**Junior**

**Теория тестирования**

1. Что такое тестирование?

* Сопоставление реального и ожидаемого поведения программы/системы в смоделированных условиях на основании требований предъявленных к данной системе. Проверка насколько фактическое соответствует ожидаемому результату.

2. Зачем тестировать ПО?

* Для проверки соответствия требованиям и своевременного выявления имеющихся или возможных багов (отклонение фактического от ожидаемого).

3. Какие существуют этапы тестирования?

* Планирование
* Мониторинг фактических с ожидаемым результатами
* Анализ тестирования
* Проектирование тестов
* Реализация тестов
* Выполнение тестов
* Завершение тестирования

4. Какие типы тестирования можете назвать?

5. Какие уровни тестирования знаете?

6. Какие техники тест-дизайна знаете?

7. Что такое техника анализа классов эквивалентности?

8. Что такое техника анализа предельных значений? В чем ценность этой техники?

9. Что такое Regression и Confirmation тестирование, какая между ними разница?

10. Как часто следует проводить регрессионное тестирование продукта?

11. Какие бывают виды интеграционного тестирования?

12. Что такое Configuration Testing?

13. Что такое Exploratory Testing?

14. Какие существуют UI-стандарты?

15. Что такое Black/Grey/White Box Testing?

16. Что такое Performance Testing?

17. Что такое Smoke и Sanity тестирование и какая между ними разница?

18. Что такое Traceability Matrix?

19. Что такое Sanity Testing?

20. Что такое End-to-End тест?

21. Что такое тестирование безопасности?

22. Что такое испытание на основе рисков?

23. Что такое динамическое тестирование?

24. Что такое «парадокс пестицида»?

25. Опишите основные фазы STLC? Дайте определение Entry и Exit Criteria.

26. Что такое Bug, Error, Failure, Fault?

27. Какие есть атрибуты баг-репорта? Какие основные поля для заполнения?

28. Какова разница между приоритетом и серьезностью?

29. Приведите примеры серьезного, но не приоритетного бага.

30. В чем разница между валидацией и верификацией?

* Оба процесса используются для обнаружения дефектов, но в разных плоскостях. Верификация используется для выявления ошибок в спецификациях требований, а валидация — для поиска дефектов в реализованном ПО.
* Валидация — это то насколько продукт или система соответствует ожиданием конечного пользователя, а верификация — это то насколько она соответствует спецификациям проекта.

31. Зачем нужна тестовая документация? Какие её виды?

* Продуктная документация – план продукта (проджект менеджер план) чек-листы, тест-кейсы, архитектура и дизайн, технические спецификации.
* Проектная документация – пользовательская и сопроводительная, маркетинговая.

32. Что такое тест-план? Какие элементы у него есть?

33. Какую обязательную информацию должен содержать тест-план? Как правильно его использовать, поддерживать и нужен ли он вообще для большинства проектов?

34. Какая разница между чек-листом и тест-кейсами?

35. Приведите пример хорошего тест-кейса.

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсе [QA Стартовый](https://itvdn.com/ru/video/qastarter19) ([урок 1](https://itvdn.com/ru/video/qastarter19), [урок 3](https://itvdn.com/ru/video/qastarter19/text-doc), [урок 4](https://itvdn.com/ru/video/qastarter19/testdesign-qastarter), [урок 5](https://itvdn.com/ru/video/qastarter19/bugreport19)), [Основы тестирования](https://itvdn.com/ru/video/manual_qa) ([урок 6](https://itvdn.com/ru/video/manual_qa/testing_types)), [Основы тестирования ПО](https://itvdn.com/ru/video/testing-process) ([урок 1-5](https://itvdn.com/ru/video/testing-process)), а также в вебинаре “QA практикум. Техники тест дизайна” ([часть 1](https://itvdn.com/ru/webinars/description/qa-pracrice1) и [часть 2](https://itvdn.com/ru/webinars/description/qa-pracrice2)).

**AQA (Automation QA)**

**Программирование**

36. Что такое ООП? Назовите его принципы с примерами?

37. Что такое интерфейс? Что такое абстрактный класс? Чем они отличаются?

38. Что такое SOLID? Приведите примеры.

39. Что такое DRY, KISS, YAGNI?

40. Какие паттерны GOF вам известны? Приведите примеры их использования.

41. Что такое PageObject и PageFactory?

42. Какая иерархия Collections?

43. Какая разница между Thread class и Runnable interface?

44. Какая разница между String, Stringbuffer и Stringbuilder?

45. Разница между final, finally и finalize?

**Selenium**

46. ​​Что такое Selenium и зачем его используют?

47. Что такое драйвер браузера?

48. Какие виды локаторов страницы существуют? Каковы их преимущества и недостатки?

49. Что такое Selenium Waits? Какие есть и чем отличаются?

50. Какие exceptions может бросить Selenium? Что они означают и как их обрабатывать?

51. Для чего используют JavaScriptExecutor? Приведите примеры.

52. Что такое Selenium Grid?

53. Какие способы click и send keys Selenium?

54. Как вы запускаете параллельное выполнение тестов? Что такое ThreadLocal?

55. Какая разница между Action и Actions?

56. Как написать метод isElementPresent?

57. Как вычитать данные из динамической веб-таблицы?

58. Можете ли вы назвать 10 интерфейсов в Selenium?

59. Назовите два способа, позволяющих автоматизировать капчу.

60. Вспомните типы навигационных команд Selenium.

61. Как найти поврежденные ссылки в Selenium WebDriver?

62. Какую технику следует рассмотреть, используя весь сценарий, если «нет ни frame id, ни frame name»?

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсах [Web Testing Automation on Java](https://itvdn.com/ru/video/webtesting-automation) ([урок 1](https://itvdn.com/ru/video/webtesting-automation)), [Автоматизация тестирования мобильных приложений](https://itvdn.com/ru/video/testing-automation-mobile) ([урок 5](https://itvdn.com/ru/video/testing-automation-mobile/locators)), а также в вебинаре “[Selenoid или Selenium Grid — что лучше?](https://itvdn.com/ru/webinars/description/selenoid-vs-selenium-grid)”.

**TestNG/JUnit**

63. Для чего нужны TestNG/JUnit?

64. Какие инструкции используются в TestNG/JUnit?

65. Какие assertions есть в TestNG/JUnit?

66. Как выполнять тесты параллельно TestNG/JUnit?

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсе [Web Testing Automation on Java](https://itvdn.com/ru/video/webtesting-automation) ([урок 4](https://itvdn.com/ru/video/webtesting-automation/maven-gradle)).

**Git**

67. Для чего используют системы контроля версий?

68. Что такое Git? Каков принцип его работы?

* это консольная утилита, для отслеживания и ведения истории изменения файлов, в вашем проекте. Чаще всего его используют для кода, но можно и для других файлов. Например, для картинок - полезно для дизайнеров.
* Распределенная система контроля версий,

69. Что такое commits, branches в Git?

* Коммиты – точки сохраненные проекта.

70. Для чего нужны GitHub, GitLab и другие, базирующиеся на Git, вебхостинги проектов?

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсе [Основы работы с Git](https://itvdn.com/ru/video/git-basics).

**CI**

71. Что такое CI?

72. Как автоматическое тестирование интегрируется в CI?

73. Как настроить Job или Pipeline на знакомом вам CI-инструменте?

74. Какие инструменты для генерации репорта после выполнения автоматических тестов вы знаете?

75. Какую информацию должен содержать отчет о выполнении автоматических тестов?

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсе [Web Testing](https://itvdn.com/ru/video/web-testing) ([урок 4](https://itvdn.com/ru/video/web-testing/ci-cd-cd)).

**Web**

76. Что такое клиент-серверная архитектура?

77. Что может выступать в роли клиента?

78. Что такое REST API, SOAP? В чем разница?

79. Какие протоколы передачи данных знаете?

80. Какие способы взаимодействия с API существуют? В чем разница между ними?

Тема связана со специальностями:

[Quality Assurance](https://itvdn.com/ru/specialities/qa)[С чего начать?](https://itvdn.com/ru/specialities/start)[Frontend Developer](https://itvdn.com/ru/specialities/frontend-developer)

81. Как можно протестировать API, что там нужно проверять?

82. Как расшифровывается CRUD?

83. Чем отличается GET от POST?

84. Какие отличия между XML и JSON?

85. Какие знаете форматы передачи данных?

86. Как происходит шифрование?

87. Какие бывают виды баз данных?

88. Охарактеризуйте каждый класс status code (1хх; 2xx; 3xx; 4xx; 5xx).

89. Какие есть HTTP-методы?

90. Какие знаете Web elements?

91. Какие браузеры знаете? В чем их отличие?

92. Для чего необходимы инструменты разработчика в браузере (Chrome DevTools) и как они помогают в тестировании.

93. Что такое кэш?

94. Что такое сессия?

95. Зачем нужны cookies?

96. Что такое фрейм?

97. Что такое HTML/CSS/JavaScript?

98. Какую структуру имеет веб-страница?

99. Зачем чистить кэш?

100. Какие виды тестирования можно применить только к Web?

101. Для чего в веб-страницах используют JavaScript?

102. Что такое REST?

103. Что такое AJAX?

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсах [QA Стартовый](https://itvdn.com/ru/video/qastarter19) ([урок 6](https://itvdn.com/ru/video/qastarter19/http-request-http-response)), [Web Testing](https://itvdn.com/ru/video/web-testing), [SQL Базовый](https://itvdn.com/ru/video/sql-essential).

**Mobile**

104. Какие мобильные платформы существуют?

105. Какие версии Android и iOS используются на рынке (минимальные и максимальные)?

106. Какие версии Android нужно тестировать, если заказчик сказал поддерживать с версии 5.0?

107. Назовите типы мобильных приложений.

108. Каков формат файлов сборок приложений для Android и iOS?

109. Что такое ADB?

110. Как снять логи с AOS/IOS?

111. Что нужно проверять при использовании сканера отпечатка/Face ID?

112. Как я могу запускать тесты Android без Appium?

113. Объясните концепцию дизайна Appium.

**Ответы**на некоторые из этих вопросов вы можете найти в видео курсе [Автоматизация тестирования мобильных приложений](https://itvdn.com/ru/video/testing-automation-mobile).

**Практические задания**

114. Написать чеклист для функционала корзины в интернет-магазине.

115. Написать тестовые наборы данных для поля ввода даты, которое отсеивает пользователей в возрасте до 18 лет.

116. Написать чеклист тестирования формы ввода данных платежной карты.

117. Протестовать «предмет» относительно различных видов тестирования. (Предмет - лифт, карандаш, калькулятор и т. д.)

118. Есть Input поле, принимающее целые значения от 18 до 99 включительно. Надо протестировать с помощью техники тест-дизайна Boundary Values ​​Analysis и Equivalence Partitioning.

119. Есть веб-страница с полями: e-mail, password и кнопкой submit. Необходимо привести примеры отрицательных тест-кейсов, которыми можно проверить эту страницу.

120. Привести примеры тест-кейсов для функционала, находящегося на нескольких страницах проекта (например, поле поиска).

121. Как протестировать процесс оплаты в интернет-магазине?

122. Как протестировать сломанный тостер?

123. Объясните для 7-летнего ребенка, что такое база данных.

124. Определите необходимое количество функциональных тест-кейсов, чтобы проверить Log in форму.

125. Есть форма регистрации в веб-приложении с полями (first name, last name, username, password, repeat password) и кнопкой Register. Какие проверки нужно провести?

126. Поле username должно быть обязательным, но оно не является обязательным. Приведите пример баг-репорта, созданного для этой ошибки.

127. Как бы вы провели smoke-testing для приложения типа Telegram?

133. Как будет выглядеть баг-репорт, если, к примеру, не работает электрический чайник?

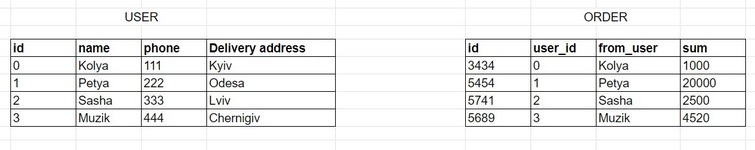
128. Есть таблица books с полями: name, price, page\_count. Следует выбрать все имена книг, в которых price более 10 единиц и количество страниц от 20 до 100.

129. У вас есть функционал калькулятора, который доступен через веб-браузер по ссылке. Он имеет только функцию делить, так сказать, MVP-версию. Диапазоны для вписывания в числитель и делитель от 0,1 до 99,9. Вывод значения происходит автоматически, потому что front-end реализован на React JS. Как вы будете тестировать этот функционал? Какие виды тестирования примените? Какие техники тест-дизайна используете?



130. Задание на работу с SQL.

* извлечь номер телефона и адрес пользователя Muzik.
* Извлечь данные о пользователях, имеющих сумму заказа более 2000 грн.
* Подсчитать количество заказов в таблице и общую сумму сделанных заказов.



131. Ваша компания разрабатывает программное обеспечение для медицинских систем, и вы тестируете компонент, управляющий дефибриллятором сердца. Вы заметили, что одно решение в тестовом модуле состоит из 34 независимых атомарных условий. Какой метод тестирования белого ящика следует выбрать для этого и почему?

132. Оздоровительная программа для сотрудников совмещена с оплатой медицинского страхования и имеет следующие правила:

* сотрудники, потребляющие 17 единиц или менее алкоголя в неделю, получают $28 скидки на оплату.
* Для сотрудников, которые заполнят «Оценку риска для здоровья», оплата уменьшается на $23.
* Сотрудники, участвующие в ежегодном контроле за состоянием здоровья в компании: получат скидку на $50 за то, что имеют индекс массы тела (ИМТ) 25,5 или менее, и $19 скидки при ИМТ ниже 30. Некурящие получают дополнительную скидку на $46. Курильщики, присоединившиеся к курсу отказа от курения, получают скидку в $24. Курильщики, не присоединившиеся к курсу отказа от курения, оплачивают дополнительно $75.

133. Используя технику классов эквивалентности, сколько тестов нужно написать, чтобы покрыть вышеупомянутые условия на 100%?

134. Какое минимальное количество тестов необходимо для покрытия следующих условий автогражданки:

* лица до 18 лет не застраховываются.
* Для мужчин на красном авто прибавляется +15% к стоимости полиса.
* Для женщин от 18 до 64 лет страховая премия 1000 грн.
* Для мужчин от 18 до 64 лет страховая премия 1200 грн.
* Для лиц старше 64 лет страховая премия 1800 грн.

135. Напишите сценарии автоматического тестирования для сортировки по цене и добавлению товара в корзину [на сайте](https://www.saucedemo.com/). К вашим тестам добавьте документацию с настройками и разместите ваше решение на GitHub.