

Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий
Кафедра технологий программирования

Методические указания
к выполнению лабораторной работы №13

по дисциплине «**Надёжность программного обеспечения**»
для специальности 1-40 01 01 Программное обеспечение информационных
технологий

на тему «**Автоматизация тестирования для Web-приложений**»

Новополоцк, 2017 г.

Название: «Автоматизация тестирования для Web-приложений».

Цель работы: ознакомиться с автоматизированным тестированием Web-приложений. Научиться составлять тесты с помощью Selenium IDE.

Теоретическая часть

Много, возможно большинство, приложений сегодня пишутся как Web-приложения. Эффективность тестирования таких приложений в различных компаниях неодинаковая. В настоящее время, для многих организаций, использующих различные формы гибкой методологии разработки, автоматизация тестирования становится необходимым требованием для создания программного обеспечения.

У автоматизации тестирования есть много плюсов. Большинство из них связано с возможностью повторного использования тестов и скоростью, с которой тесты могут быть выполнены. Рынок предлагает большое количество платных и бесплатных инструментальных средств, которые облегчают работу по автоматизации тестирования. Selenium, возможно, одно из самых распространённых некоммерческих решений с открытым исходным кодом.

Автоматизация тестирования имеет определенные преимущества для повышения долгосрочной эффективности тестирования программного обеспечения. Автоматизация тестирования поддерживает:

- быструю обратную связь с разработчиками;
- практически неограниченные итерации выполнения тестов;
- поддержка гибкой и экстремальной методологий разработки;
- настраиваемая отчётность об ошибках;
- поиск ошибок, пропущенных при ручном тестировании;
- хорошая документация тестов.

Автоматизация тестирования не всегда имеет место быть. Возможны случаи, когда ручное тестирование предпочтительнее. Например, если пользовательский интерфейс приложения в будущем будет сильно изменён, то любая автоматизация должна быть переписана. Так же, иногда просто недостаточно времени, для автоматизации тестирования. Поэтому, разработчик сам выбирает какое тестирование использовать.

Немного о Selenium

Selenium – это набор программных средств с разными подходами к автоматизации тестирования. Большинство тестировщиков работают с одним или двумя инструментами, которые наиболее полно удовлетворяют потребностям проекта. Но изучения всех инструментов даст вам много возможностей по решению различных проблем с автоматизацией тестирования. Весь набор инструментов имеет богатый набор функций тестирования, специально предназначенных для тестирования веб-приложений всех типов. Эти операции очень гибки, что позволяет сравнивать ожидаемые результаты испытаний с реальным поведением приложения.

Одной из ключевых особенностей Selenium является поддержка выполнения тестов в различных браузерах.

Набор инструментов Selenium

Selenium состоит из различных программных инструментов. Каждый из них играет свою роль.

Selenium 2 (Selenium Webdriver)

Selenium 2 - будущее направление проекта и новейшее дополнение к инструментарию Selenium. Этот новый инструмент автоматизации предоставляет все виды удивительных особенностей, в том числе более связанный и объектно-ориентированный набор API, а также ответ на ограничения старой версии.

Selenium 2 является продуктом слияния Selenium и Webdriver. Он поддерживает Webdriver API и базовые технологии, а также Selenium 1 технологии под Webdriver API для максимальной гибкости в переносе тестов. Кроме того, Selenium 2 работает с интерфейсом Selenium RC для обратной совместимости.

Selenium 1 (Selenium RC or Remote Control)

Selenium RC был основным проектом в течение длительного времени, пока не произошло слияние с Webdriver, что привело к созданию нового и более мощного инструмента.

Selenium 1 по-прежнему активно используется и предоставляет некоторые особенности, которые недоступны в Selenium 2, включая поддержку нескольких языков программирования (Java, Javascript, Ruby, Python, Perl и C#) и совместимость почти со всеми браузерами.

Selenium IDE

Selenium IDE (Integrated Development Environment) является инструментом для создания тестовых сценариев. Представляет собой плагин для браузера Firefox и обеспечивает легкий в использовании интерфейс для разработки автоматизированных тестов. Selenium IDE имеет функцию записи, которая позволяет записывать действия пользователя, как они выполняются, а затем экспортирует их в многоразовый код одного из языков программирования, который в последующем может быть выполнен.

Selenium Grid

Selenium Grid позволяет масштабировать Selenium RC для больших наборов тестов и тестов, которые должны быть запущены в нескольких средах. Selenium Grid позволяет запускать тесты параллельно, то есть, различные тесты могут быть запущены одновременно на различных удалённых машинах. Это имеет два преимущества. Во-первых, если у вас есть большой набор тестов, или медленно выполняющиеся тесты, то вы можете существенно увеличить производительность с помощью Selenium Grid, разделив этот набор тестов на отдельные тесты, которые будут одновременно запускаться на разных удалённых машинах. Во-вторых, если вам надо запустить набор тестов в нескольких средах, то вы можете использовать отдельную удалённую машину для каждой среды. В обоих

случаях Selenium Grid значительно уменьшает время, необходимое для запуска наборов тестов за счёт использования параллельных вычислений.

Selenium IDE

Selenium IDE является инструментом для создания тестов. Это простой в использовании плагин для Firefox и, как правило, наиболее эффективный способ создания тестов. Он содержит контекстное меню, которое позволяет сначала выбрать элемент пользовательского интерфейса с текущей страницы браузера, а затем выбрать команду из списка с установленными параметрами в соответствии с контекстом выбранного элемента. Это не только экономит время, но и является отличным способом изучить синтаксис скриптов Selenium.

Установка Selenium IDE

Используя Firefox, сначала скачайте IDE плагин с официального сайта SeleniumHQ: <http://seleniumhq.org/download>.

Когда скачивание завершится, вы увидите сообщение, представленное на рисунке 1.

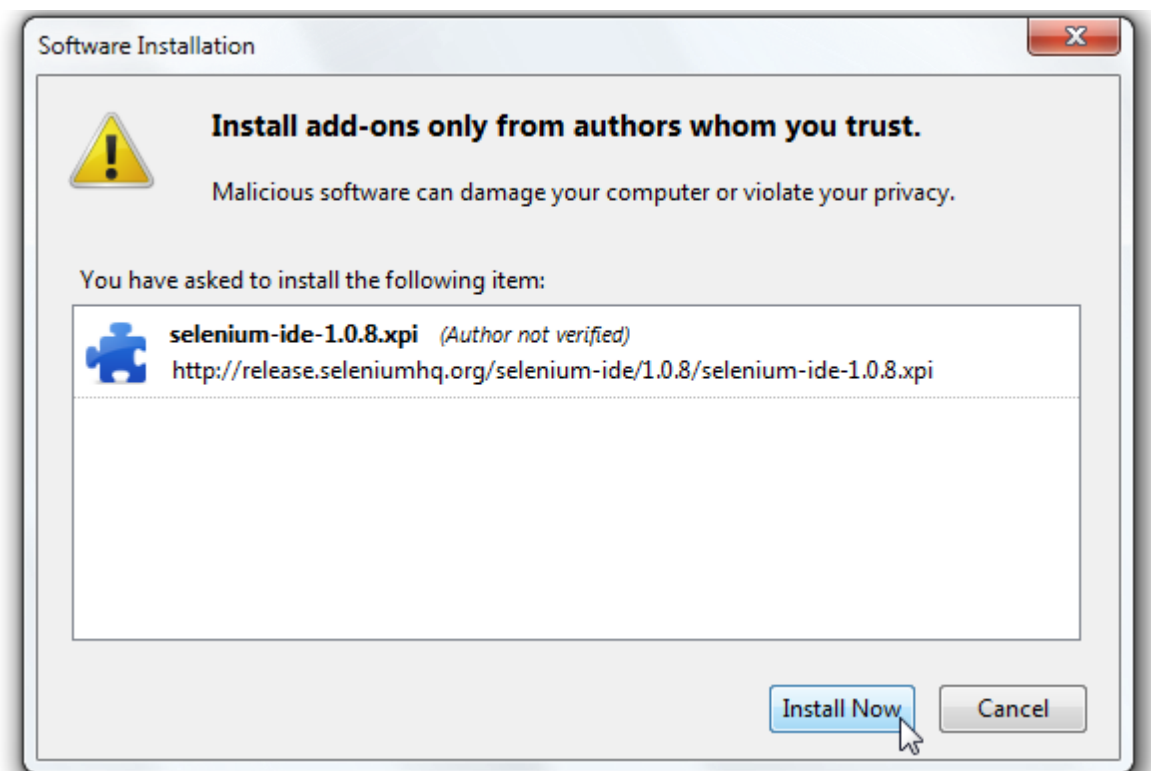


Рисунок 1 – Окно установки Selenium IDE

Выберите Install Now. После этого появится окно Add-ons (рисунок 2), показывающее процесс установки. А когда она закончится, то предложит перезапустить Firefox.

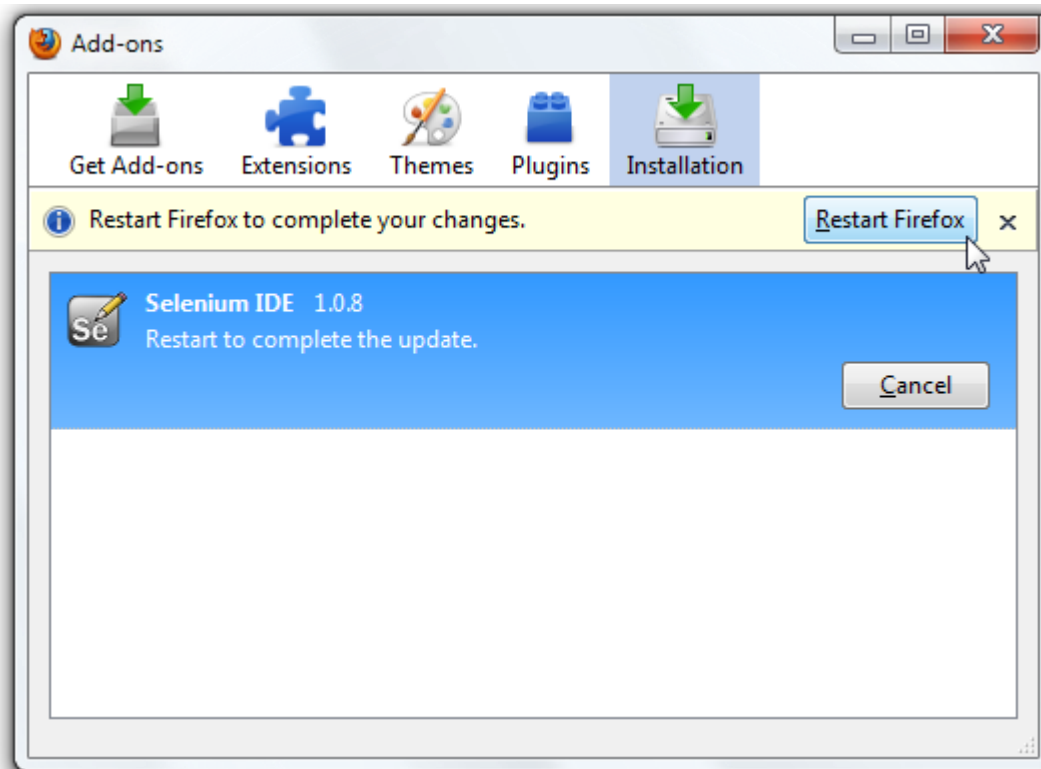


Рисунок 2 – Окно установки Selenium IDE

Перезапустите Firefox. После перезапуска вы найдёте Selenium IDE в меню Tools (Инструменты), расположение данного меню видно на рисунке 3.

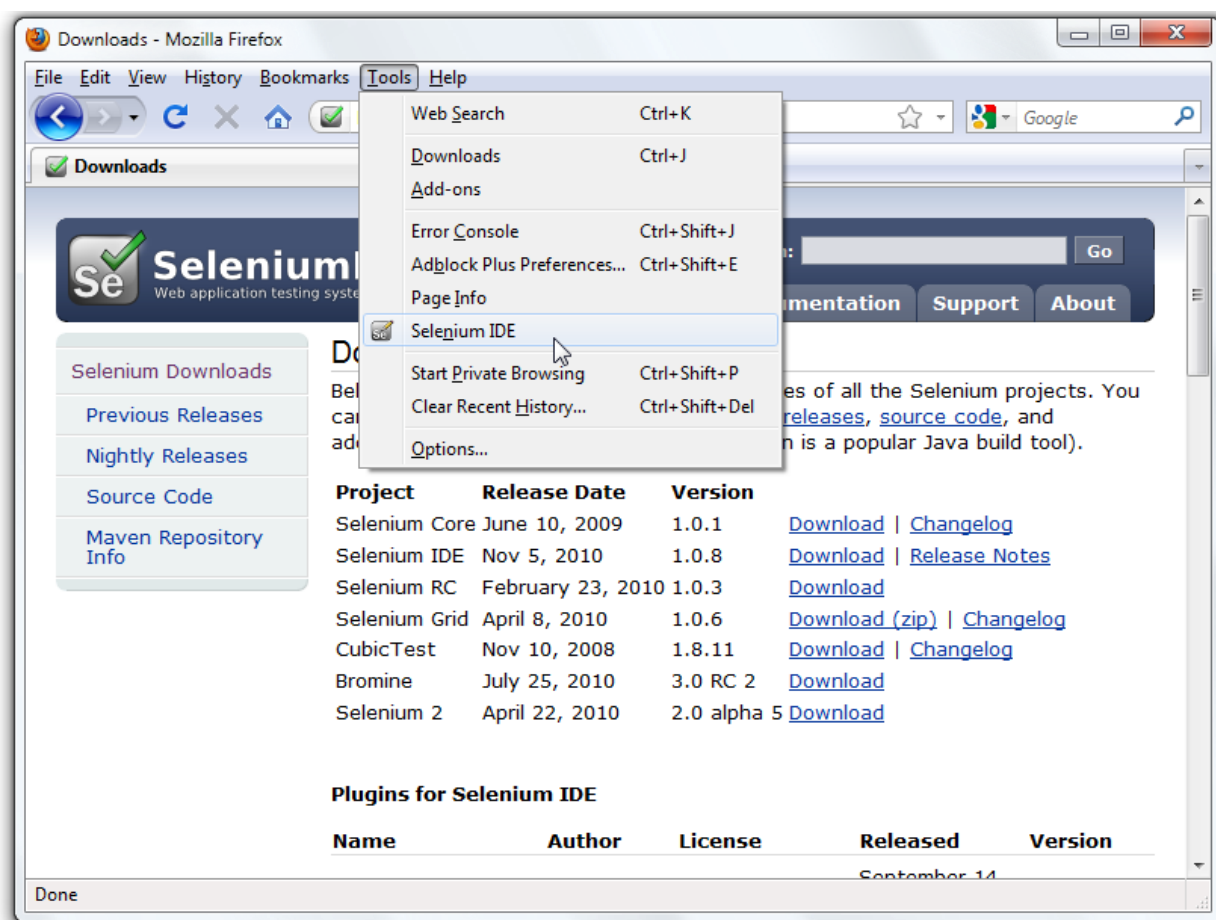


Рисунок 3 – Меню «Инструменты» в браузере Mozilla Firefox

Открытие IDE

Чтобы запустить Selenium IDE, просто выберите соответствующий пункт меню Tools. Откроется окно готовое к созданию нового теста.

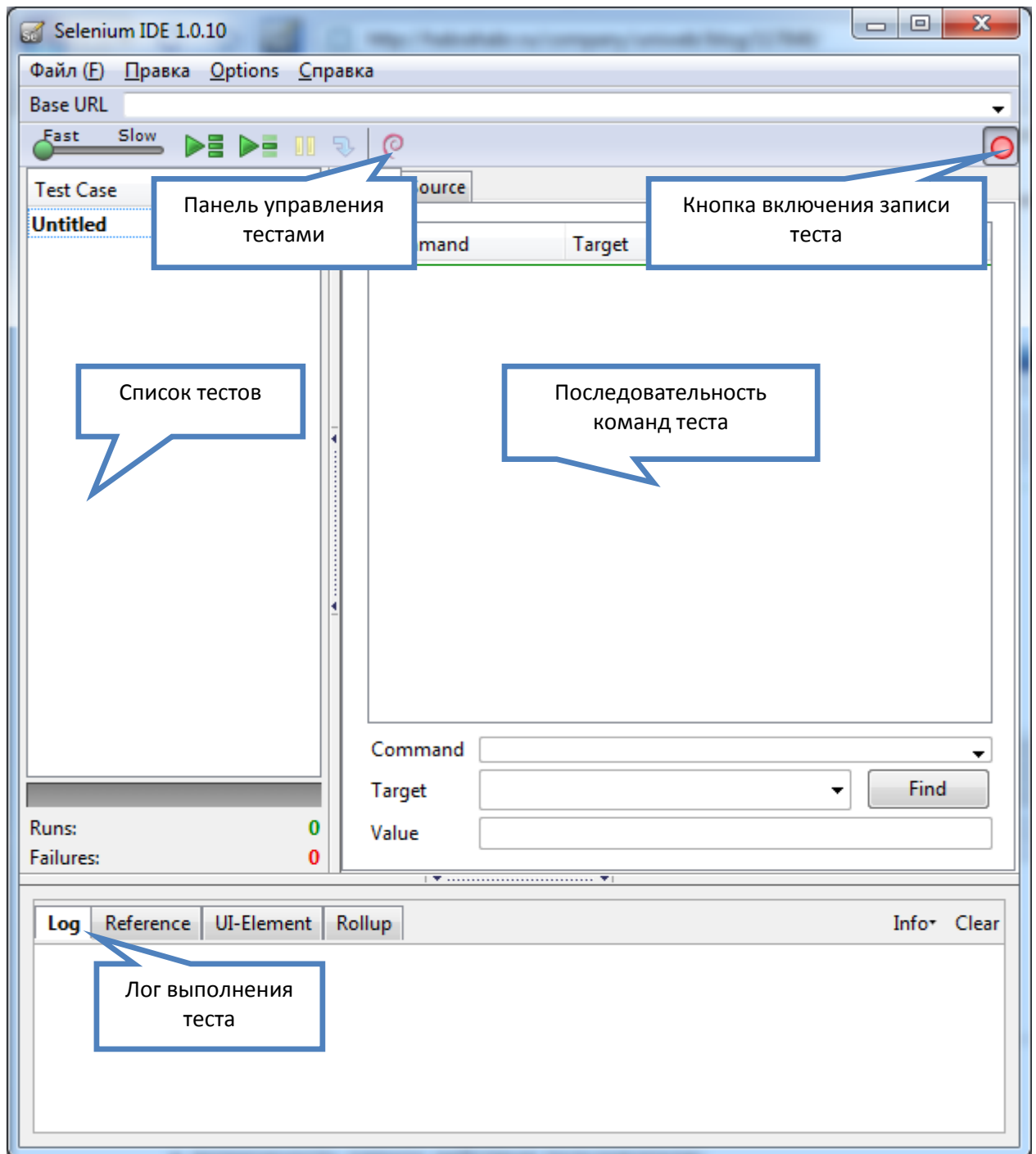


Рисунок 4 – Начальное окно Selenium IDE

Особенности IDE

Menu Bar

Меню File позволяет не только создавать, открывать и сохранять файлы тестов или наборов тестов, а также экспортировать исходный сценарий теста в файл, предварительно преобразовав его в код выбранного

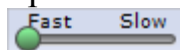
языка программирования. Пункт Edit позволяет копировать, вставлять, удалять, отменять, и выбрать все операции для редактирования команд в тестовых данных. Меню Options позволяет изменять настройки. Вы можете установить время ожидания для некоторых команд, указать формат (язык), используемый при сохранении тестов. Меню Help представляет собой стандартное меню справки.

Toolbar

Панель инструментов содержит кнопки для управления процессом выполнения ваших тестов, в том числе шаг функций для отладки тестов. Самая правая кнопка, в виде красного круга, это кнопка записи.



Speed Control: Управляет скоростью выполнения ваших тестов.



Run All: Выполняет весь набор тестов.



Run: Запускает выбранный тест. Если загружен только один тест, то кнопка Run All имеет тот же эффект.



Pause/Resume: Позволяет останавливать и продолжать выполнение запущенного теста.



Step: Позволяет выполнить тест по шагам. Используется для отладки теста.



TestRunner Mode: Позволяет вам запустить тест в браузере с помощью Selenium Core TestRunner. TestRunner в настоящее время обычно не используется. Эта кнопка предназначена для оценки тестов для обратной совместимости с TetsRunner.



Apply Rollup Rules: Эта дополнительная функция позволяет группировать повторяющиеся последовательности команд в единое действие.



Record: Записывает действия пользователя.



Test Case Pane

Ваш скрипт отображается в test case pane. Она имеет две вкладки, одна для отображения команд и их параметров в табличном представлении.

| Table Source | | |
|--------------|--------------|---------------|
| Command | Target | Value |
| open | / | |
| click | //input[@... | |
| type | q | selenium i... |
| clickAndWait | //ol[@id=... | |

Рисунок 5 – Test Case Pane (таблица)

Другая вкладка — отображает исходный текст теста, который будет сохранён в файл. По умолчанию, это HTML код, который может быть преобразован к другому языку (Java, C# и др.). Также на этой вкладке предоставляется возможность править код.

| Table | Source |
|--|--------|
| <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head profile="http://selenium-ide.openqa.org/selenium-ide.profile"> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> <link rel="selenium.base" href="http://www.selenium-ide.com/"> <title>New Test</title> </head> <body> <table cellpadding="1" cellspacing="1" border="1"> <thead> <tr> <td rowspan="1" colspan="3">New Test</td> </tr> </thead> <tbody> <tr> <td>open</td> <td>/</td> <td></td> </tr> <tr> <td>click</td> <td>//input[@id='q']</td> <td>selenium i...</td> </tr> <tr> <td>clickAndWait</td> <td>//ol[@id='1']</td> <td></td> </tr> </tbody> </table> </body> </html></pre> | |

Рисунок 6 – Test Case Pane (исходный код на языке HTML)

Поля Command, Target и Value отображают выбранную команду с её параметрами. Используя эти поля, вы можете изменить выделенную команду. Первый параметр команды, описанный внизу во вкладке Reference всегда идёт в поле Target. Если второй параметр задаётся в Reference, то он всегда идёт в поле Value.

| Table | Source |
|---|--------|
| <pre> using System; using System.Text; using System.Text.RegularExpressions; using System.Threading; using NUnit.Framework; using Selenium; namespace SeleniumTests { [TestFixture] public class BSUIR_test { private ISelenium selenium; private StringBuilder verificationErrors; </pre> | |

Рисунок 7 – Test Case Pane (исходный код на языке C#)

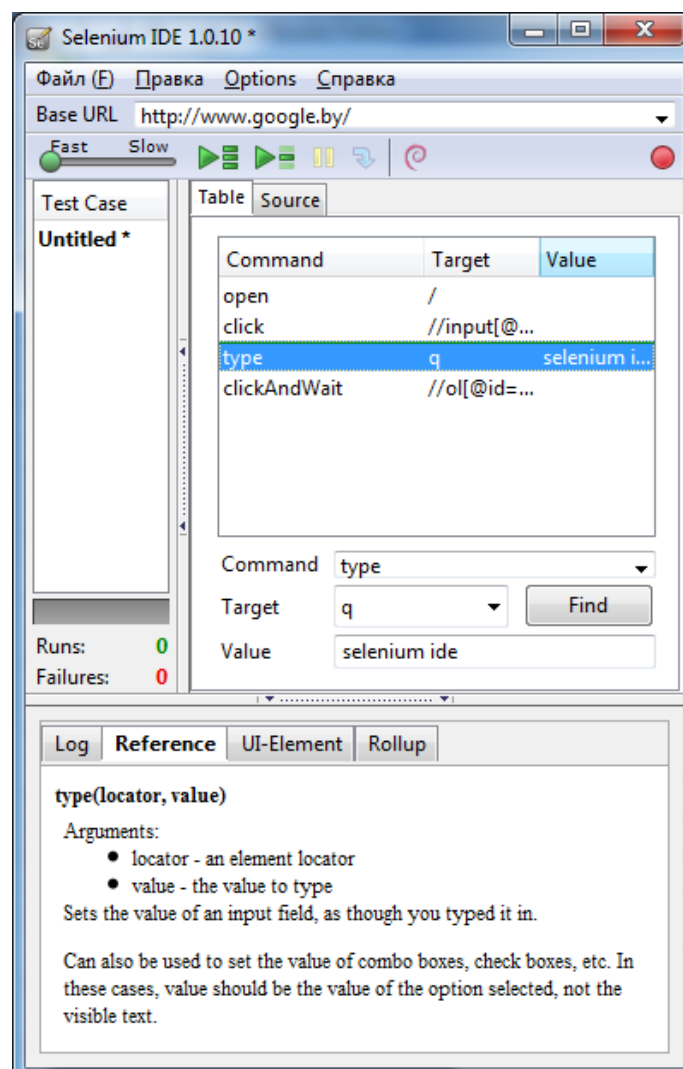


Рисунок 8 – Поля для создания команд

Log/Reference/UI-Element/Rollup

Нижняя панель используется для различных функций — Log, Reference, UI-Element и Rollup — в зависимости от выбранной вкладки.

Log

При запуске теста, сообщения об ошибках и информационные сообщения, которые отражают прогресс, отражаются на этой панели автоматически. Эти сообщения часто полезны для отладки тестов. Обратите внимание на кнопку Clear для очистки журнала. Также полезной будет кнопка Info, которая представляет собой выпадающий список с различными уровнями информации, отображающейся в Log.

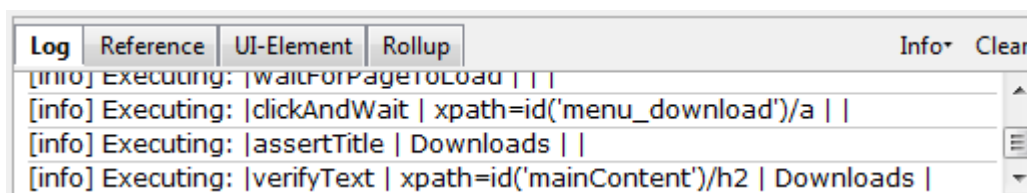


Рисунок 9 – Нижняя панель Selenium IDE (вкладка Log)

Reference

Эта панель выбрана по умолчанию, когда вы заходите или изменяете команды и их параметры в режиме Table. В этом режиме, панель Reference будет отображать документацию на текущую команду. При вводе или изменении команды, будь то режим Table или Source, крайне важно обеспечить, чтобы параметры, указанные в полях Target и Value, соответствовали параметрам, перечисленным в панели Reference. Число параметров должно совпадать с числом, указанным в Reference, также как и их последовательность и тип. Если есть несоответствия хоть в чём-то одном, команда не будет работать правильно.



Рисунок 10 – Нижняя панель Selenium IDE (вкладка Reference)

UI-Element u Rollup

Подробная информация об этих двух панелях, которые охватывают дополнительные функции, может быть найдена в UI-Element Documentation в меню Help.

Создание тестов

Существует три метода создания тестов. Как правило, разработчик тестов их все и использует.

Запись

Многие пользователи начинают создание тестов с записи их взаимодействий с веб-сайтом. При первом открытии Selenium IDE кнопка записи включена.

Во время записи, Selenium IDE автоматически вставляет команды в тест, основываясь на ваших действиях. Как правило, это включает в себя:

- нажатия по ссылкам – команды `click` и `clickAndWait`;
- ввод значений – тип команды;
- выбор опций из выпадающего списка – команда `select`;
- нажатия по флажкам и переключателям – команда `click`.

Несколько ценных замечаний:

- для определения типа команды может потребоваться нажать на некоторой другой области веб-страницы;

- переход по ссылке обычно записывает команду `click`. Однако, часто её придётся менять на `clickAndWait`, чтобы ваш тест останавливался пока новая страница полностью не загрузится. В противном случае, выполнение теста приведёт к непредсказуемым результатам.

Добавление Verifications и Asserts с помощью контекстного меню

В некоторых случаях, ваши тесты должны будут проверять свойства веб-страницы. Для этого необходимы команды `assert` и `verify`. Эти команды отличаются друг от друга только тем, что команда `assert` остановит выполнение теста при возникновении ошибки, а `verify` только запишет в log сообщение об ошибке и продолжит выполнение теста.

Во время записи, перейдите в браузер и нажмите правую кнопку мыши где-либо на странице. Вы увидите контекстное меню. Содержащее команды `verify` и/или `assert`.

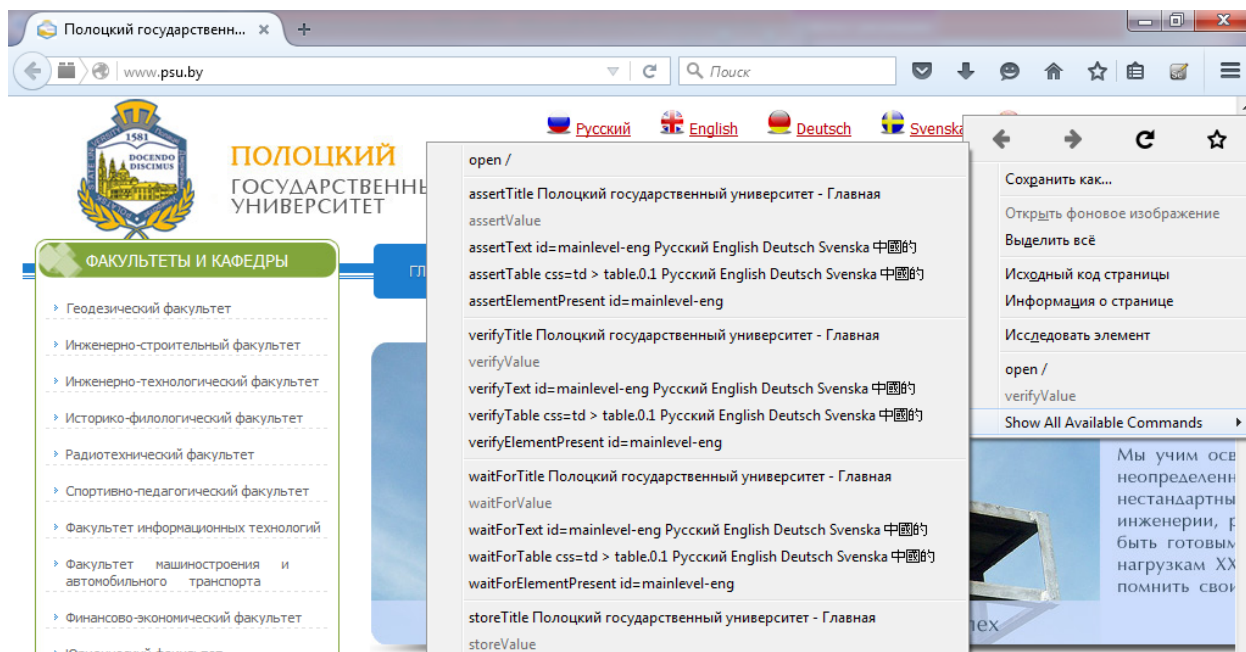


Рисунок 11 – Вызов контекстного меню

Откройте любую страницу и выделите на ней любой блок текста. Абзац или заголовок прекрасно подойдёт. Теперь щёлкните правой кнопкой мыши по выделенному тексту. В контекстном меню вы найдёте команду `verifyTextPresent`, параметром которой должен быть выделенный текст.

Обратите внимание на пункт `Show All Available Commands`. Он покажет много других доступных команд с параметрами, которые позволяют тестировать другие элементы пользовательского интерфейса.

Попробуйте проделать эту операцию ещё с несколькими элементами пользовательского интерфейса. Возможно, необходимые вам команды вы найдёте в пункте `Show All Available Commands`. Однако, часто используемые команды будут отображаться в контекстном меню.

Редактирование

Вставка команд

Режим Table

Выберите тест, в который вы хотите вставить команду. Затем выберите команду, перед которой надо вставить новую, и щёлкните правой кнопкой мыши.

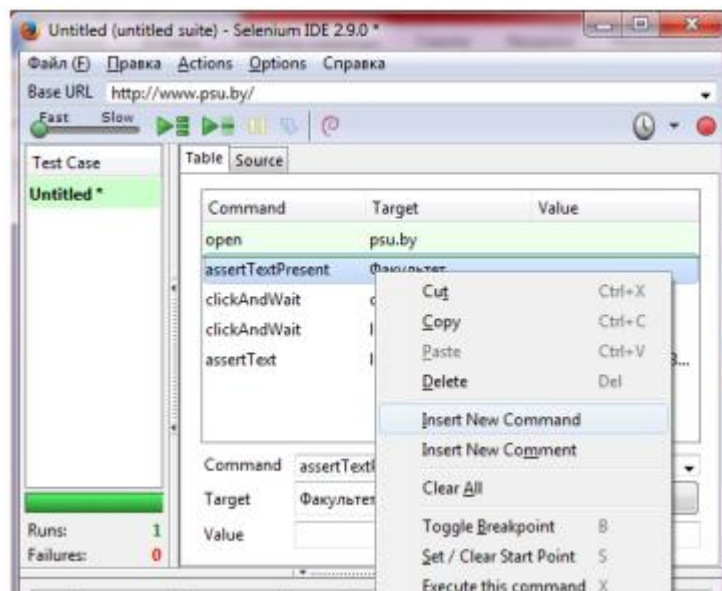


Рисунок 12 – Добавление команды

В появившемся контекстном меню выберите Insert New Command. Selenium IDE вставит пустую строку. Теперь используйте поля редактирования команды Command, Target и Value, чтобы задать команду и её параметры.

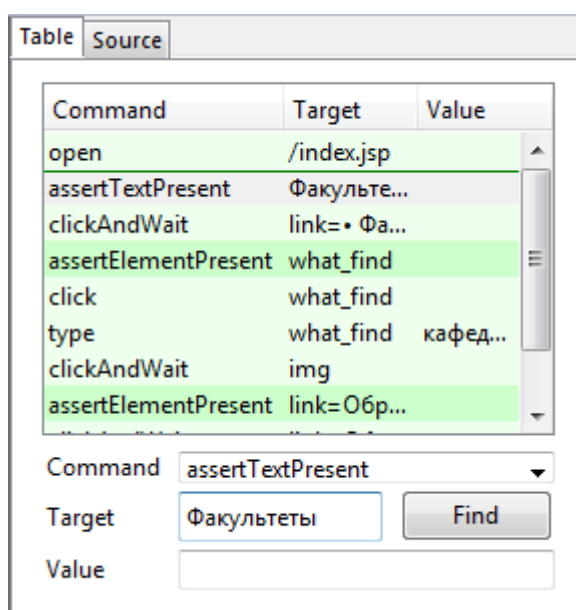


Рисунок 13 – Добавление команды

Режим Source

Выберите тест, в который хотите вставить команду. Откройте его представление в режиме Source. Найдите место, куда вы хотите вставить новую команду. Это место должно быть между тегами </tr> и <tr>. Вставьте сюда HTML-тег строки с тремя столбцами. Первый столбец содержит команду, а два других — первый и второй параметры команды, если они необходимы.

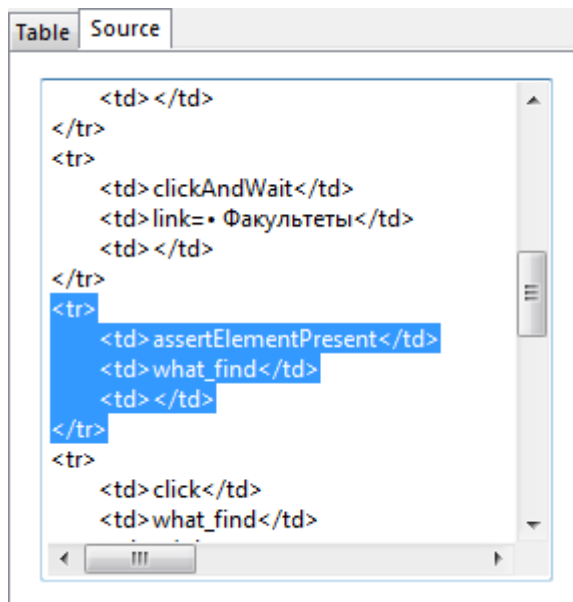


Рисунок 14 – Добавление команды

Сохраните изменения в тесте, чтобы они вступили в силу.

Вставка комментариев

Для того чтобы тест было легко читать и понимать, следует добавлять в него комментарии. При выполнении теста комментарии игнорируются.

Режим Table

Выберите строку в тесте, куда вы хотите вставить комментарий, нажмите правую кнопку мыши и выберите Insert New Comment.

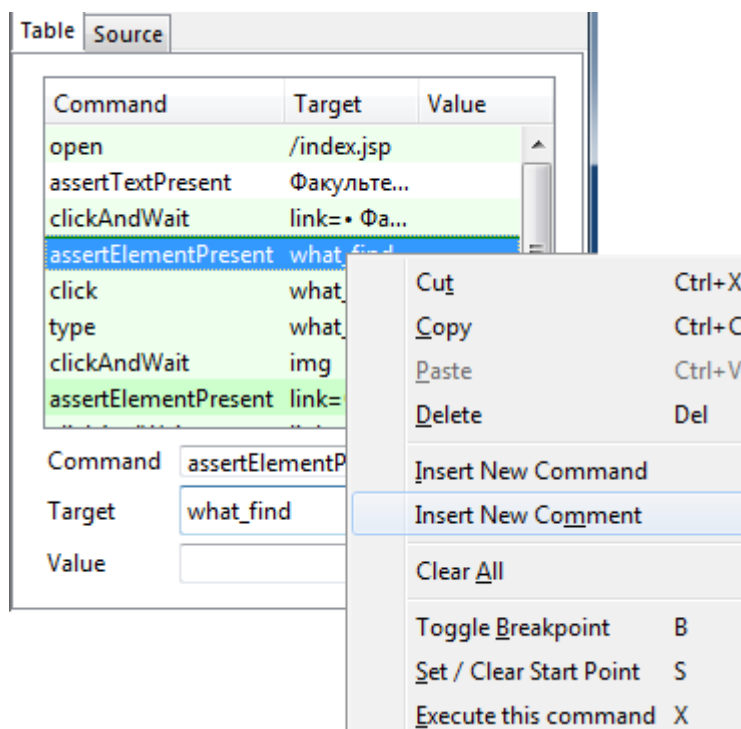


Рисунок 15 – Добавление комментария

Теперь используйте поле Command для ввода комментария. У текста комментария пурпурный цвет.

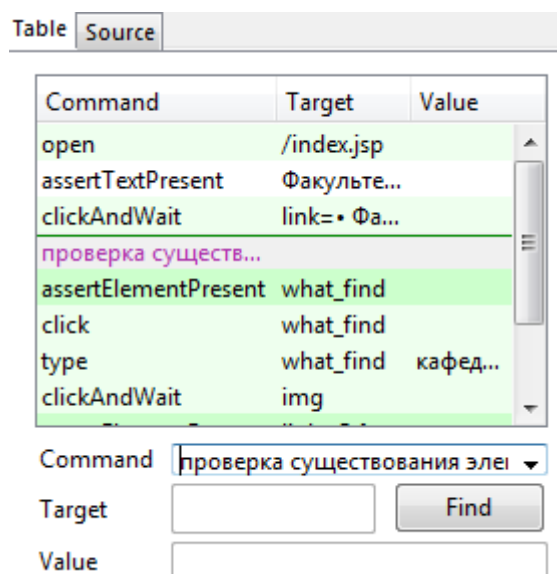


Рисунок 16 – Добавление комментария

Режим Source

Выберите место для комментария. Добавьте HTML-тег комментария, <!-- ваш комментарий -->.

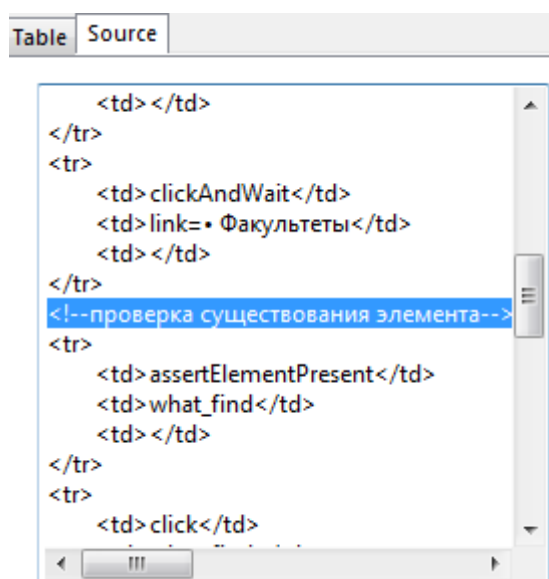


Рисунок 17 – Добавление комментария

Редактирование команды или комментария

Режим Table

Просто выберите нужную строку и редактируйте её, используя поля Command, Target и Value.

Режим Source

Этот режим выглядит как обыкновенный текстовый редактор. Просто найдите необходимый блок текста и правьте его.

Открытие и сохранение теста

Как и в большинстве программ, в этой есть команды Open и Save в меню File. Однако, Selenium различает один тест от набора тестов. Чтобы сохранить один тест воспользуйтесь командой Save Test Case. Чтобы сохранить весь набор тестов используйте Save Test Suite.

При открытии уже существующих тестов или набора тестов, Selenium IDE отображает список открытых тестов в Test Case панели.

Запуск тестов

Selenium IDE предоставляет много опций для выполнения тестов. Вы можете запустить на выполнение сразу весь тест, приостановить и продолжить выполнение, выполнять по одной строке за раз, запустить одну команду, над которой вы работаете, а также вы можете запустить весь набор тестов. Выполнение тестов в IDE очень гибкое.

Запуск теста

Нажмите на кнопке Run для запуска отображаемого теста.

Запуск набора тестов

Нажмите Run All для запуска всех тестов из загруженного набора.

Остановить и продолжить

Кнопка Pause служит для остановки теста во время выполнения. После нажатия, кнопка изменяется на команду Resume. Нажмите на Resume. Чтобы продолжить выполнение теста.

Точки прерывания

Вы можете установить на любой команде теста точку прерывания. Это очень полезно при отладке вашего теста. Для установки точки выберите команду, нажмите правую кнопку мыши, выберите из контекстного меню Toggle Breakpoint.

Изменить начальную точку

Вы можете указать IDE, что тест должен начаться с определённой команды. Это также очень помогает при отладке теста. Чтобы установить новую начальную точку, выберите команду, нажмите правую кнопку мыши и в контекстном меню найдите Set/Clear Start Point.

Запуск любой команды

Дважды нажмите на любой команде, чтобы выполнить её. Это полезно при написании отдельных команд. Это позволяет проверить команду, которую вы написали, если не уверены в её правильности.

Использование Base URL для запуска тестов на разных доменах

Поле Base URL очень полезно, так как позволяет запускать тесты на разных доменах. Предположим, что у сайта <http://news.portal.com> в домене есть бета сайт <http://beta.news.portal.com>. Любой тест для этих сайтов должен указать относительный URL в качестве аргумента, вместо абсолютного. Selenium IDE создаст абсолютный URL путём добавления аргумента

открытых команд в конец значения поля Base URL. Тест на рисунке 18 откроет страницу <http://news.portal.com/about.html>.

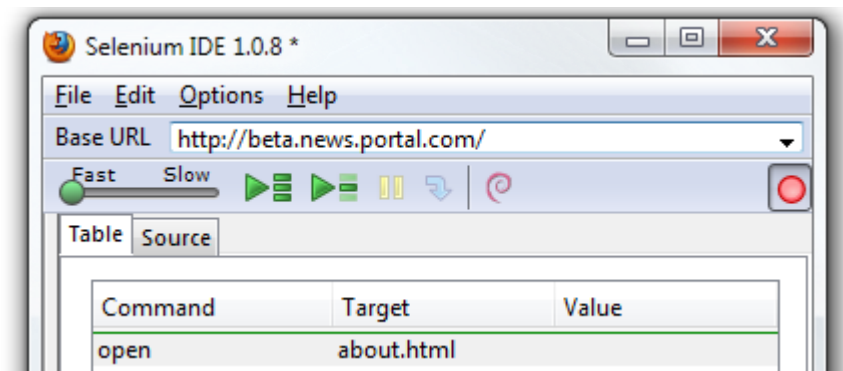


Рисунок 18 – Использование Base URL

Этот же тест, но с изменённым значением Base URL, откроет <http://news.portal.com/about.html>, пример на рисунке 19.

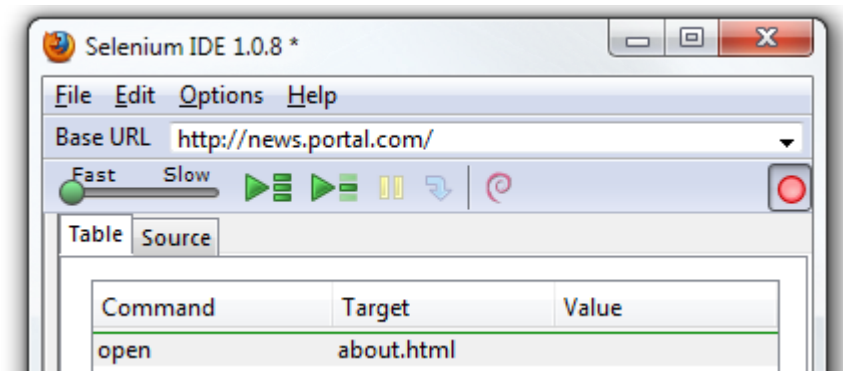


Рисунок 19 – Использование Base URL

Практическая часть

Содержание задания

Научиться составлять тесты с помощью Selenium IDE. Вариант задания взять у преподавателя. Автоматизировать тест из варианта.

Порядок выполнения работы

- 1 Ознакомиться с теоретической частью.
- 2 Установить Selenium IDE.
- 3 Взять вариант у преподавателя.
- 4 Автоматизировать тест из варианта задания с помощью Selenium IDE.
- 5 Составить отчет о проделанной работе.
- 6 Показать работу преподавателю.
- 7 После защиты, выполненного задания, скинуть отчёт преподавателю (в названии указать **фамилию и номер лабораторной**, например, Ivanov13).

Содержание отчета

- 1 Титульный лист.
 - 2 Ход выполнения задания, с пояснением каждого шага и скриншотами, а также с параметрами Command, Target и Value для каждого выполненного шага.
 - 3 Скриншот полученного теста в режиме таблицы (с комментариями).
 - 4 Исходный код теста в формате HTML (с комментариями).
 - 5 Вывод о проделанной работе.
- Защита работ проводится индивидуально.

Контрольные вопросы

- 1 Преимущества автоматизированного тестирования.
- 2 Особенности Selenium IDE.
- 3 Особенности Selenium Grid.
- 4 Какие команды, при записи теста, автоматически вставляет Selenium IDE?
- 5 Отличия команды Assert от Verify в Selenium IDE.

Вариант 1

ТЕСТИРОВАНИЕ ПОИСКА И НАВИГАЦИИ ПО САЙТУ:

- 1 Зайти на сайт www.psu.by.
- 2 Проверить на наличие, видимость и редактируемость поле ввода данных для поиска.
- 3 Ввести в поле для поиска заведомо несуществующую информацию.
- 4 Проверить наличие на новой странице запрашиваемого результата и текста «найдено 0 объектов».
- 5 Перейти на главную страницу.
- 6 Выполнить проверки из пункта 2.
- 7 Ввести в поле ввода заведомо существующую информацию.
- 8 Проверить наличие на новой странице текста с вашей информацией и ссылки на эту информацию.
- 9 Перейти по найденной ссылке.
- 10 Проверить наличие на странице искомого текста.
- 11 Проверить наличие и работоспособность кнопки для перемещения в начало страницы.
- 12 Определить наличие ссылки «На главную страницу»
- 13 Перейти по ссылке «Главная».
- 14 В списке факультетов проверьте наличие ссылки на «Факультет информационных технологий» и перейдите по этой ссылке.
- 15 Проверить правильность заголовка окна браузера.
- 16 На странице факультета проверьте наличие и доступность вкладки «Кафедры». Откройте данную вкладку.
- 17 Перейти по ссылке «Кафедра технологий программирования».
- 18 Проверить наличие текста «Голубева Оксана Валерьевна», наличие и доступность ссылки «Вперед».
- 19 Перейти по ссылке «Вперед».
- 20 Убедиться, что переход произошел.
- 21 Выполните переход на главную страницу.
- 22 Закрывать окно браузера.

Вариант 2

ТЕСТИРОВАНИЕ РАЗДЕЛА «УНИВЕРСИТЕТ»:

- 1 Зайти на сайт www.psu.by.
- 2 Проверить наличие ссылки «Персоналии (А-Я)» в разделе «Университет».
- 3 Перейти по этой ссылке.
- 4 Проверить правильность заголовка окна браузера.
- 5 Проверить страницу на наличие текста «Персоналии».
- 6 Проверить наличие и доступность первой буквы вашей фамилии в предложенном списке.
- 7 Перейти по данной ссылке.
- 8 Проверить на наличие, видимость и работоспособность одной из ссылок на преподавателя.
- 9 Проверить соответствие названия ссылки содержанию полученной страницы.
- 10 Вернуться назад.
- 11 Убедиться, что возврат произошёл на предыдущую страницу.
- 12 Проверить наличие ссылки «Общие отделы и службы» в разделе «Университет».
- 13 Перейти по этой ссылке.
- 14 Проверить правильность перехода по данной ссылке.
- 15 Проверить наличие и доступность ссылки «Здравпункт».
- 16 Проверить наличие и доступность ссылки «Бухгалтерия».
- 17 Перейти по данной ссылке.
- 18 Подождать загрузки страницы 10 секунд.
- 19 Проверить наличие электронного адреса бухгалтерии.
- 20 Выполните переход на главную страницу.
- 21 Закрыть окно браузера.

Вариант 3

ТЕСТИРОВАНИЕ РАСПИСАНИЯ ЗАНЯТИЙ:

- 1 Зайти на сайт www.psu.by
- 2 Проверить наличие ссылки «Расписание занятий» в разделе «Сотрудникам».
- 3 Перейти по этой ссылке.
- 4 Проверить правильность заголовка окна браузера.
- 5 Проверить наличие таблицы «Расписание звонков» со столбцами пар и учебных корпусов.
- 6 Проверить наличие списка всех факультетов.
- 7 Проверить работоспособность 5 ссылок для скачивания расписания (для различных факультетов и курсов).
- 8 Проверить работоспособность ссылки «Расписание кураторского часа».
- 9 Проверить наличие и доступность ссылки «Назад».
- 10 Перейти по ссылке «Назад».
- 11 По ссылке «Вперед» вернуться на страницу с расписанием.
- 12 Убедиться, что возврат произошел на нужную страницу.
- 13 Выполнить переход на главную страницу.
- 14 Проверить на наличие и видимость поле ввода данных для поиска.
- 15 Ввести в поле ввода заведомо существующую информацию.
- 16 Проверить наличие на новой странице текста с вашей информацией и ссылки на эту информацию.
- 17 Перейти по найденной ссылке.
- 18 Проверить наличие на странице искомого текста и наличие ссылки «Главная».
- 19 Перейти по ссылке «Главная».
- 20 Закрыть окно браузера.

Вариант 4

ТЕСТИРОВАНИЕ САЙТА НА АНГЛИЙСКОМ ЯЗЫКЕ:

- 1 Зайти на сайт www.psu.by.
- 2 Проверить наличие ссылки выбора языка сайта «EN».
- 3 Перейти по данной ссылке.
- 4 Проверить наличие ссылки выбора языка «RU».
- 5 Проверить наличие меню с рубриками «University», «Science», «Education», «Faculties».
- 6 Проверить доступность и работоспособность каждой рубрики.
- 7 Проверить наличие подразбук у рубрики «Faculties».
- 8 Получить название первой подразбук.
- 9 Перейти по этой ссылке.
- 10 Проверить наличие на загруженной странице текста, совпадающего с названием подразбук.
- 11 Вернуться на главную страницу.
- 12 Проверить наличие ссылки на факультет информационных технологий в рубрике «Faculties».
- 13 Перейти к странице факультета.
- 14 Проверить наличие информации о декане.
- 15 На странице факультета проверить наличие и доступность ссылки «Кафедра технологий программирования».
- 16 На странице факультета проверить наличие и доступность ссылки «Кафедра вычислительных систем и сетей».
- 17 Проверить наличие, доступность и работоспособность кнопки возврата в начало страницы.
- 18 Выполнить переход на главную страницу.
- 19 Сменить язык на «Русский».
- 20 Закрывать окно браузера.

Вариант 5

ТЕСТИРОВАНИЕ НАВИГАЦИИ И ПОИСКА НА САЙТЕ:

- 1 Зайти на сайт <http://lib.psu.by>.
- 2 Проверить наличие ссылки для поиска.
- 3 Нажать на данную ссылку.
- 4 Проверить на наличие, видимость и редактируемость поле ввода данных для поиска.
- 5 Ввести в поле для поиска заведомо несуществующую информацию.
- 6 Проверить наличие на новой странице запрашиваемого результата и текста «Найдено 0 объектов».
- 7 Выполнить проверки из пункта 4.
- 8 Ввести в поле ввода заведомо существующую информацию.
- 9 Проверить наличие на новой странице текста с вашей информацией и ссылки на эту информацию.
- 10 Перейти по найденной ссылке.
- 11 Проверить наличие на странице искомого текста и наличие ссылки «Главная».
- 12 Перейти по ссылке «Главная».
- 13 Проверить наличие и доступность ссылки «Ресурсы».
- 14 Перейти по данной ссылке.
- 15 Подождать загрузки страницы 15 секунд.
- 16 Проверить правильность заголовка окна браузера.
- 17 Проверить наличие ссылки «Новые поступления» в разделе «Читателям» и перейти по данной ссылке.
- 18 Проверить наличие, доступность и работоспособность кнопки для возврата к началу страницы.
- 19 Проверить произошел ли возврат к началу страницы.
- 20 Выполнить переход на главную страницу.
- 21 Закрыть окно браузера.

Основные команды для тестов в Selenium IDE

Open – открыть страницу в браузере по определенному адресу.

Click – произвести нажатие по элементу страницы.

Type – ввести значение в текстовое поле страницы.

Select – выбрать значение из выпадающего списка.

GetTitle – возвращает заголовок текущей страницы.

GetValue – возвращает значение элемента страницы.

GoBack – вернуться на предыдущую страницу.

Close – закрыть текущее окно.

VerifyLocation / AssertLocation – проверить адрес текущей страницы.

VerifyTitle / AssertTitle – проверить значение заголовка страницы.

VerifyValue / AssertValue – проверить значение элемента страницы.

VerifyTextPresent / AssertTextPresent – проверить, что страница содержит указанный в команде текст.

VerifyElementPresent / AssertElementPresent – проверить, есть ли на странице указанный элемент.

WaitForCondition – ожидание выполнения определенного события на странице, указанного в параметре script (например, загрузка определенного элемента страницы, или страницы в целом).

WaitForFrameToLoad – ждет загрузки фрейма на странице указанное количество времени.

WaitForPageToLoad – ждет загрузки страницы указанное количество времени.

WaitForPopUp – ждет появления PopUp элемента страницы указанное количество времени.