**Основы автоматизированного тестирования**

*Продолжительность курса – 21 пара*

# Версия 1.0.1

**Цель курса**

Познакомить слушателя с принципами автоматизированного тестирования. Научить проводить автоматизированное тестирование с использованием различных утилит. Научить выбирать правильные инструменты тестирования для решения практических задач.

**По окончании курса слушатель будет:**

* Понимать, что такое автоматизированное тестирование
* Разбираться в основах HTML/CSS
* Разбираться в основах XML
* Понимать архитектуру и принципы использования Selenium
* Уметь анализировать структуру веб-страниц и взаимодействовать с ними
* Уметь выбирать правильный инструмент для проведения автоматизированного тестирования
* Создавать скрипты в Selenium для тестирования веб-страниц

По окончании данного курса студент сдаёт практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену, должны быть сданы все домашние и практические задания.

*Модуль 1 (1 пара)*

# Что такое автоматизированное тестирование

1. Что такое автоматизация тестирования?
2. Почему важна автоматизация при тестировании?
3. Цели и задачи автоматизированного тестирования
4. Место автоматизированного тестирования в процессе разработки продукта
5. Этапы автоматизированного тестирования
6. Подходы к автоматизированному тестированию
7. Почему необходимо владеть основами HTML/CSS для тестировщика?

*Модуль 2 (1 пара)*

# Введение в Web-технологии, структура HTML, форматирование текста при помощи HTML

1. Введение в HTML.
2. Введение в языки разметки. Язык разметки гипертекста HTML.
   1. Развитие HTML, версии. Текущие используемые версии: HTML и XHTML.
   2. Вопросы межбраузерной совместимости. Война браузеров.
   3. W3C.
3. Теги – основной элемент структуры HTML. Правила записи тегов и их атрибутов в стандарте XHTML на примере тегов <b>, <i>, <u>, <font>, <sup>, <sub>, <br />.

Синтаксические отличия HTML4 от XHTML.

1. Основные ошибки в записях тегов.
   1. Спецификации <!DOCTYPE HTML>.
   2. Валидация документа при помощи FireFox – дополнение HTML Validator.
   3. Понятие well-formed.
   4. Прародители HTML4/5 и XHTML: SGML и XML.
2. Основная структура XHTML документа. Основные элементы и их назначение.
3. Кодировки страницы и теги <meta>:
   1. Применение тега <meta> – задание информации о странице (expires, refresh, autor, copyright, keywords, description).
   2. Задание кодировки страницы при помощи тега <meta>.
   3. Символьные подстановки и кодировки.
4. Классификация тегов: линейные и блочные.
   1. Линейные: <b>, <i>, <u>, <font>.
   2. Блочные: <p>, <h1>..<h6>.
5. Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>..<h6>.

Выравнивание текста в блочных элементах: атрибут align.

1. Классификация тегов: логическое и физическое форматирования.
   1. Теги физического форматирования: <b>, <i>.
   2. Теги логического форматирования: <strong>, <em>. Их отличие.
   3. Краткий обзор основных тегов логического форматирования: <abbr>, <acronym>, <cite>, <code>, <del>, <dfn>, <ins>.
2. Работа с изображениями и ссылками
3. Практика: создание простейшей web-страницы.

*Модуль 3 (1 пара)*

# Форматирование при помощи CSS

1. CSS – каскадные таблицы стилей.
   1. Введение. Обзор версий. Назначение: HTML служит для задания структуры, CSS – для форматирования.
   2. Встраивание CSS в HTML при помощи атрибута style. Правила записи CSS свойств.
2. Теги без форматирования <div> – блочный, <span> – линейный.
3. Аналогия HTML и CSS на примере линейных и блочных тегов:
   1. Тег <font> – свойства color, font-size, font-family.
   2. Тег <b> – свойства font-weight.
   3. Тег <i> – свойства font-style.
   4. Тег <u> – свойства text-decoration.
   5. Тег <sup>, <sub> – свойства vertical-align.
   6. Атрибут align – свойства text-align.
   7. Сокращенная запись свойства font.
   8. Дополнительные свойства CSS для форматирования текста: letter-spacing, lineheight, text-intend, text-transform, white-space, word-spacing.
4. Использование атрибутов class и id для задания стилей.
   1. Создание стилей для тегов, классов, идентификаторов внутри тега <style>. Понятие селекторов. Правило записи селекторов: селектор тегов, селектор классов, селектор идентификаторов, универсальный селектор \*.
   2. Приоритет использования стилей (tag / class / id / style). Повышение приоритета правилом !important.
   3. Наследуемость стилей. Стандартные значения свойств.
   4. Отслеживание стилей при помощи средства разработки firebug (дополнение для Firefox)
5. Использование внешних CSS файлов стилей.
   1. Подключение CSS файлов при помощи тега <link> и инструкции @import.
   2. CSS файлы и кэш браузера.
6. Практика: форматирование текста при помощи CSS.

*Модуль 4 (1 пара)*

# Таблицы

1. Создание простейшей таблицы. Теги <table>, <tr> и <td>.
   1. Атрибуты border, cellspacing cellpadding. Их возможные аналоги CSS: border, padding.
   2. Указание ширины и высоты ячейки: атрибуты width, height. Правила задания ширины и высоты. Аналоги CSS: свойства width, height.
   3. Выравнивание данных в таблице: атрибуты align и valign. Аналоги CSS: свойства text-align, vertical-align.
   4. Управление цветом фона и цветом рамок таблицы (отдельной строки, отдельной ячейки).
   5. Использование изображений в качестве фона таблицы (отдельной строки, отдельной ячейки).
2. Объединение ячеек: атрибуты colspan, rowspan.
3. Теги логического структурирования таблиц: <thead>, <tbody>, <tfoot>. Теги логического группирования столбцов: <colgroup>, <col>.
4. Управление рамками таблицы: атрибуты frame, rules.
5. Практика: создание сложных таблиц.
6. Основы табличной верстки. Пример табличной верстки: ее минусы.

*Модуль 5 (1 пара)*

# Формы, фреймы

1. Введение в формы.
2. Управляющие элементы форм.
   1. Кнопки (отправки, сброса, пр.).
   2. Флажки.
   3. Кнопки с зависимой фиксацией (радиокнопки).
   4. Всплывающие списки.
   5. Текстовый ввод.
   6. Выбор файлов.
   7. Скрытые управляющие элементы.
3. Создание форм при помощи HTML:
   1. Элемент <form>.
   2. Элемент <input>.
   3. Элемент <button>.
   4. Элементы <select>, <optgroup> и <option>.
   5. Элемент <textarea>.
   6. Метки <label>.
   7. Структура форм: <fieldset> и <legend>.
4. Форматирование элементов форм при помощи CSS.
5. Фреймы и их структура (теоретические сведения).
   1. Тег <iframe>.
   2. Использование Спецификации <!DOCTYPE HTML> для фреймов.
   3. Вредность использования фреймов.
   4. Применение тега <iframe> в визуальных редакторах WYSIWYG.

# *Модуль 6 (1 пара)*

# Введение в Selenium

1. Что такое Selenium?
2. Почему нужно использовать Selenium?
3. Архитектура Selenium
4. Что такое WebDriver?
5. Инсталляция Selenium

*Модуль 7 (1 пара)*

# Что такое тест, command

1. Что такое тест в Selenium?
2. Как записать тест?
3. Как запустить тест?
4. Что такое selenese?
5. Понятие command в Selenium
6. Типы command
7. Примеры использования command

*Модуль 8 (4 пары)*

# Верификация в Selenium, WebDriver

1. Верификация в Selenium
2. assert и verify
3. Примеры использования
4. Понятие test suite
5. Примеры использования
6. WebDriver
   1. И ещё раз о том, что такое WebDriver
   2. Цели и задачи WebDriver
   3. Настройка WebDriver
   4. Создание простого скрипта
   5. Какие бывают элементы страницы
   6. Поиск элементов
   7. Взаимодействие с элементами страницы
      * Получение значений элементов
      * Выбор элементов
      * Активизация элементов
      * Практические примеры взаимодействия с разными элементами
7. Понятие Waits
   1. Что такое Waits?
   2. Типы Waits  explicit
      * implicit
      * Примеры использования

*Модуль 9 (2 пары)*

# Введение в XML, XPath в WebDriver

1. Что такое XML?
2. История возникновения XML
3. Цели и задачи XML
4. XML документ
5. Синтаксис и структура XML-документа
6. Описание структуры XML-документа с помощью DTD
7. Пространства имен XML
8. Понятие схемы, отличия схем от DTD
9. Парсеры XML
   1. Что такое парсер?
   2. Цели и задачи парсера
   3. DOM- и SAX-парсеры
10. Примеры создания XML-документов
11. XML-документация
    1. Что такое xml-документация кода?
    2. Зачем использовать xml-документацию
    3. Примеры использования
12. XPath
    1. Что такое XPath?
    2. Цели и задачи XPath
    3. Примеры использования XPath
    4. Получение данных страницы с помощью XPath в WebDriver

*Модуль 10 (2 пары)*

# Selenium Server

1. Что такое Selenium Server?
2. Цели и задачи Selenium Server3. Что такое Selenium Grid?
3. Цели и задачи Selenium Grid
4. Настройка и запуск Selenium Server
5. Использование Selenium Server
6. Настройка и запуск Selenium Grid
7. Использование Selenium Grid

*Модуль 11 (2 пары)*

# Создание framework для тестирования в Selenium

1. Что такое framework?
2. Почему необходимо создавать framework?
3. Принципы создания framework
4. Архитектура framework
5. Примеры использования framework

*Модуль 12 (2 пары)*

# Анализ продуктов для автоматизации тестирования, автоматизированное тестирование веб-сервисов и мобильных приложений

1. Обзор различных инструментов для тестирования
   1. Sikuli
   2. TestComplete
   3. Cucumber
   4. QTP
   5. Другие продукты
2. Автоматизированное тестирование веб-сервисов
   1. Что такое веб-сервис?
   2. Цели и задачи веб-сервиса
   3. Примеры веб-сервисов
   4. Основы взаимодействия с веб-сервисом
   5. Утилиты для автоматизированного тестирования веб-сервисов

 Soap UI

* + - Apache HttpComponents
    - Другие продукты
  1. Примеры использования

1. Автоматизированное тестирование мобильных приложений
   1. Почему нужно автоматизировать тестирование мобильных приложений?
   2. Трудности при автоматизации тестирования мобильных приложений
   3. Обзор инструментов для тестирования мобильных приложений
   4. Тестирование мобильных приложений с помощью Appium Что такое Appium?
      * Цели и задачи Appium
      * Архитектура Appium
      * Примеры использования

*Модуль 13 (2 пары)* **Экзамен**