1. **Цели и задачи тестирования, что входит в задачи тестировщика:**
   * *Цели:* Обеспечение качества продукта, выявление дефектов, уверенность в соответствии продукта требованиям.
   * *Задачи тестировщика:* Планирование тестирования, создание тест-кейсов, выполнение тестов, отслеживание дефектов, обеспечение документации.
2. **Тестирование пользовательского интерфейса, ручное тестирование интерфейса, плюсы и минусы:**
   * *Плюсы:* Выявление нюансов, легкость восприятия, тестирование реального пользовательского опыта.
   * *Минусы:* Трудоемкость, низкая масштабируемость, подверженность человеческим ошибкам.
3. **Тестирование пользовательского интерфейса, автоматическое тестирование интерфейса, плюсы и минусы:**
   * *Плюсы:* Высокая скорость выполнения, повторяемость, автоматизация монотонных задач.
   * *Минусы:* Трудность в распознавании некоторых аспектов пользовательского опыта, сложность поддержки, высокие затраты на начальную настройку.
4. **Участники процесса тестирования, их назначение:**
   * *Участники:* Тестировщики, разработчики, менеджеры проекта, заказчики.
   * *Назначение:* Выявление и исправление дефектов, обеспечение соответствия продукта требованиям, контроль качества.
5. **Жизненный цикл ПО. Этапы:**
   * *Этапы:* Планирование, анализ, проектирование, реализация, тестирование, внедрение, сопровождение.
6. **Интеграционное тестирование, достоинства и недостатки:**
   * *Достоинства:* Выявление проблем взаимодействия компонентов, повышение надежности системы.
   * *Недостатки:* Трудность в выявлении всех возможных взаимодействий, высокая стоимость.
7. **Модульное тестирование, достоинства и недостатки:**
   * *Достоинства:* Выявление дефектов на ранних этапах, легкая локализация и исправление ошибок.
   * *Недостатки:* Не всегда отражает взаимодействие с другими модулями, может упустить интеграционные проблемы.
8. **Системное тестирование, достоинства и недостатки:**
   * *Достоинства:* Проверка системы в целом, выявление проблем в функциональности и производительности.
   * *Недостатки:* Высокая стоимость, сложность в репликации реальных условий использования.
9. **Функциональное тестирование, дать описание:**
   * *Описание:* Проверка соответствия функционала продукта заявленным требованиям.
10. **Инсталляционное тестирование, на что оно направлено:**
    * *Направленность:* Проверка корректности установки и деинсталляции продукта.
11. **Тестирование безопасности, на что оно направлено:**
    * *Направленность:* Выявление уязвимостей и обеспечение защиты от несанкционированного доступа.
12. **Тестирование удобства использования, на что оно направлено:**
    * *Направленность:* Оценка удобства, эргономики и пользовательского опыта продукта.
13. **Что такое требование, источники требований, методы выявления требований:**
    * *Требование:* Описание функциональных и нефункциональных характеристик продукта.
    * *Источники:* Документация, бизнес-аналитики, заказчики.
    * *Методы:* Интервью, анализ документации, прототипирование.
14. **Тестовая стратегия, что включает в себя тестовая стратегия:**
    * *Включает:* Цели тестирования, охват тестирования, выбор методов и инструментов тестирования.
15. **План тестирования, цели плана тестирования:**
    * *Цели:* Организация и планирование процесса тестирования, определение ресурсов и расписания.
16. **План тестирования, критерии начала и окончания тестирования:**
    * *Критерии:* Начало - завершение всех этапов тестирования, достижение заранее определенного уровня качества.
17. **Тест-дизайн, цели и задачи тест-дизайна:**
    * *Цели:* Создание эффективных тестов, охватывающих различные аспекты продукта.
    * *Задачи:* Выбор тестовых случаев, определение порядка выполнения тестов.
18. **Тест-кейс, атрибуты тест-кейса, плюсы и минусы тест-кейса:**
    * *Атрибуты:* Шаги теста, ожидаемые результаты, приоритет, статус выполнения.
    * *Плюсы:* Структурированное тестирование, повторяемость.
    * *Минусы:* Трудность в обновлении при изменениях, не всегда охватывает все сценарии использования.
19. **Приоритет и Важность, градации важности дефекта (TestPlan):**
    * *Приоритет:* Определение порядка исправления дефектов.
    * *Важность:* Оценка влияния дефекта на функциональность и критичность продукта.
    * *Градации:* Низкая, средняя, высокая.
20. **Модульное тестирование, цели модульного тестирования, драйвер, заглушка:**
    * *Цели:* Проверка отдельных модулей на корректность их работы.
    * *Драйвер:* Модуль, вызывающий тестируемый модуль.
    * *Заглушка:* Модуль, заменяющий вызываемый модуль и имитирующий его функциональность.
21. **Системное тестирование, объекты выявления при системном тестировании:**
    * *Объекты:* Недостатки в функциональности, производительности, надежности, безопасности, совместимости и удобстве использования.
22. **Интеграционное тестирование, возможные ошибки при интеграции, объекты интеграционного тестирования:**
    * *Ошибки:* Несовместимость интерфейсов, неправильное взаимодействие компонентов.
    * *Объекты:* Интеграция между модулями, системами или компонентами.
23. **Что такое тестирование, что такое тест, тестовый случай:**
    * *Тестирование:* Это процесс проверки программного продукта с целью выявления дефектов и уверенности в его соответствии требованиям.
    * *Тест:* Это конкретный сценарий или набор действий, направленных на проверку определенной функциональности или аспекта продукта.
    * *Тестовый случай:* Это документ, описывающий шаги выполнения теста, ожидаемые результаты и другие детали тестирования.
24. **Тестовое покрытие, что входит в состав тестового набора, критерии покрытия кода:**
    * *В состав тестового набора входят:* Тест-кейсы, сценарии тестирования, проверки и анализ результатов тестирования.
    * *Критерии покрытия кода:* Строковое покрытие, покрытие функций, ветвей, условий и путей выполнения.
25. **Ошибка, дефект, отказ. Что это такое, в чем отличия:**
    * *Ошибка:* Ошибка в программе, допущенная человеком.
    * *Дефект:* Недостаток или несоответствие требованиям в программе, вызванное ошибкой.
    * *Отказ:* Невозможность продукта выполнять требуемые функции из-за дефекта.
26. **Тестирование пользовательского интерфейса, задачи тестирования пользовательского интерфейса:**
    * *Задачи:* Проверка навигации, визуального дизайна, реакции на ввод пользователя, совместимости с различными устройствами и браузерами.
27. **Верификация и валидация, что это такое, в чем отличия:**
    * *Верификация:* Проверка соответствия продукта заданным спецификациям и требованиям.
    * *Валидация:* Проверка соответствия продукта ожиданиям и потребностям пользователя.
    * *Отличия:* Верификация - "делаем правильную вещь", валидация - "делаем вещь правильно".
28. **Тестирование пользовательского интерфейса, виды тестирования пользовательского интерфейса:**
    * *Виды:* Функциональное тестирование, тестирование удобства использования, совместимость с разными устройствами и браузерами, доступность.
29. **Классы эквивалентности. Анализ граничных значений:**
    * *Классы эквивалентности:* Группировка входных данных на классы, внутри которых ожидается одинаковое поведение системы.
    * *Анализ граничных значений:* Тестирование значений, близких к границам классов эквивалентности, так как здесь часто происходят ошибки.
30. **Матрица принятия решений:**
    * *Матрица:* Инструмент для анализа тестовых случаев в зависимости от различных критериев, таких как приоритет, сложность и т.д.
31. **Unit тест. Типы тестирования:**
    * *Unit тест:* Тестирование отдельных модулей (или компонентов) программы.
    * *Типы тестирования:* Позитивное, негативное, проверка граничных значений.
32. **Свойства хорошего Unit Test. Принцип именования Unit Test:**
    * *Свойства:* Независимость, повторяемость, автоматизируемость.
    * *Принцип именования:* Использование ясных и описательных имен для тестов, отражающих их назначение.
33. **Когда не нужно создавать Unit Test. Когда нужно создавать Unit Test:**
    * *Не нужно:* Если функциональность меняется часто и трудно поддается тестированию.
    * *Нужно:* При стабильной функциональности, которую можно автоматизировать.
34. **Шаблон AAA. Методы класса Assert:**
    * *Шаблон AAA:* Arrange (подготовка), Act (действие), Assert (утверждение).
    * *Методы Assert:* Проверка ожидаемых результатов теста.
35. **Test Driven Development. Методы класса StringAssert:**
    * *Test Driven Development:* Методология, при которой сначала пишутся тесты, а затем код, чтобы они проходили.
    * *Методы StringAssert:* Проверка строковых значений в тестах.
36. **Именование методов и классов тестирования. Методы класса CollectionAssert:**
    * *Именование:* Использование понятных и описательных названий методов и классов.
    * *Методы CollectionAssert:* Проверка коллекций на равенство, подмножество и т.д.
37. **Тесты состояния. Тесты взаимодействия:**
    * *Тесты состояния:* Проверка поведения системы в различных состояниях.
    * *Тесты взаимодействия:* Проверка корректности взаимодействия компонентов.
38. **Внешняя зависимость. Виды тестовых объектов:**
    * *Внешняя зависимость:* Зависимость от внешних ресурсов или условий.
    * *Тестовые объекты:* Зависимые и независимые объекты, влияющие на результаты тестов.
39. **Нагрузочное тестирование. Стресс-тестирование:**
    * *Нагрузочное тестирование:* Проверка стойкости системы при увеличении нагрузки.
    * *Стресс-тестирование:* Проверка стойкости системы при крайних условиях.
40. **Apache JMeter. Объект Assertion:**
    * *Apache JMeter:* Инструмент для проведения нагрузочного и стресс-тестирования.
    * *Объект Assertion:* Компонент JMeter для проверки результатов теста и установки критериев успешности.
41. **Apache JMeter. Объект Listener:**
    * *Объект Listener:* Компонент JMeter для отображения результатов тестирования в различных форматах (графики, таблицы и т.д.).
42. **Apache JMeter. Объект Samplers:**
    * *Объект Samplers:* Компонент JMeter, определяющий запросы к серверу в ходе тестирования.