

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

КАФЕДРА “Инфокогнитивные технологии”

Автоматизированное тестирование

ФИО студента: Переверзев Иван Дмитриевич

Группа: 221-321

Дата: 22.04.24

Номер вопроса	Ответ
1	<p>Тестирование программного обеспечения представляет собой процесс проверки программы на соответствие заданным критериям. Очень часто эти критерии представлены функциональными требованиями к разрабатываемому продукту.</p> <p>Тестирование необходимо для гарантирования того, что разрабатываемый программный продукт будет работать корректно, соответствует требованиям заказчика и обладает приемлемыми показателями производительности, надежности, безопасности и многих других характеристик.</p>
2	<p>Функциональное тестирование направлено на отладку непосредственно функционала разрабатываемого программного продукта.</p> <p>Нефункциональное тестирование направлено на проверку таких вещей как: документация, установка, производительность, безопасность и так далее.</p>
3	<p>Виды функционального тестирования:</p> <ol style="list-style-type: none">1. Unit-тестирование – тестирование на этапе написания основного кода2. Интеграционное тестирование – тестирование нескольких компонентов проекта как единое целое. Изучается их взаимодействие между собой.3. Тестирование интерфейса

	<p>4. Системное тестирование – оценка всей системы в целом:</p> <ol style="list-style-type: none"> 1. Сквозное тестирование - один большой тест, который проверяет полный процесс работы с системой, включая все подсистемы. 2. Смоук-тестирование - тестирование самых основных, критически важных функций системы. 3. Happy Path Testing - тестирование по положительному сценарию. Без проверки отрицательных или ошибочных условий. Только с ожидаемыми значениями. 4. Обезьянье Тестирование - Проверка программы на моменты, которые могут ее сломать. 5. Приемочное тестирование – тестирование с помощью бизнес-сценариев. 6. Регрессионное тестирование – тестирование неизменных функций приложения, чтобы убедиться, что программа не сломалась после обновления функционала тестирования.
4	<p>Ручное тестирование предполагает, что проверочные сценарии выполняются самим тестировщиком, который проверяет функциональность разрабатываемого продукта. Этот способ не очень эффективен и трудозатратен.</p> <p>Автоматизированное тестирование – это метод проверки ПО, который использует специализированные программные средства для ускорения процесса тестирования. Также этот способ позволяет переиспользовать тестовые сценарии и проводить более тщательное тестирование.</p>
5	<p>Тест-кейс – это документ, который описывает последовательность действий, необходимых для проведения определенного теста. Он включает в себя само описание действий, ожидаемые результаты, а также возможные варианты развития событий.</p> <p>Данный документ позволяет стандартизировать процесс тестирования.</p>
6	<p>X-path – это инструмент для поиска определенных элементов или узлов в XML-документах. Поиск производится на основе атрибутов, значений или позиции искомого элемента в документе.</p> <p>В библиотеке Selenium данная технология используется для того же – гибкого поиска нужных элементов в DOM-дереве.</p>
7	<p>Существует два вида x-path:</p> <ol style="list-style-type: none"> 1. Абсолютный – указание точного пути до нужного элемента через все dom-дерево. 2. Относительный – указание только некоторых параметров, которые помогут найти нужный элемент, например, по значению атрибутов или содержимому элемента.

	<p>Пример абсолютного пути: <code>/html/body/main/section/h2</code></p> <p>Пример относительного пути: <code>//input[@name="email"]</code></p>
8	<p>XPath для месяца: <code>//div[@class = 'Header_date_94eb6c5']//preceding-sibling::span[@class = 'Text_lg_16bflff Typography_defaultFont_16bflff Typography_semiBold_16bflff']</code></p> <p>XPath для года: <code>//div[@class = 'Header_date_94eb6c5']//preceding-sibling::span[@class = 'Header_year_94eb6c5 Text_lg_16bflff Typography_defaultFont_16bflff Typography_semiBold_16bflff']</code></p>
9	<p>POM (Page Object Model) – это шаблон проектирования, который используется для кейсов для автоматизированного тестирования. Основная цель этого шаблона – избежать дублирования и позволить переиспользование кода.</p> <p>Для реализации POM в selenium нужно сделать следующее:</p> <ol style="list-style-type: none"> 1. Создать классы-обертки (page object) для каждой веб-страницы, которые будут содержать локаторы элементов и методы для взаимодействия между ними. 2. В тестовых сценариях использовать созданные Page Objects вместо прямого взаимодействия с элементами страницы. 3. При изменении структуры или локаторов веб-страницы, необходимо будет обновить только соответствующий Page Object, не затрагивая тестовые сценарии.
10	<p>Явные и неявные ожидания в Selenium представляют собой методы синхронизации тестов с динамическими элементами веб-страниц:</p> <ul style="list-style-type: none"> • Неявное ожидание (Implicit Wait): Это метод, который позволяет WebDriver ожидать определенное время перед выбросом исключения, если элемент не найден сразу. Неявное ожидание устанавливается один раз и применяется ко всему тесту, что делает его легким в использовании. • Явное ожидание (Explicit Wait): В отличие от неявного ожидания, явное ожидание используется для конкретного поиска элемента в определенный момент времени. Этот метод позволяет задать условие, по которому WebDriver будет ждать, прежде чем продолжить выполнение теста.

