристика и влияние на производительность и надежность системы в целом.

**Процессор, оперативная память, внешняя память, материнская плата, периферийные устройства и адаптеры**

**Слабости:**

1. **внешняя память - мощная буферизация**
2. **периферийные устройства - зависимость от человека**

94) Текстовый и графический режим работы, влияние режима ввода-вывода на производительность и надежность системы.

**В графическом режиме объемы передачи данных превосходят в разы, отсюда затраты на операцию и производительность**

**Обслуживание графики на порядок сложнее**

95) Системы счисления. Показать просмотр файла в символьном и шестнадцатиричном формате.

**Бинарная, восьмеричная, hex, десятичная**

96) Типы данных: их переопределение в языках программирования на примере C. Связь с надежностью сопровождения.

**Binary, address, char, string, struct, enum**

**Массовые ошибки при попытке преобразования с одной системы в другую**

**Связь с fox**

97) Повышение надежности при использовании условных операторов.

**else-запрещены (не видно вложенности)**

98) Типизация оформления исходного текста с целью повышения надежности сопровождения.

**Вложенная блочность (скобки {})**

99) Текстовые, объектно-ориентированные и графические редакторы. Типизация редактирования текстов с целью повышения надежности сопровождения.

**Редакторы:**

1. **текстовые (теги, можно редактировать таблицы (шрифт моноширинный (непропорциональный)))**
2. **графические(пиксель)**
3. **объектно-ориентированные (могут вставляться различные объекты)**

**Классические операции (добавить, вставить, редактировать)**

**Самые простые и эффективные – текстовые**

100) Виды поражения систем. Восстановление систем.

**Виды:**

**1) Игровое**

**2) Видимое поражение**

**3) Поражение невидимое**

**4) Поражение аппаратуры**

**Восстановление системы:**

**1) Сброс ситуации для разбора**

**2) Сброс ОС и установка с запечатанных копий**

**3) Установка программ с запечатанных копий**

**4) Для своих систем сброс данных доввод данных**

101) Архиваторы, их использование в производственных системах(показать на примере архиватора от прозводителя).

**Оригинальные - от производителя**

**Системные архиваторы - более слабые вследствие на фирме должен быть использован только 1 архиватор**

102) Принципы обеспечения безопасности системы.

1. **Копия - основа восстановления**
2. **Ограничение прав доступа**
3. **Не подпускать идиота ближе чем радиус твой руки.**

103) Этапы разработки и эксплуатации программного обеспечения.Специфика этапов с точки зрения обеспечения надежности

**1) проектирование**

**2) разработка**

**3) передача**

**4) сопровождение**

**Самые серьезные (затратные) просчеты на этапе проектирования.**

**При передаче в эксплуатацию должно быть:**

1. **Тестирование в дружественном режиме**
2. **Должны существовать протоколы по снятию системы**

**Хороший продукт – продукт с отработанными процедурами сопровождения.**

**Принципы именования:**

1. **Самоидентификация**
2. **Подчеркивание - святой знак**
3. **Двухуровневая система - диск, каталог, файл, больше и меньше не надо**
4. **Только англ буквы**
5. **Шифрование, файлы должны начинаться как каталог и Т.д.**

104) Типизация операторов языка програмирования.

1. **Определение данных (классика (бинарники, строка, структура, массив), адресные переменные, связывание адресных с классикой (в С строка - массив символов, адрес на массив символов), присвоение (специфика бинарных операций - флажки, взятие адреса)**
2. **Оператор ветвления if (else нельзя – не видно вложенности)**
3. **Функция (передача параметров по значению, по адресу, например можно передавать адрес, для изменения значений в структуре, только одна переменная передается)**
4. **Цикл (только WHILE DO).**

**Вопросы 105-109.**

1. **Два уровня**
2. **Работы должны соответствовать сложившемуся функционалу - все показать. Не стоит придумывать свои термины, все должно быть понятно пользователю.**

105) Принципы "хорошего" меню. Реализация меню (C).

106) Принципы "хорошего" меню. Реализация меню (DBase). Минусы данной реализации.

107) Принципы "хорошего" меню. Реализация меню (MS Access). Минусы данной реализации.

108) Принципы "хорошего" меню. Реализация меню (CBulder).

109) Принципы "хорошего" меню. Реализация меню (QT).

110) Использование адресных переменных с точки зрения обеспечения надежности.

**В системах позволяющими работать с адресными переменными, надежность нулевая, поэтому Джава без указателей.**

111) Операторы цикла с точки зрения обеспечения надежности.

**While do – лучше всего.**

**For - слабо понятно внешне, много операций**

**Do while - сначала проверь потом делай**

**For each - зачем? и так всё есть**

112) Использование структур с точки зрения надежности.

1. **При передаче параметров легче отследить трафик.**
2. **Более простая передача параметров.**

113) Препроцессорная обработка данных с точки зрения обеспечения надежности разработки.

**Препроцессор позволяет создавать свою семантическую среду (спрятать вызовы с типизацией, можно определить переменные, все свое создать +спрятанные параметры с вытекающей надежностью)**

114) Использование заголовочных файлов с точки зрения

обеспечения надежности разработки.

**Одно место, контроль за вводом параметров. В отдельное место выносится функционал, море настроек, надежность растет, защищены от придурков (хэдеры).**

115) Правила "хорошего" кодирования (формат исходных текстов) на примере простого программного приложения.

**См. Гудлиф - Ремесло программиста**

**Основное - скобки с табуляцией**

116) Сведение условных операторов к простой форме оператора IF

на примере.

**См. 104**

117) Сведение операторов цикла к оператору WHILE-DO на примере.

**См. 104**

118) Операционные системы: функциональное назначение и особенности защиты.

**См. 89**

**Вопросы 119-121**

**Вырезать, сортировать по уровням, передача управление на создание печатной формы. 1 и 2 - sql средствами**

119) Типовое формирования печатной формы (DBase).

120) Типовое формирования печатной формы (MS Access).

121) Типовое формирования печатной формы (QT). +

122) Динамическое распределение памяти. Проблема защита памяти.

**Ресурсный объект программа требует память, память выделяется и передается управление. Проблемы - фрагментация памяти. (dll так работают)**

123) Факторы, влияющие на производительность ЭВМ. Использование буферизации ввода-вывода. Переполнение буферов.

**Факторы: Кол-во слабых операций ввода-вывода (ручной ввод и механические внешние устройства). Один из способов реализации - буферизация в оперативной памяти, но это вызывает дополнительные проблемы с надежностью.**

**Переполнение: отсутствии контроля пересечение по памяти идет. (100 байтов зарезервировал, а пошло 120, все)**

124) Принципы именования информационных объектов.

**Принципы именования:**

1. **Самоидентификация**
2. **Подчеркивание - святой знак**
3. **Двухуровневая система - диск, каталог, файл, больше и меньше не надо**
4. **Только англ буквы**
5. **Шифрование, файлы должны начинаться как каталог и Т.д.**

125) Специфика шрифтов с точки зрения надежности сопровождения.

1. **Пропорциональный и непропорциональный**
2. **Операционка (дос, винда)**
3. **Поддерживает русскую кодировку или нет (специфика поддержки)**

126) Специфика использования С++ с точки зрения надежности.

**Классы придуманы для того чтобы идиотам было легче жить.**

**+ Надежность повышается**

**- В слишком ограниченном пространстве жить скучно**

127) Основные модели используемые в теории надежности. Показатели

надежности.

**Хотя математические модели надежности являются значительной идеализацией законов функционирования технических объектов (систем), они позволяют в вероятностной форме предсказать поведение объектов в реальных условиях функционирования и оценить многие количественные характеристики надежности. При этом степень идеализации в основном определяется требованием простоты используемых моделей. Сложные модели надежности могут потребовать очень большого объема выборки для оценки ее параметров при экспериментальных исследованиях, в результате чего использовать такие модели становится технически и экономически невыгодно (бессмысленно). Математические модели надежности элементов, используемые на практике, представляют собой, как правило, простые законы распределения, которые выражаются элементарными функциями или их интегралами, – законы надежности.**

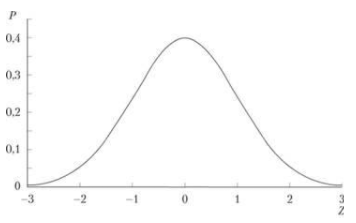
**Показатели надежности:**

1. **функция распределения отказов (оценка ВО)**
2. **функция надежности (оценка ВБР)**
3. **плотность распределения отказов (оценка ПРО)**
4. **интенсивность отказов (оценка ИО)**

128) Оценка надежности аппаратных средств. Надежность программного обеспечения, ее специфика.

**Существуют вероятностные методы оценок. Законы связанные с нормальным распределением (вопрос может быть, горочка. дисперсия), чисто вероятностная отработка системы.**

**Нормальное распределение:**

****

**Обкатка - первая фаза, слева внизу графика; на верху рабочие устройства, оптимальные; справа внизу старые, нерабочие**

129) Причины появления ошибок в программном обеспечении. Повышение надежности программных средств.

**1 несоответствие пунктов**

**1 типизации**

**2 минимальное и достаточное документирование**

**3 жесткий контроль за всем**

**4 поведение идиотов непредсказуемо**

130) Модели и оценка надежности программного обеспечения

согласно стандартов.

**Модели: ГОСТ 28806-90 и СТБ ISO/IEC 9126**

**В обоих присутствуют такие характеристики надежности, как завершенность, устойчивость и восстанавливаемость. СТБ ISO/IEC 9126 так же содержит согласованность надежности.**

**Бахтизин, В. В. предлагает методику оценки надежности программных средств (ПС) основанную на: оценке надежности отдельно взятого модуля, оценке веса модуля, определении метрик для оценки веса модуля (Этап выполнения, время работы, частота использования, объем, время, затраченное на разработку).**

131) Организация тестирования программного обеспечения. Проектирование тестов.

1. **Тестирование может быть конкретной ситуации, локальное тестирование**
2. **Когда происходит изменение, создаются тесты на проверку всей системы в целом**

132) Модульное тестирование. Тестирование проекта.

**См 131**

133) Регрессионное тестирование.

**Запускаете полный цикл тестов, которые проверяют все ситуации, чтобы проверить, что ваше изменения не повлекли за собой нарушение все системы.**

134) Документирование тестирования.

**Последний листик в фоксе и эксесе - карта тестирования, выполнение работы. Подача данных, ответная реакция - это входит в карту тестирования.**

135) Жизненный цикл дефекта программного обеспечения.

**Наличие дефекта не вызывающее снятие (не увидишь при отладке)**

136) Сопровождение и тестирование версий программного продукта.

**Сопровождение - это изменение программ исходя из: ошибки, расширения функционала, изменения функционала. Гарантия надежного П продукта - это гарантия и сопровождение.**

**Тестирование - проверка сданного в работу продукта с использованием тестов, которые отвечают списку работу, которые обязан выполнять это ПП.**

**Версии ПП - разработка, сопровождение, тестирование строится на основе от малого к большому (минимум обеспечния функций, + больше функций, +больше, и т.д.)**

137) Согласование работы разработчиков и эксплуатационных служб.

**Задача разработчиков - Обеспечение сохранения старых модулей и передача новых. Эксплутационники - В случае снятия новых, из старых буферов восстанавливают.**

138) Циклическое копирование.

**Никогда нельзя делать копию на один носитель. (1-ый, 2-ой)**

139) Осмысленное документирование как фактор надежности.

**Показываете свое описание Accessa, функциональные схемы.**

140) Типизация объектов как фактор надежности.

**Минимальный комплект сопровождения и документирования согласно реалиям, а именно скрины, тейблы, отображения проги, квери, репорты, менюшки, документирование. Должно быть ничего типизированного.**

141) Жесткий контроль документирования и типизации объектов как фактор надежности.

**Показываете свое описание Accessa, функциональные схемы, к нему задаются вопросы.**

142) Социальные типажи программистов-вредителей.

1. **Умная сволочь. Почему опасен? Умный. Выгонять.**
2. **Псевдогуру**
3. **Лентяй**
4. **Остальные (идиоты, ковбои), например, провода таскать**

143) Надежность как главный параметр качества программного продукта. Гарантии сопровождения как главный параметр надежности программного продукты.

**Повторить первое предложение. (ответ в вопросе). Гарантия сопровождения - наличие исходных текстов. Гарантированные по обязательствам разработчика.**

144) Обеспечение версий программного продукта в системах реального времени.

**Буферное хранение версий.**

**В системах реального времени: ни в коем случае нельзя размещать системы с большой динамикой в оперативке.**

145) Влияние построения командной работы разработчиков на надежность программного обеспечения.

**Четкая иерархия выполнения работ (начальник, подчиненный) и децентрализованные работы, которые носят творческий характер. Жесткий контроль, зарплата, и наказание за разгильдяйство.**

146) Особенности инструментальной среды с точки зрения надежности программного обеспечения (С).

**Си - это по сути ассемблер. Язык - человекоподобный интерфейс. Гарантия надежности. Все отслеживается. Транслятор простейший.**

147) Особенности инструментальной среды с точки зрения надежности

программного обеспечения (DBase).

**Языковая среда, исключительно простая. Вероятность ошибки почти нулевая.**

148) Особенности инструментальной среды с точки зрения надежности программного обеспечения (CBuilder).

**Выделенные ресурсы для передачи параметров отслеживаются обыкновенным тестовым редактором.**

149) Особенности инструментальной среды с точки зрения надежности программного обеспечения (MS Access).

**Access - интерфейсы запутанные, нестандартная обработка данных, полюби Майкрософт.**

150) Особенности инструментальной среды с точки зрения надежности программного обеспечения (QT).

**Не будут особо спрашивать. Главное, чтобы работало.**

151) Макросы с точки зрения повышения надежности.

**Макрос - это расширенная прога. С помощью макроса можно создать свою языковую систему.**

152) Библиотеки с точки зрения повышения надежности.

**Типизированные модуля выносятся с высокой степенью реализации. Уменьшается нагрузка на типизированные объекты. Но переработанные библиотеки - это плохо.**

153) Копия как основа восстановления. Типизация процедур восстановления.

**На основе копии**

1. **попытаться сделать при поражении копии**
2. **установка операционки запечатанных копий**
3. **установка программ запечатанных копий**
4. **копирование данных последней копии**
5. **доввод данных после последнего копирования.**

154) "Кряки" и надежность программного продукта.

**Если есть кряк, то продукт ненадежен.**

155) Лицензионные соглашения с точки зрения надежности.

**Сравнить лицензию Qt и GTK. Qt хреновая, замотаешься продавать. Почему?**

156) Учет прав доступа при документировании разработки.

**Разрабы делятся на**

**1) базовые/основные и**

**2) пользователи. Каталоги можно менять людям с определенными правами доступа.**

157) "Облачные" технологии для разработки программного обеспечения и надежность.

**Все что на облаках, вроде не твое. Общее.**