1. Дайте определение понятию ЖЦ ПО. Какой документ регламентирует состав процессов ЖЦ ПО?

Жизненный цикл программного обеспечения определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации

Основным нормативным документом, регламентирующим состав процессов ЖЦ ПО, является международный стандарт ISO/IEC 12207:1995"Информационные технологии - Процессы жизненного цикла программного обеспечения "

2. Назовите основные объекты диаграмм функциональных спецификаций.

Основными объектами диаграмм функциональных спецификаций являются:

1. Функциональные блоки - описывают функциональность системы
2. Связи между функциональными блоками - описывают взаимодействие между функциональными блоками
3. Входные и выходные данные - описывают данные, которые система получает и возвращает.

3. Опишите спиральную модель разработки ПО, приведите схему, укажите достоинства и недостатки.

Спиральная модель разработки ПО (Spiral Model) - это модель разработки, которая сочетает в себе элементы итеративной и инкрементальной модели. Она используется для разработки больших и сложных систем.

Спиральная модель включает в себя следующие этапы:

Планирование: определение целей и задач разработки, оценка рисков и планирование ресурсов.

Анализ: изучение требований и построение архитектуры системы.

Разработка: создание кода и тестирование системы.

Оценка и итерация: оценка результатов разработки и планирование изменений для следующей итерации.

Схема спиральной модели разработки ПО представляет собой несколько спиралей, каждая из которых соответствует одной итерации разработки.

Достоинства:

Возможность управления рисками на каждом этапе разработки.

Возможность постоянной обратной связи и изменения требований к системе.

Возможность постоянного тестирования и оценки системы.

Недостатки:

Высокая сложность и затраты на планирование и управление каждой итерацией.

Возможность задержек и простоев в разработке из-за постоянного изменения требований.

Возможность недостаточной документации и отсутствия полного понимания системы из-за множества итераций.



5. Дайте определение понятию «требование». Назовите показатели качества ПО и критерии оценки качества программы. (50/50)

Требование — это формулировка ожидаемого свойства, поведения или характеристик продукта.

6. Дайте описание компонентам DFD: внешние сущности системы, подсистемы, процессы, накопители данных, потоки данных.

Внешние сущности системы (External Entities) - это объекты или агенты, которые находятся снаружи системы, но взаимодействуют с ней. Например, пользователь, другая система, организация и т.д.

Подсистемы (Subsystems) - это части системы, которые могут быть отдельно изучены и поняты. Они могут включать в себя другие подсистемы, процессы и накопители данных.

Процессы (Processes) - это действия или функции, которые выполняются системой или ее подсистемой. Они могут принимать данные от внешних сущностей или других процессов и возвращать результаты.

Накопители данных (Data Stores) - это базы данных или хранилища, в которых система или ее подсистемы хранят информацию.

Потоки данных (Data Flows) - это потоки информации, которые передаются между внешними сущностями, процессами, подсистемами и накопителями данных в системе. Они показывают, как данные передаются и обрабатываются в системе, и могут быть использованы для описания бизнес-процессов и логики работы системы.

7. Назовите преимущества и недостатки паттерна «Фабричный метод».

**Достоинства:**

* Создает объекты разных типов, позволяя системе оставаться независимой как от самого процесса создания, так и от типов создаваемых объектов.

**Недостатки:**

* В случае классического варианта паттерна даже для порождения единственного объекта необходимо создавать соответствующую фабрику

8. Дайте определение понятиям «сущность», «связь», «атрибут» для диаграмм моделирования процессов.

Сущность (Entity) - это конкретный объект, информация о котором обрабатывается в контексте определенного процесса или системы. Например, продукт, заказ, клиент, задача и т.д.

Связь (Relation) - это отношение между сущностями, которое определяет какую-либо логику или зависимость между ними. Например, заказ может иметь связь с клиентом, который его сделал, или задача может иметь связь с проектом, к которому она относится.

Атрибут (Attribute) - это характеристика, описывающая сущность или связь. Например, у заказа может быть атрибут "номер", "дата" или "статус", у клиента "имя" и "адрес", и у задачи "название" и "дата окончания".

9. Какой набор диаграмм содержит стандарт UML версии 1.1? Опишите принципы построения модели IDEF0.

Стандарт UML версии 1.1 содержит следующий набор диаграмм:

Диаграмма классов (Class Diagrams)

Диаграмма объектов (Object Diagrams)

Диаграмма последовательности (Sequence Diagrams)

Диаграмма взаимодействия (Collaboration Diagrams)

Диаграмма состояний (State Diagrams)

Диаграмма активности (Activity Diagrams)

Диаграмма компонентов (Component Diagrams)

Диаграмма деплоя (Deployment Diagrams)

Модель IDEF0 (Integration DEFinition for Function Modeling) является методом моделирования бизнес-процессов, который создан для описания и анализа системы управления. Она использует специальные символы и связи для описания функций системы и их взаимодействия. Основные принципы построения модели IDEF0:

Функциональность системы должна быть описана на различных уровнях детализации.

Функциональность системы должна быть описана в терминах входных и выходных данных.

Функциональность системы должна быть описана в терминах действий, которые должны быть выполнены для достижения результата.

Функциональность системы должна быть описана в терминах последовательности действий, которые должны быть выполнены.

Модель IDEF0 использует метод иерархического декомпозирования, который позволяет описать систему на различных уровнях детализации, начиная с высокоуровневого представления и постепенно уточняя его до более низкоуровневого представления.

Модель IDEF0 использует специальные символы и связи, чтобы описать функциональность системы и ее взаимодействия.

Модель IDEF0 позволяет анализировать систему, обнаруживать проблемы и идентифицировать возможности для улучшения.

10. Охарактеризуйте каскадную модель разработки ПО, приведите схему, укажите достоинства и недостатки

Каскадная модель разработки ПО (Waterfall Model) - это методология разработки ПО, в которой каждый этап разработки (например, анализ, проектирование, реализация, тестирование) выполняется последовательно, один за другим. Схема каскадной модели может быть представлена ​​следующим образом:

Анализ требований

Проектирование

Реализация

Тестирование

Внедрение

Поддержка

Достоинства каскадной модели:

Простота и надежность. Каскадная модель имеет очень хорошо определенные этапы и процессы, которые можно легко контролировать и отслеживать.

Легкость планирования. Каскадная модель позволяет легко планировать и запланировать все этапы разработки заранее.

Легкость управления заказчиком. Каскадная модель позволяет заказчику легко понимать и отслеживать развитие проекта, так как каждый этап разработки имеет свои четко определенные цели и дедлайны.

Недостатки каскадной модели:

Невозможность динамического изменения требований. Каскадная модель сложно адаптировать к изменению требований в процессе разработки, так как каждый этап завершен перед началом следующего.

Невозможность использования инкрементального подхода. Каскадная модель не поддерживает инкрементальный подход, когда проект разрабатывается поэтапно, отдельными частями, которые могут быть запущены и тестированы на ранней стадии разработки.

Невозможность отладки на ранней стадии. Каскадная модель не позволяет отлаживать и тестировать систему на ранних этапах разработки, потому что каждый этап выполняется последовательно.

11. Дайте определение понятию «диаграмма последовательности».

Диаграмма последовательности является другой формой визуализации взаимодействия в  
модели и, как и диаграмма кооперации, оперирует объектами и сообщениями.  
На диаграмме последовательности неявно присутствует ось времени, что позволяет визуализировать временные отношения между передаваемыми сообщениями. С помощью диаграммы последовательности можно представить взаимодействие элементов модели как своеобразный временной график "жизни" всей совокупности объектов, связанных между собой  
для реализации варианта использования программной системы, достижения бизнес-цели или выполнения какой-либо задачи.

12. Назовите элементы, присутствующие на диаграмме вариантов использования. Опишите их.

Элементы, присутствующие на диаграмме вариантов использования, включают:

Актор: представляет роль, которую играет пользователь, интерактивно взаимодействующий с системой.

Вариант использования: представляет сценарий использования системы, описывающий диалог между актором и системой.

Отношение использования: представляет связь между двумя вариантами использования, описывающими зависимость между ними.

Область видимости: представляет область, в которой доступен определенный вариант использования.

Диаграмма вариантов использования позволяет описать как система будет использоваться и как она будет взаимодействовать с пользователем и другими системами. Он помогает проектировщикам и заказчикам лучше понимать и описывать требования к системе, идентифицировать возможные сценарии использования, а также планировать тестирование и отладку. Диаграмма вариантов использования является важным элементом моделирования процессов и помогает обеспечить согласованность между требованиями к системе и ее реализацией.

13. Назовите преимущества и недостатки паттерна «Абстрактная фабрика».

**Достоинства:**

- Скрывает сам процесс порождения объектов, а также делает систему независимой от типов создаваемых объектов, специфичных для различных семейств или групп (пользователи оперируют этими объектами через соответствующие абстрактные интерфейсы);

- Позволяет быстро настраивать систему на нужное семейство создаваемых объектов.

**Недостатки:**

- Трудно добавлять новые типы создаваемых объектов или заменять существующие, так как интерфейс базового класса абстрактной фабрики фиксирован.

14. Дайте определение понятию «диаграмма состояний».

Диаграмма состояний по существу является графом специального вида (который представляет некоторый автомат). Вершинами этого графа являются состояния. Дуги графа служат для обозначения переходов из состояния в состояние. Диаграммы состояний могут быть вложены друг в друга, образуя вложенные диаграммы более детального представления отдельных элементов модели.

15. Назовите графические элементы и их назначение на диаграмме последовательности.

Диаграмма последовательности использует следующие графические элементы и их назначение:

Лицо: представляет сущность, которая может отправлять или получать сообщения. Это может быть класс, объект, компонент или другая сущность.

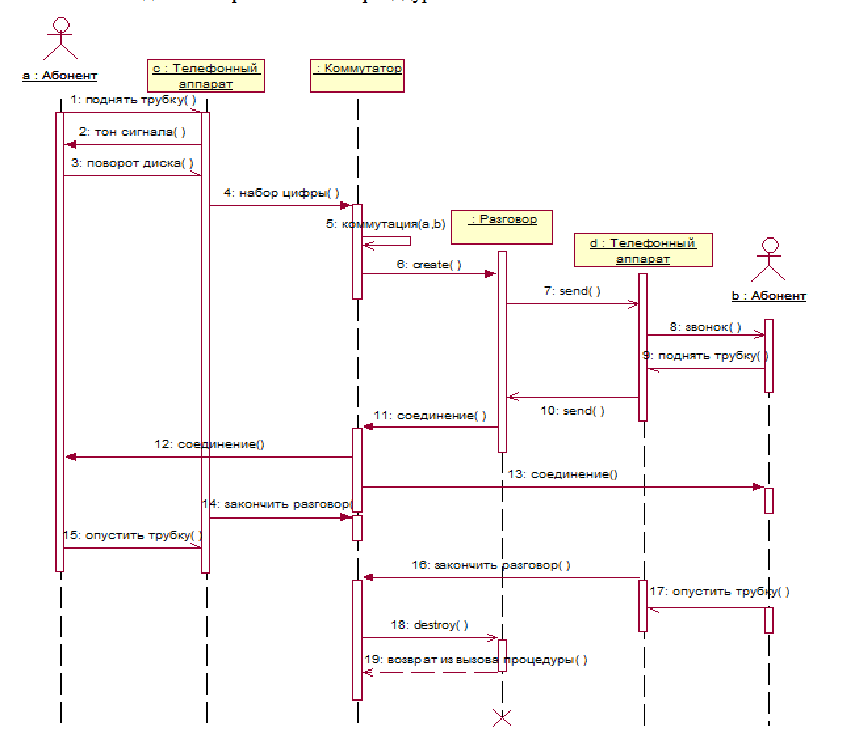
Сообщение: представляет взаимодействие между лицами.

Стрелка: представляет направление сообщения от отправителя к получателю.

Нумерованная линия времени: отображает последовательность сообщений во времени.

Временной интервал: отображает промежуток времени, который проходит между двумя сообщениями.

Область видимости: представляет область, в которой доступен определенный лицо или сообщение.

Диаграмма последовательности используется для моделирования интерактивных процессов и диалогов между сущностями в системе. Она помогает понять и описать последовательность сообщений, обмен данными и взаимодействия между сущностями в системе.

16. Опишите достоинства и недостатки восходящего проектирования

При восходящем проектировании разработка идет снизу вверх. На первом этапе разрабатываются модули самого низкого уровня.

Преимущество восходящего программирования – не нужно писать заглушки.

Недостаток восходящего программирования – головной модуль разрабатывается на завершающем этапе проектирования, что порой приводит к необходимости дорабатывать модули более низких уровней.

17. Дайте определение понятию «диаграмма деятельности».

На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Сам же результат может привести к изменению состояния системы или возвращению некоторого значения.

18. Назовите графические элементы и их назначение на диаграмме кооперации.

Диаграмма кооперации (Sequence Diagram) использует следующие графические элементы и их назначение:

Лицо (Actor): представляет роль, которую играет пользователь, интерактивно взаимодействующий с системой или другими лицами.

Объект (Object): представляет сущность, которая может отправлять или получать сообщения. Это может быть класс, объект, компонент или другая сущность.

3. Сообщение (Message): представляет взаимодействие между лицами или объектами, описывает действия и связи между ними.

Стрелка (Arrow): представляет направление сообщения от отправителя к получателю.

Нумерованная линия времени (Timeline): отображает последовательность сообщений во времени.

Временной интервал (Duration): отображает промежуток времени, который проходит между двумя сообщениями.

Диаграмма кооперации используется для моделирования последовательности сообщений и взаимодействия между объектами или классами в системе. Она помогает понять как объекты или классы взаимодействуют между собой, какие сообщения они отправляют и получают и как изменяется состояние объектов в результате этого взаимодействия.

19. Назовите преимущества и недостатки паттерна «Строитель».

**Достоинства:**

- Возможность контролировать процесс создания сложного продукта.

- Возможность получения разных представлений некоторых данных.

**Недостатки:**

- ConcreteBuilder и создаваемый им продукт жестко связаны между собой, поэтому при внесеннии изменений в класс продукта скорее всего придется соответствующим образом изменять и класс ConcreteBuilder.

20. Какие модели ЖЦ ПО вы знаете? Перечислите и опишите их.

**Каскадная модель** Принципиальной особенностью каскадного подхода является следующее: переход на следующую стадию осуществляется только после того, как будет полностью завершена работа на текущей стадии, и возвратов на пройденные стадии не предусматривается.

**Спиральная модель** Спиральная модель избавляет пользователей и разработчиков ПО от необходимости полного и точного формулирования требований к системе на начальной стадии, поскольку они уточняются на каждой итерации.

21. Назовите графические элементы и их назначение на диаграмме состояний.

22. Назовите преимущества и недостатки паттерна «Прототип».

**Достоинства:**

- Для создания новых объектов клиенту необязательно знать их конкретные классы.

- Возможность гибкого управления процессом создания новых объектов за счет возможности динамических добавления и удаления прототипов в реестр.

**Недостатки:**

- Каждый тип создаваемого продукта должен реализовывать операцию клонирования clone(). В случае, если требуется глубокое копирование объекта (объект содержит ссылки или указатели на другие объекты), это может быть непростой задачей.

23. Что такое модульное программирование?

Модульное программирование — это процесс разделения компьютерной программы на отдельные подпрограммы. Модуль — это отдельный программный компонент. Он часто может использоваться в различных приложениях и функциях с другими компонентами системы.

24. Назовите основные элементы графической нотации диаграммы деятельности.

25. Опишите достоинства и недостатки нисходящего проектирования.

Недостатки нисходящего проектирования: необходимость заглушек;  
до самого последнего этапа проектирования неясен размер программного  
комплекса и его эксплуатационные характеристики, за которые, как правило, отвечают модули самого низкого уровня.

Преимущество нисходящего проектирования – на самом начальном этапе проектирования отлаживается головной модуль (логика программы).

26. Что такое рутинность модуля? Дайте определение.

*Рутинность модуля* − это его независимость от предыстории обращений к нему. Модуль будем называть рутинным, если результат (эффект) обращения к нему зависит только от значений его параметров (и не зависит от предыстории обращений к нему). Модуль будем называть зависящим от предыстории, если результат (эффект) обращения к нему зависит от внутреннего состояния этого модуля, изменяемого в результате предыдущих обращений к нему. Майерс не рекомендует использовать зависящие от предыстории (непредсказуемые) модули, так как они провоцируют появление в программах хитрых (неуловимых) ошибок. Однако такая рекомендация является неконструктивной, так как во многих случаях именно зависящий от предыстории модуль является лучшей реализаций информационно прочного модуля. Поэтому более приемлема следующая (более осторожная) рекомендация:

27. Перечислите компоненты DFD диаграммы и опишите их.

28. Назовите преимущества и недостатки паттерна «Одиночка».

**Достоинства:**

* Класс сам контролирует процесс создания единственного экземпляра;
* Шаблон легко адаптировать для создания нужного числа экземпляров;
* Возможность создания объектов классов, производных от Singleton.

**Недостатки:**

В случае использования нескольких взаимозависимых одиночек их реализация

29. В чем заключаются основные принципы объектно-ориентированного подхода?

30. Опишите отношения агрегации, обобщения, композиции, используемые на диаграмме классов.

31. Что такое рефакторинг? Дайте определение.

Рефакторинг кода — это улучшение внутренней структуры программы таким образом, чтобы ее внешний вид, функциональность и производительность не изменились.

32. Назовите преимущества и недостатки паттерна «Адаптер».

**Преимущества:**

Отделяет и скрывает от клиента подробности преобразования различных интерфейсов.

**Недостатки:**

Усложняет код программы из-за введения дополнительных классов.

33. Что такое тестирование? Дайте определение.

Тестирование ПО - это процесс проверки готовой программы в статике (просмотры, инспекции, отладки исходного кода) и в динамике путем прогона конечного набора тестовых данных, проверяющих разные пути выполнения программы и сравнении полученных результатов с заранее запланированными.

34. Что такое «управление конфигурацией ПО»? Дайте определение.

35. Назовите преимущества и недостатки паттерна «Мост».

**Преимущества:**

Позволяет строить платформо-независимые программы.

Скрывает лишние или опасные детали реализации от клиентского кода.

Реализует *принцип открытости/закрытости*.

**Недостатки:**

Усложняет код программы из-за введения дополнительных классов.

36. Опишите типовой цикл тестирования.

37. Перечислите и опишите уровни тестирования.

• **тестирование отдельных элементов**, которое заключается в проверке отдельных, изолированных и независимых частей ПО;  
• **интеграционное тестирование**, которое ориентировано на проверку связей и  
способов взаимодействия (интерфейсов) компонентов друг с другом, включая компоненты, расположенные на разных архитектурных платформах распределенной среды;  
• **тестирование системы** предназначено для проверки правильности  
функционирования системы в целом, с обнаружением отказов и дефектов в системе и их устранение. При этом контролируется выполнение сформулированных нефункциональных требований (безопасность, надежность и др.) в системе, правильность задания и выполнения внешних интерфейсов системы со средой окружения и др

38. Назовите преимущества и недостатки паттерна «Компоновщик».

**Преимущества**

Упрощает архитектуру клиента при работе со сложным деревом компонентов.

Облегчает добавление новых видов компонентов.

**Недостатки**

Создаёт слишком общий дизайн классов.

39. Какие типы тестов вы знаете? Перечислите и опишите их.

40. Опишите «парное программирование».

Коллективное владение означает, что **каждый член команды несёт ответственность за весь исходный код**. Таким образом, каждый вправе вносить изменения в любой участок программы. Парное программирование поддерживает эту практику: работая в разных парах, все программисты знакомятся со всеми частями кода системы.

41. Назовите преимущества и недостатки паттерна «Декоратор».

**Преимущества:**

Большая гибкость, чем у наследования.

Позволяет добавлять обязанности на лету.

Можно добавлять несколько новых обязанностей сразу.

Позволяет иметь несколько мелких объектов вместо одного объекта на все случаи жизни.

**Недостатки**

Трудно конфигурировать многократно обёрнутые объекты.

Обилие крошечных классов.

42. Перечислите и опишите основные характеристики программного модуля.

размер модуля,  
прочность модуля,  
сцепление с другими модулями,  
рутинность модуля (независимость от предыстории обращений к нему).

43. Назовите преимущества и недостатки паттерна «Фасад».

**Преимущества:**

Изолирует клиентов от компонентов сложной подсистемы.

**Недостатки**

Фасад рискует стать [божественным объектом](https://refactoring.guru/ru/antipatterns/god-object), привязанным ко всем классам программы.

44. Назовите преимущества и недостатки паттерна «Легковес».

**Преимущества:**

Экономит оперативную память.

**Недостатки:**

Расходует процессорное время на поиск/вычисление контекста.

Усложняет код программы из-за введения множества дополнительных классов.

45. Что такое экстремальное программирование? Дайте определение

**Экстремальное программирование** (XP) – это упрощенная методология организации разработки программ для небольших и средних по размеру команд разработчиков, занимающихся созданием программного продукта в условиях неясных или быстро меняющихся требований.

Основными целями XP являются повышение доверия заказчика

46. Назовите преимущества и недостатки паттерна «Заместитель».

**Преимущества:**

Позволяет контролировать сервисный объект незаметно для клиента.

Может работать, даже если сервисный объект ещё не создан.

Может контролировать жизненный цикл служебного объекта.

**Недостатки:**

Усложняет код программы из-за введения дополнительных классов.

Увеличивает время отклика от сервиса.

47. Что такое непрерывная интеграция? Дайте определение.

48. Что такое технология DotNet? Дайте определение.

Платформа . NET Framework — это технология, которая поддерживает создание и выполнение веб-служб и приложений Windows. При разработке платформы . NET Framework учитывались следующие цели.

49. Назовите преимущества и недостатки паттерна «Цепочка обязанностей».

**Преимущества:**

Уменьшает зависимость между клиентом и обработчиками.

Реализует *принцип единственной обязанности*.

Реализует *принцип открытости/закрытости*.

**Недостатки:**

Запрос может остаться никем не обработанным.

50. Что такое ASP.Net. Опишите его архитектуру.

ASP.NET (Active Server Pages для . NET) — **платформа разработки веб-приложений, в состав которой входят: веб-сервисы, программная инфраструктура, модель программирования, от компании Майкрософт**. ASP.NET входит в состав платформы . NET Framework и является развитием более старой технологии Microsoft ASP.

51. Что такое реинженерия? Дайте определение.

Реинженерия (reengineering) - это эволюция программы (системы) путем ее изменения в целях повышения удобства ее эксплуатации, сопровождения или изменения ее функций.

52. Назовите преимущества и недостатки паттерна «Команда».

**Преимущества**

Убирает прямую зависимость между объектами, вызывающими операции, и объектами, которые их непосредственно выполняют.

Позволяет реализовать простую отмену и повтор операций.

Позволяет реализовать отложенный запуск операций.

Позволяет собирать сложные команды из простых.

Реализует *принцип открытости/закрытости*.

**Недостатки**

Усложняет код программы из-за введения множества дополнительных классов.

53. Опишите процесс документирования ПО.

54. Назовите преимущества и недостатки паттерна «Итератор».

**Преимущества**

Упрощает классы хранения данных.

Позволяет реализовать различные способы обхода структуры данных.

Позволяет одновременно перемещаться по структуре данных в разные стороны.

**Недостатки**

Не оправдан, если можно обойтись простым циклом.

55. Опишите критерии качества ПО.

**Защищенность** - способность компонентов программного средства защищать программы и информацию от любых негативных воздействий.

**Надежность** - обеспечение комплексом программ достаточно низкой вероятности отказа в процессе функционирования программного средства в реальном времени.

**Эффективность** - свойства программного средства, обеспечивающие требуемую производительность решения функциональных задач, с учетом количества используемых вычислительных ресурсов в установленных условиях.

**Практичность (применимость**) - свойства программного средства,  
обусловливающие сложность его понимания, изучения и использования, а также привлекательность для квалифицированных пользователей при применении в указанных условиях.

**Сопровождаемость** - приспособленность программного средства к модификации и изменению конфигурации и функций.

**Мобильность** - подготовленность программного средства к переносу из одной аппаратно-операционной среды в другую.

**Функциональные возможности** - способность программного средства  
обеспечивать решение задач, удовлетворяющих сформулированные потребности заказчиков и пользователей при применении комплекса программ в заданных условиях.

**Функциональная пригодность** - набор и описания субхарактеристики и ее атрибутов, определяющие назначение, номенклатуру, основные, необходимые и достаточные функции программного средства, соответствующие техническому заданию и спецификациям требований заказчика или потенциального пользователя.

**Правильность (корректность**) - способность программного средства обеспечивать правильные или приемлемые для пользователя результаты и внешние эффекты.

**Способность к взаимодействию** - свойство программных средств и их компонентов взаимодействовать с одной или большим числом компонентов внутренней и внешней среды.

56. Перечислите и опишите факторы, определяющие надежность системы.

57. Назовите преимущества и недостатки паттерна «Посредник».

**Преимущества**

Устраняет зависимости между компонентами, позволяя повторно их использовать.

Упрощает взаимодействие между компонентами.

Централизует управление в одном месте.

**Недостатки**

Посредник может [сильно раздуться](https://refactoring.guru/ru/smells/large-class)

58. Перечислите методы и средства разработки ПО.

59. Назовите преимущества и недостатки паттерна «Снимок».

**Преимущества**

Не нарушает инкапсуляции исходного объекта.

Упрощает структуру исходного объекта. Ему не нужно хранить историю версий своего состояния

**Недостатки**

Требует много памяти, если клиенты слишком часто создают снимки.

Может повлечь дополнительные издержки памяти, если объекты, хранящие историю, не освобождают ресурсы, занятые устаревшими снимками.

В некоторых языках (например, PHP, Python, JavaScript) сложно гарантировать, чтобы только исходный объект имел доступ к состоянию снимка.

60. Определите функциональные и нефункциональные требования.

Ф у н к ц и о н а л ь н ы е т р е б о в а н и я определяют действия, которые должна выполнять система, без учета ограничений, связанных с ее реализацией. Тем самым функциональные требования определяют поведение системы в процессе обработки информации.  
Н е ф у н к ц и он а л ь н ы е т р е б о в а н и я не определяют поведение системы, но описывают ее атрибуты или атрибуты системного окружения.

61. Назовите преимущества и недостатки паттерна «Наблюдатель».

**Преимущества:**

Издатели не зависят от конкретных классов подписчиков и наоборот.

Вы можете подписывать и отписывать получателей на лету.

Реализует *принцип открытости/закрытости*.

**Недостатки:**

Подписчики оповещаются в случайном порядке.

62. Назовите преимущества и недостатки паттерна «Состояние».

**Преимущества:**

Избавляет от множества больших условных операторов машины состояний.

Концентрирует в одном месте код, связанный с определённым состоянием.

Упрощает код контекста.

**Недостатки:**

Может неоправданно усложнить код, если состояний мало и они редко меняются.

63 Какие методы применяются на всех уровнях тестирования? Перечислите и опишите их.

На всех уровнях тестирования применяются методы:  
• функционального тестирования, обеспечивающие проверку реализации функций, которые определены в требованиях, а также правильность их выполнения;  
• регрессионного тестирования, ориентированные на повторное выборочное  
тестирование системы или ее компонентов после внесения в них изменений на тех же тестах, что и до модификации;  
• тестирования эффективности, проверяющие производительность, пропускную способность, максимальный объем данных и системные ограничения в соответствии со спецификациями требований;  
• стресс-тестирования, проверяющие поведение системы при максимально  
допустимой нагрузке или при ее превышении;  
• альфа и бета тестирования, выполняющие внутреннее тестирование кодов системы и внешнее тестирование интерфейсов. Альфа - это внутреннее тестирование (функций и алгоритмов), а бета – внешнее (взаимосвязей с другими системами и средой)  
• конфигурационного тестирования, проверяющие структуры и идентификации системы на различных наборах данных, а также работу системы на различных конфигурациях аппаратуры и оборудования.

64. Назовите преимущества и недостатки паттерна «Стратегия».

**Преимущества**

Горячая замена алгоритмов на лету.

Изолирует код и данные алгоритмов от остальных классов.

Уход от наследования к делегированию.

Реализует *принцип открытости/закрытости*.

**Недостатки**

Усложняет программу за счёт дополнительных классов.

Клиент должен знать, в чём состоит разница между стратегиями, чтобы выбрать подходящую.

65. Опишите цели и задачи этапа сопровождения ПО.

66. Назовите преимущества и недостатки паттерна «Шаблонный метод».

**Преимущества**

Облегчает повторное использование кода.

**Недостатки**

Вы жёстко ограничены скелетом существующего алгоритма.

Вы можете нарушить *принцип подстановки Барбары Лисков*, изменяя базовое поведение одного из шагов алгоритма через подкласс.

С ростом количества шагов шаблонный метод становится слишком сложно поддерживать.

67. Опишите оценку трудоемкости разработки ПО.

68. Перечислите десять советов, как получить точную оценку затрат.

1Не следует умышленно ее занижать. Заведомо нереальные показатели итоге обойдутся дороже, чем завышенная оценка.  
2 Не следует указывать "процент уверенности", если у вас для этого нет количественной базы.  
3 Не занижайте оценки, данные разработчиками, поскольку, скорее всего, эти показатели и так чересчур оптимистичны.  
4 Попытайтесь оценить размер создаваемого программного обеспечения (в строках кода). Впервую очередь именно этот показатель позволяет определить трудозатраты и установить сроки выполнения проекта.  
5 Не следует рассчитывать на то, что трудозатраты будут расти линейно при увеличении масштабов проекта. Скорее они будут увеличиваться экспоненциально.  
6 Используйте уже имеющиеся данные как основу для оценки производительности.  
7 Используйте сведения, касающиеся уже выполненного объема работ, для получения более точных оценок затрат по оставшейся части проекта.  
8 Оценки на уровне задач должны делать те специалисты, которые будут выполнять эту работу.  
9 Сравнивайте реальную производительность с предполагаемой для того, чтобы со временем уточнять оценки производительности каждого отдельного участника проекта.  
10 Не следует сокращать сроки реализации, если вы одновременно не предполагаете увеличить трудозатраты.

69. Назовите преимущества и недостатки паттерна «Посетитель».

**Преимущества:**

Упрощает добавление операций, работающих со сложными структурами объектов.

Объединяет родственные операции в одном классе.

Посетитель может накапливать состояние при обходе структуры элементов.

**Недостатки:**

Паттерн не оправдан, если иерархия элементов часто меняется.

Может привести к нарушению инкапсуляции элементов.

70. Опишите структуру ЖЦ ПО. (50/50)

В состав жизненного цикла ПО обычно включаются следующие  
стадии:  
1. Формирование требований к ПО.  
2. Проектирование.  
3. Реализация.  
4. Тестирование.  
5. Ввод в действие.  
6. Эксплуатация и сопровождение.  
7. Снятие с эксплуатации

71. Назовите этапы ЖЦ ПО, опишите, что осуществляется на каждом этапе и что является результатом каждого этапа.

72. Опишите метод моделирования данных в нотации Баркера (выделение сущностей, определение связей).

73. Опишите понятия «класс», «объект», «экземпляр класса».

74. Назовите виды отношений между классами. Опишите их назначение.

75. Дайте определение понятию «Диаграмма кооперации».

76. Опишите производящие шаблоны

**Порождающие шаблоны проектирования** предназначены для создания объектов, позволяя системе оставаться независимой как от самого процесса порождения, так и от типов порождаемых объектов. Прежде чем рассматривать особенности каждого из порождающих шаблонов, рассмотрим на примере типичные проблемы, встающие перед разработчиками при порождении в системе объектов новых типов.

77. Опишите поведенческие шаблоны.

Поведенческие шаблоны (англ. behavioral patterns) — шаблоны проектирования, определяющие алгоритмы и способы реализации взаимодействия различных объектов и классов.

78. Опишите отношения агрегации, обобщения, композиции.

79. Перечислите и опишите принципы проектирования интерфейса пользователя.

80. Что такое RAD (Rapid Application Development)?

Одним из возможных подходов к разработке ПО в рамках спиральной модели ЖЦ является получившая в последнее время широкое распространение методология быстрой разработки приложений RAD (Rapid Application Development). Под этим термином обычно понимается процесс разработки ПО, содержащий 3 элемента:  
• небольшую команду программистов (от 2 до 10 человек);  
• короткий, но тщательно проработанный производственный график (от 2 до 6 мес.);  
• повторяющийся цикл, при котором разработчики, по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные через взаимодействие с заказчиком.

81. Опишите, в каких случаях применяется RAD.

Применение технологии RAD целесообразно, когда:  
• требуется выполнение проекта в сжатые сроки (90 дней). Быстрое выполнение проекта позволяет создать систему, отвечающую требованиям сегодняшнего дня. Если система проектируется долго, то весьма высока вероятность, что за это время существенно изменятся  
фундаментальные положения, регламентирующие деятельность организации, то есть, система морально устареет еще до завершения ее проектирования.

• нечетко определены требования к ПО. В большинстве случаев заказчик весьма приблизительно представляет себе работу будущего программного продукта и не может четко сформулировать все требования к ПО. Требования могут быть вообще не определены к началу проекта либо могут изменяться по ходу его выполнения.

• проект выполняется в условиях ограниченности бюджета. Разработка ведется небольшими RAD группами в короткие сроки, что обеспечивает минимум трудозатрат и позволяет вписаться в бюджетные ограничения.

• интерфейс пользователя (GUI) есть главный фактор. Нет смысла заставлять пользователя рисовать картинки. RAD технология дает возможность продемонстрировать интерфейс в прототипе, причем достаточно скоро после начала проекта.

• проект большой, но поддается разделению на более мелкие функциональные компоненты. Если предполагаемая система велика, необходимо, чтобы ее можно было разбить на мелкие части, каждая из которых обладает четкой функциональностью. Они могут выпускаться последовательно или параллельно (в последнем случае привлекается несколько RAD групп).  
• ПО не обладает большой вычислительной сложностью.  
RAD-технология не является универсальной, то есть ее применение целесообразно не всегда. Например, в проектах, где требования к программному продукту четко определены и не должны меняться, вовлечение заказчика в процесс разработки не требуется и более  
эффективной может быть иерархическая разработка (каскадный метод). То же касается проектов, ПО, сложность которых определяется необходимостью реализации сложных алгоритмов, а роль и объем пользовательского интерфейса.

82. Опишите проблему сложности больших систем. Приведите пример использования RAD.

83. Что такое ActiveX?

ActiveX - технология Microsoft, предназначенная для написания сетевых приложений. Она предоставляет программистам наборы стандартных библиотек, значительно облегчающих процесс кодирования. Если раньше при написании программ использовались механизмы OLE (OLE Automation, OLE Documents, OLE Controls,...), основанные на компонентной объектной модели (COM - Component Object Model), то теперь библиотеки OLE переписаны так, чтобы обеспечивать функциональность, достаточную для написания сетевых приложений. Таким образом, теперь при написании программ используется DCOM (Distributed Component Object  
Model) - распределенная компонентная объектная модель, а реализуют ее библиотеки ActiveX, которые по объему оказались гораздо меньше, чем библиотеки OLE, а по скорости - быстрее.  
Сохранилась и совместимость - любой программный компонент OLE будет работать с библиотеками ActiveX

84. Опишите преимущества использования технологии ActiveX.

Основные преимущества использования технологии ActiveX:  
• **Быстрое написание программного кода**. Программирование сетевых взаимодействий становится очень похожим на программирование для отдельного компьютера.  
• **Открытость и мобильность**. Спецификации технологии недавно были переданы в Open Group как основа открытого стандарта. Кроме того, Microsoft совместно с компаниями Metrowerks и Bristol заканчивает реализацию технологий ActiveX для платформ Macintosh и UNIX.  
• **Возможность написания приложений с использованием знакомых средств  
разработки.** Программные элементы ActiveX могут быть созданы с помощью Visual Basic, Visual C++, Borland Delphi, Borland C++, любых средств разработки на Java.  
• **Большое количество уже существующих программных элементов ActiveX,** которые бесплатно могут применяться на серверах Web и в приложениях независимых разработчиков. Кроме того почти любой программный компонент OLE совместим с технологиями ActiveX и может применяться без модификаций в сетевых приложениях.  
• **Стандартность.** Технология ActiveX основана на широко используемых стандартах Internet (TCP/IP, HTML, Java) с одной стороны и стандартах, введенных в свое время Microsoft и необходимых для сохранения совместимости (COM, OLE).

85. Опишите фазы ЖЦ ПО по методологии RAD.

Жизненный цикл ПО по методологии RAD состоит из четырех фаз:  
• фаза анализа и планирования требований;  
• фаза проектирования;  
• фаза построения;  
• фаза внедрения.

86. Приведите пример 3-х модулей диаграмме компонентов по теме «Работа лифта».

87. Дайте определение понятию COM (Component Object Model).

COM (Component Object Model) - это объектная модель компонентов. Данная технология является базовой для технологий ActiveX и OLE. Технологии OLE и ActiveX - всего лишь надстройки над данной технологией. В качестве примера можно привести объект TObject, как базовый объект VCL Delphi. Точно так же технология СОМ является базовой по отношению к OLE и ActiveX.

88. Назовите виды отношений между классами.

Классы, подобно объектам, не существуют в изоляции. Напротив, с отдельной проблемной областью связывают ключевые абстракции, отношения между которыми формируют структуру из классов системы.  
Всего существует четыре основных вида отношений между классами:  
1. ассоциация (фиксирует структурные отношения — связи между экземплярами классов);  
2. зависимость (отображает влияние одного класса на другой класс);  
3. обобщение-специализация («is а»-отношение);  
4. целое-часть («part of»-отношение).  
Для покрытия основных отношений большинство объектно ориентированных языков программирования поддерживает следующие отношения:  
1) ассоциация;  
2) наследование;  
3) агрегация;  
4) зависимость;  
5) конкретизация;  
6) метакласс;  
7) реализаци

89. Перечислите основные приемы и принципы экстремального программирования. Приведите пример класса «окно».

90. Что означает «коллективное владение кодом»? (google)

Коллективное владение означает, что **каждый член команды несёт ответственность за весь исходный код**. Таким образом, каждый вправе вносить изменения в любой участок программы. Парное программирование поддерживает эту практику: работая в разных парах, все программисты знакомятся со всеми частями кода системы.

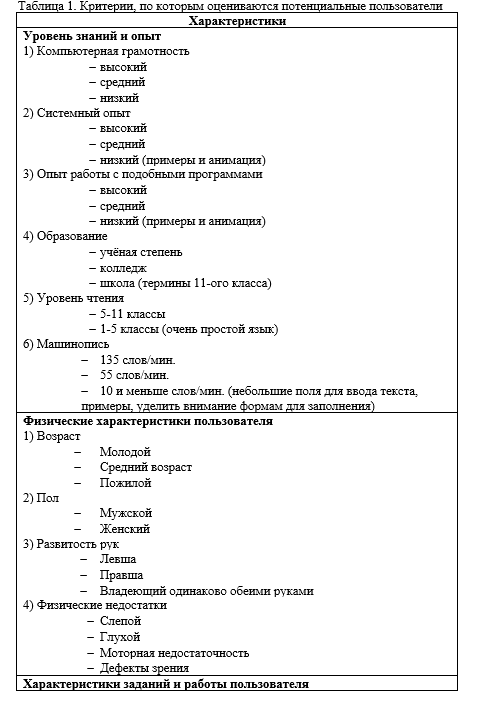
91. Дайте определение понятиям «фукциональное моделирование», «бизнес-процесс», «процесс».

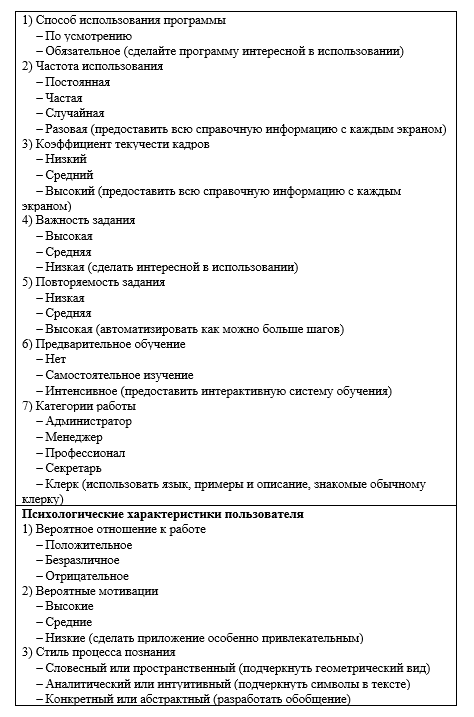
92. Перечислите и опишите шаги разработки пользовательских интерфейсов. Приведите пример коллективного владения кодом.

93. Что такое COM-интерфейс?

СОМ-интерфейс. Клиенты СОМ связываются с объектами при помощи СОМ-интерфейсов. Интерфейсы -- это группы логически или семантически связанных процедур,  
которые обеспечивают связь между поставщиком услуги (сервером) и его клиентом. На рисунке 1 схематично изображен стандартный СОМ-интерфейс.

94. Опишите критерии, по которым оцениваются потенциальные пользователи программы. Дайте описание графического представления класса на диаграмме классов.





95. Что такое COM-классы?

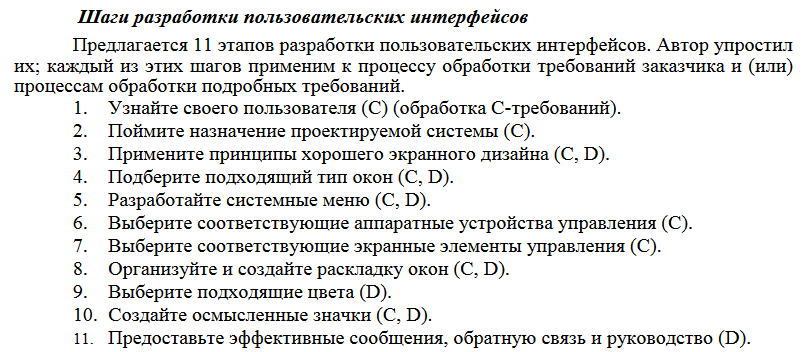
СОМ-классы (coclass) - это классы, которые содержат один или более СОМ-интерфейс. Вы можете не обращаться к СОМ-интерфейсу непосредственно, а получать доступ к СОМ интерфейсу через со-класс. Со-классы идентифицируются при помощи идентификаторов класса  
(CLSID).

96. Кто является «заказчиком» в XP?

97. Что такое модульное программирование?

Модульное программирование — это процесс разделения компьютерной программы на отдельные подпрограммы. Модуль — это отдельный программный компонент. Он часто может использоваться в различных приложениях и функциях с другими компонентами системы.

98. Опишите принципы проектирования пользовательского интерфейса.



99. Приведите пример использования экстремального программирования.

100. Опишите технологию DCOM.

Технология DCOM - это распределенная СОМ-технология. Она применяется для предоставления средств доступа к СОМ-объектам, расположенным на других компьютерах в сети (в том числе и сети Internet)

101. Что такое рефакторинг?

Рефакторинг кода — это улучшение внутренней структуры программы таким образом, чтобы ее внешний вид, функциональность и производительность не изменились.

102. Назовите показатели качества ПО и критерии оценки качества программы. Приведите пример рефакторинга кода.

103. Опишите технологию и структуру платформы dot.Net, перечислите и охарактеризуйте средства разработки приложений.