

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Кафедра информационных
систем управления

ИТОГОВЫЙ ОТЧЕТ О ПРОХОЖДЕНИИ КУРСА
«ТЕСТИРОВАНИЕ И ОЦЕНКА КАЧЕСТВА ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ»

Выполнил:

студент 2 курса 14 группы
факультета прикладной
математики и информатики
Петров Андрей Александрович

Руководитель:

Старший преподаватель Кваша
Д.Ю.

Минск
2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ЛАБОРАТОРНАЯ РАБОТА №1. СОСТАВЛЕНИЕ СХЕМЫ ТЕСТИРОВАНИЯ ПРЕДМЕТА	4
ЛАБОРАТОРНАЯ РАБОТА №2.1. ТЕСТИРОВАНИЕ ПРЕДМЕТА. СОЗДАНИЕ УЧЕБНОГО ТЕСТА С ОТВЕТАМИ ПРИ ПОМОЩИ HTML И JAVASCRIPT ..	5
ЛАБОРАТОРНАЯ РАБОТА №2.2. ТЕСТИРОВАНИЕ ПРЕДМЕТА. ТЕСТИРОВАНИЕ ТРЕУГОЛЬНИКА	6
ЛАБОРАТОРНАЯ РАБОТА №4. СОЗДАНИЕ ПРОЕКТОВ В TESTCOMPLETE	9
ЛАБОРАТОРНАЯ РАБОТА №5.1. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ. ТЕСТОВАЯ ДОКУМЕНТАЦИЯ .	10
ЛАБОРАТОРНАЯ РАБОТА №5.2. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ В TESTLINK.....	13

ВВЕДЕНИЕ

Тестирование программного обеспечения — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование — это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

Качество программного обеспечения (Software Quality) — это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

Цели тестирования:

- повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах;
- повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям;
- предоставление актуальной информации о состоянии продукта на данный момент.

Этапы тестирования:

1. Анализ продукта
2. Работа с требованиями
3. Разработка стратегии тестирования
и планирование процедур контроля качества
4. Создание тестовой документации
5. Тестирование прототипа
6. Основное тестирование
7. Стабилизация
8. Эксплуатация

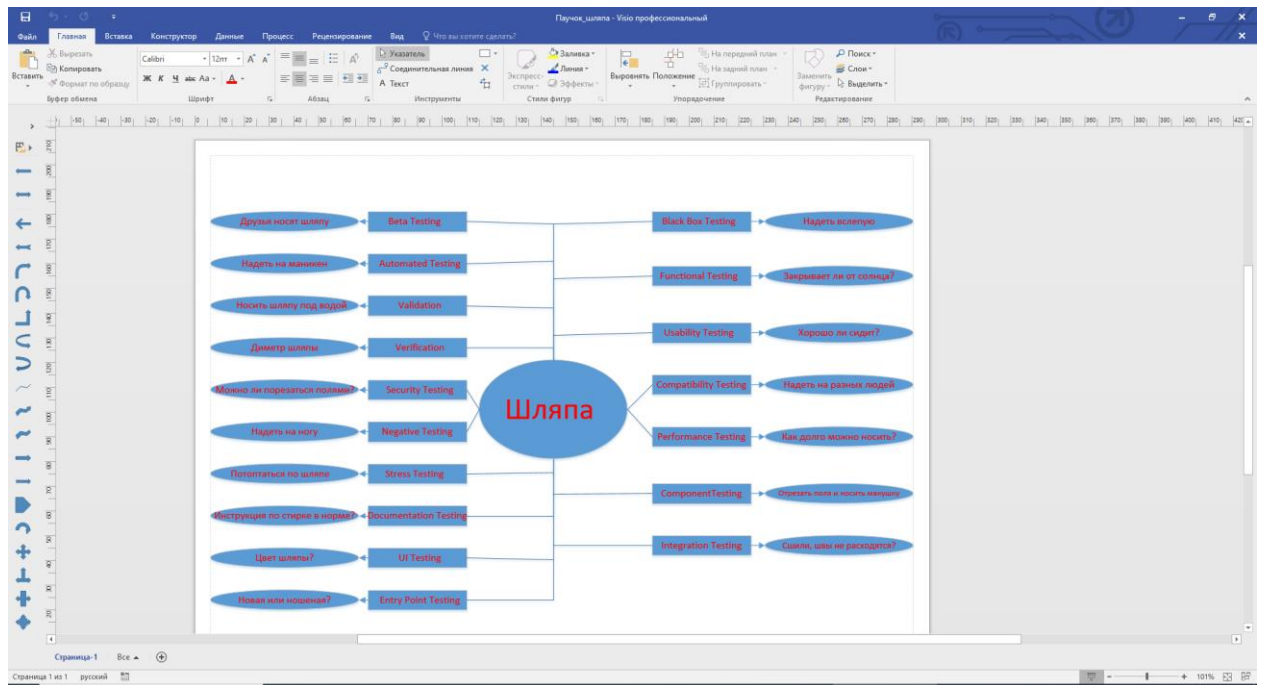
ЛАБОРАТОРНАЯ РАБОТА №1. СОСТАВЛЕНИЕ СХЕМЫ ТЕСТИРОВАНИЯ ПРЕДМЕТА

Цель работы: создать схему тестирования предмета встроенными средствами программного обеспечения MS Visio.

Ход работы:

При выполнении данной лабораторной работы нами были изучены встроенные средства для разработки и создания схем в MS Visio.

Предметом тестирования была выбрана «шляпа». Исходя из фиксированных категорий стадий тестирования, нами были подготовлены и отражены в схеме вопросы, ответы на которые помогут произвести тестирования нашего предмета.



ЛАБОРАТОРНАЯ РАБОТА №2.1. ТЕСТИРОВАНИЕ ПРЕДМЕТА. СОЗДАНИЕ УЧЕБНОГО ТЕСТА С ОТВЕТАМИ ПРИ ПОМОЩИ HTML И JAVASCRIPT

Цель работы: в соответствии с выбранным индивидуальным заданием составить тест для своей информационной системы (тестирование предмета) создать свой тест по шаблону теста на 15 вопросов с изображениями, перемешиванием вопросов и соответствующим оформлением и словесными ответами.

Ход работы:

При выполнении данной лабораторной работы мной был исследован интернет-ресурс, располагаемый по ссылке: <https://webriz.ru/36-sozдание-uchebnogo-testa-s-otvetami.html>.

Для создания теста был выбран предмет «Принтер». Тест содержит 15 вопросов с изображениями и перемешиванием вопросов, соответствующим оформлением и словесными ответами.

The image shows a screenshot of a web-based test interface. The background is light green. At the top, it says 'Тест. Тема - "Принтер".' Below that, it indicates 'Задания выполнены верно на 93%.' in blue text. A list of 15 questions follows, each with a status and the user's answer. Questions 1-14 are marked as correct, while question 15 is marked as incorrect. To the right, there is a separate box showing 'Тест. Тема - "Принтер".' and 'Вопрос 1' with the question 'Напечатал лист?'. Below the question is an empty text input field and a button labeled 'Следующий вопрос'.

Тест. Тема - "Принтер".

Задания выполнены верно на 93%.

Вопрос 1. Вы ответили верно. Ваш ответ: да
Вопрос 2. Вы ответили верно. Ваш ответ: да
Вопрос 3. Вы ответили верно. Ваш ответ: да
Вопрос 4. Вы ответили верно. Ваш ответ: нет
Вопрос 5. Вы ответили верно. Ваш ответ: нет
Вопрос 6. Вы ответили верно. Ваш ответ: да
Вопрос 7. Вы ответили верно. Ваш ответ: нет
Вопрос 8. Вы ответили верно. Ваш ответ: да
Вопрос 9. Вы ответили верно. Ваш ответ: да
Вопрос 10. Вы ответили верно. Ваш ответ: да
Вопрос 11. Вы ответили верно. Ваш ответ: да
Вопрос 12. Вы ответили верно. Ваш ответ: да
Вопрос 13. Вы ответили верно. Ваш ответ: да
Вопрос 14. Вы ответили верно. Ваш ответ: нет
Вопрос 15. Вы ответили не верно. Ваш ответ: нет

Тест. Тема - "Принтер".

Вопрос 1

Напечатал лист?

Следующий вопрос

ЛАБОРАТОРНАЯ РАБОТА №2.2. ТЕСТИРОВАНИЕ ПРЕДМЕТА. ТЕСТИРОВАНИЕ ТРЕУГОЛЬНИКА

Цель работы: написать программу на Java (создание классов); создать тесты, покрывающие критерии по стратегии белого ящика (покрытие всех путей); создать тесты, покрывающие критерии по стратегии черного ящика (анализ граничных условий); в тестах указать входные данные, ожидаемые результаты; создать два драйвера для запуска всех тестов (отдельно для критериев каждой стратегии); драйвер должен выводить результаты прохождения всех тестов: исходные данные, полученные результаты, ожидаемые результаты и результаты прохождения теста – пройден тест или нет.

Ход работы:

В среде разработки IntelliJ IDEA языка Java была написана программа, которая считывает из стандартного входного потока три целочисленных значения, вводимых пользователем. Три значения интерпретируются как длины сторон треугольника. Программа вычисляет площадь, периметр, выводит вычисленные значения и выводит сообщение, которое заявляет, является ли треугольник разносторонним, равнобедренным или равносторонним.

Вывод данных был реализован в стандартный выходной поток. Тесты для каждой стратегии были пронумерованы уникальными номерами. Были реализованы:

- тест для действительного разностороннего треугольника,
- тест для действительного равностороннего треугольника,
- три теста для действительных равнобедренных треугольников;
- тест «одна, две или три стороны есть нулевое число (5 тестов)»;
- тест «у одной из сторон отрицательное значение (или у двух, или у трех)»;
- тест «сумма двух чисел равняется третьему?»;
- тест «сумма двух чисел меньше, чем третье?»;
- тест «Число нецелое?»;
- тест «Входные данные выходят за пределы допустимых значений или нет?»;
- тест «P и S выходят за пределы допустимых значений или нет?»

Ожидаемый результат: равносторонний треугольник

Test 1:

Input first number: 3 4 5

Input second number: Input third number: P = 12.0

S = 6.0

разносторонний

Тест не пройден

Test 2:

Input first number: 4 4 4

Input second number: Input third number: P = 12.0

S = 6.928203230275509

равносторонний

Тест пройден

Test 3:

Input first number: 6 5 5

Input second number: Input third number: P = 16.0

S = 12.0

равнобедренный

Тест не пройден

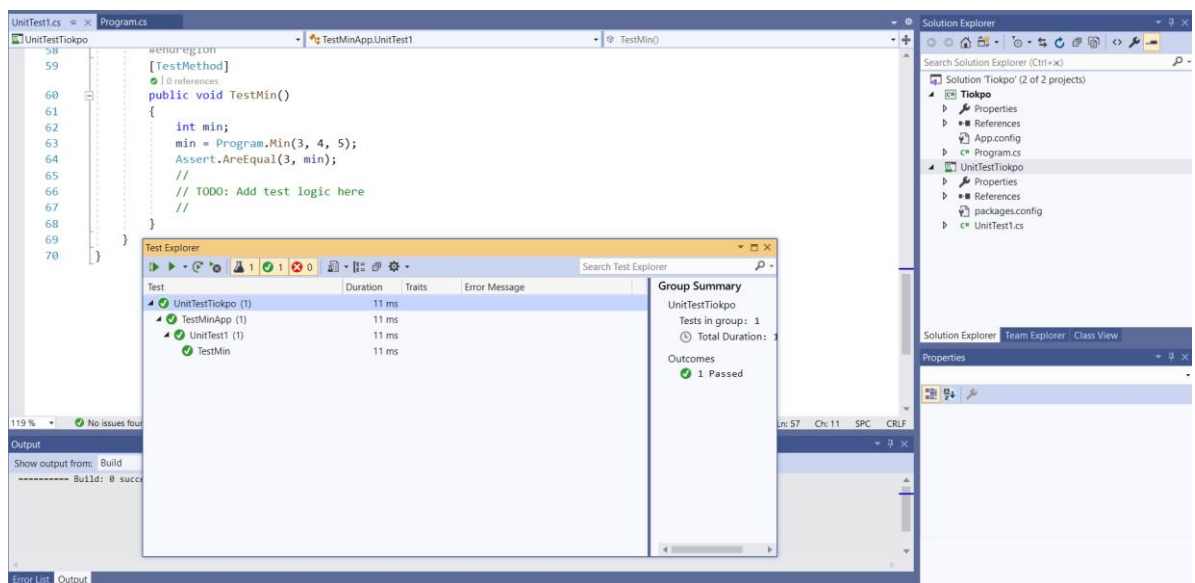
ЛАБОРАТОРНАЯ РАБОТА №3. СОЗДАНИЕ UNIT-ТЕСТА В MS VISUAL STUDIO 2010 – C#

Цель работы: привести пошаговый процесс создания простейшего Unit-теста в системе Microsoft Visual Studio 2010 (C#) для приложения типа Console Application, научиться создавать собственные Unit-тесты, продемонстрировать использование класса Assert для проведения тестирования работы функций.

Ход работы:

В среде разработки Microsoft Visual Studio было создано приложение по шаблону Console Application. Для приложения был разработан Unit-тест, который тестирует работу функции Min(), которая определяет максимальный элемент из трех чисел. Для функции Min() также был установлен метод тестирования TestMin(), после чего была осуществлена проверка работы функции.

В Microsoft Visual Studio 2010 для работы с Unit-тестами реализовано специальное меню команд, которое называется Test. С помощью него был запущен тест на выполнение и проверен результат тестирования.

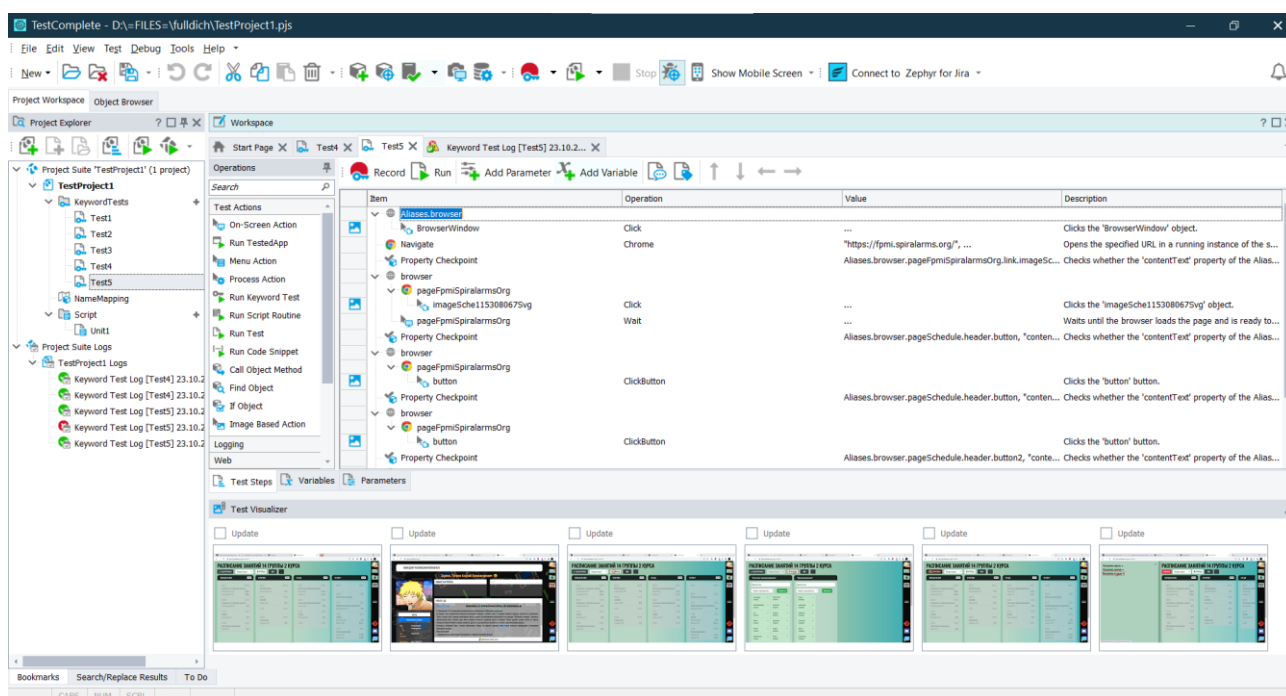


ЛАБОРАТОРНАЯ РАБОТА №4. СОЗДАНИЕ ПРОЕКТОВ В TESTCOMPLETE

Цель работы: изучить порядок установки и произвести тестирование приложений в TestComplete приложений.

Ход работы:

В ходе выполнения данной лабораторной работы на ПЭВМ было установлено программное обеспечение «TestComplete» версии 14. Предметом тестирования был выбран интернет-ресурс: <https://fpmi.spiralarms.org/> «Менеджер расписания». После исследования возможностей TestComplete был создан проект тестирования web-приложения и осуществлен его запуск и произведено тестирование интернет-ресурса.



ЛАБОРАТОРНАЯ РАБОТА №5.1. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ. ТЕСТОВАЯ ДОКУМЕНТАЦИЯ

Цель работы: разработать рабочую тестовую документацию для тестирования веб-приложения.

Ход работы:

Создание тестовой документации значительно улучшает качество продукта за счет более тесного сотрудничества, уточнения деталей при разработке плана тестирования и документации. После завершения тестирования наличие тестовой документации позволяет проверить, насколько успешно были проведены все этапы тестирования.

В качестве веб-приложения для тестирования был выбран интернет-ресурс «Менеджер расписания ФПМИ БГУ», расположенный по ссылке: <https://fpmi.spiralarms.org/>. Тестирование веб-приложений обладает рядом особенностей, и при проведении самого процесса тестирования необходимо обращать внимание на следующие аспекты:

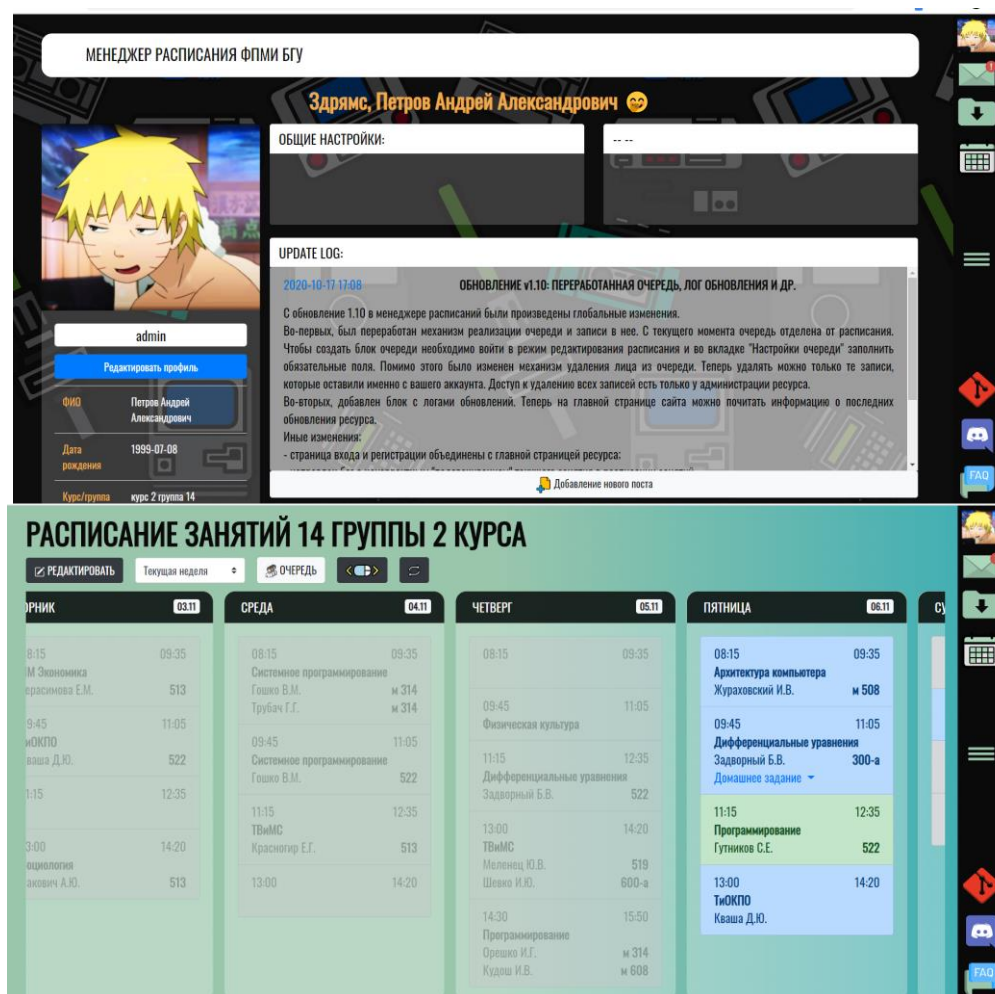
1. единство дизайна;
2. навигация;
3. функциональность;
4. совместимость с браузером;
5. совместимость с операционной системой;
6. "дружественность";
7. "работоспособность".

Исходя из этого, для выбранного мною веб-приложения была составлена следующая тестовая документация.

ТЕСТОВАЯ ДОКУМЕНТАЦИЯ ДЛЯ ТЕСТИРОВАНИЯ ВЕБ- ПРИЛОЖЕНИЯ «МЕНЕДЖЕР РАСПИСАНИЯ ФПМИ БГУ»

Раздел 1. Единство дизайна.

Страницы приложения выполнены в единой цветовой гамме, придающей всем страницам сайта «единство». Страницы выглядят современно, все элементы стилизованы в едином дизайне. Шрифты, используемые для различных уровней заголовков, отражают зависимость между ними, а иконки отражают суть ссылок и кнопок.



Раздел 2. Навигация.

Навигация по сайту осуществляется при помощи кнопок и ссылок, характеризующих необходимые страницы/элементы, расположенных в правом боковом меню. Присутствует перемещение по элементам страницы с помощью клавиши TAB, что сопровождается визуальным выделением компонента.

Раздел 3. Функциональность.

Приложение имеет возможность просматривать расписание занятий как текущей, так и следующей недели. Помимо этого, имеется возможность редактирования расписания:

- изменение даты недели;
- изменение редактирование предмета, кабинета, преподавателя, времени занятий и домашнего задания;

Присутствует функционал для работы с очередью:

- создание очереди;
- запись с очереди;
- удаление из очереди;

Также возможно редактировать свои личные данные.

Раздел 4. Совместимость с браузером.

Приложение совместимо со всеми современными браузерами (Google Chrome, IE8+, Firefox, Safari, Opera, MS Edge)

Раздел 5. Дружественность.

Приложение просто и легко в обращении. Интуитивно понятен интерфейс сайта: легка навигация, доступно меню, меню имеет возможность полноэкранного раскрытия, что упрощает выбор пункта меню на мобильных устройствах. Отсутствуют раздражающие пользователя приемы, мало всплывающих окон. Все ссылки являются «рабочими», необходимые данные присутствуют.

Раздел 6. Работоспособность.

Приложение загружается очень быстро. Используется технология PWA и модульности компонентов, что позволяет пользователю догружать необходимый контент, а не загружать страницу полностью

ЛАБОРАТОРНАЯ РАБОТА №5.2. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ В TESTLINK

Цель работы: изучить порядок установки и произвести тестирование приложений в TestLink.

Ход работы:

В ходе выполнения данной лабораторной работы были изучены возможности и функционал программного обеспечения TestLink. Предметом тестирования был выбран интернет-ресурс: <https://fpmi.spiralarms.org/> «Менеджер расписания». После исследования возможностей TestLink был создан проект тестирования web-приложения и осуществлен его запуск и произведено тестирование интернет-ресурса по заданным в проекте шагам.

Тест MSFBSU-1: TEST (Версия: 1)				
Автор: ndt080 - 05.11.2020 14:38:29				
Описание теста, цель, сценарий и исходное состояние программы:				
Проверка работоспособности приложения, его функционала. Приложение на стадии разработки				
Условия:				
Перейти на https://fpmi.spiralarms.org/				
Произвести регистрацию, после чего начать тест.				
#	Шаги	Ожидаемая реакция	Примечание	Execution Status
1	Ввести в поисковую строку https://fpmi.spiralarms.org/	Открытие приложения		Пройден
2	Произвести регистрацию	Успешная регистрация	Нет автоматического перехода к форме входа	Пройден
3	Произвести вход	Успешный переход на главную страницу		Пройден
4	Переход на страницу с расписанием	Открытие страницы с расписанием и отображение расписания		Пройден
5	Прокрутка карусели с блоками	Работоспособность карусели		Пройден
6	Добавление домашнего задания	После обновления блока с расписанием появление нового домашнего задания		Пройден
Способ прогона:				
Ожидаемая продолжительность прогона (мин):				
Важность:				
Требования:				
Ключевые слова:				
Execution Details				
Тестирование:				
Execution Result:				
Execution Mode:				
Продолжительность прогона (мин):				
Direct Link http://spiralarms.org/testlink/!nl.php?type=exec&id=14&apikey=81c9a5554a563f8beedd3ac4192c79952b4da7ce96d97d5cb5b561d06af6266b				

TestLink

ndt080 [admin]

MSFBSU

Программный продукт: MSFBSU-MANAGER SCHEDULE FAMCS BSU

MANAGER SCHEDULE FAMCS BSU / FREE (1)(0, 1, 0, 0)

COMMON TEST (1)(0, 1, 0, 0)

MSFBSU-1:TEST

Печать

Показать результаты всех прогонов

Import XML Results

Тест-план FREE

Версия (сборка) build

Группа тестов: COMMON TEST /

Последний прогон (любая сборка)

Дата : 05.11.2020 14:53:01 - Тестер : ndt080 - Версия (сборка) : build - Статус : Пройден

Последний прогон (данная сборка) - Версия (сборка) : build

Дата	Версия (сборка)	Тестер	Статус	Продолжительность (мин)	Версия	Режим запуска
05.11.2020 14:53:01	build	ndt080	Пройден	5.00	1	

Тестовый пример MSFBSU-1 :: Версия: 1 :: TEST

Ожидаемая продолжительность прогона (мин) :

Способ прогона : Ручной

Назначено : ndt080

Условия

Перейти на <https://fpmi.spiralarms.org/>

Произвести регистрацию, после чего начать тест.

TestLink

ndt080 [admin]

MSFBSU

Программный продукт MSFBSU/MANAGER SCHEDULE FAMCS BSU

MANAGER SCHEDULE FAMCS BSU / FREE (1)(0, 1, 0, 0)
COMMON TEST (1)(0, 1, 0, 0)
MSFBSU-1-TEST

Условия

Перейти на <https://fpml.spiralarms.org/>
Произвести регистрацию, после чего начать тест.

Описание теста: цель, сценарий и исходное состояние программы

Проверка работоспособности приложения, его функционала. Приложение на стадии разработки

#	Шаги	Ожидаемая реакция	Прогон	Примечание	Execution Status
1	Ввести в поисковую строку https://fpml.spiralarms.org/	Открытие приложения	Ручной		Пройден
<div> <div>Файл: 1 1 1</div> <div>Выбрать файлы</div> <div>Файл не выбран</div> </div>					
2	Произвести регистрацию	Успешная регистрация	Ручной		Пройден

TestLink

ndt080 [admin]

MSFBSU

Программный продукт MSFBSU/MANAGER SCHEDULE FAMCS BSU

MANAGER SCHEDULE FAMCS BSU / FREE (1)(0, 1, 0, 0)
COMMON TEST (1)(0, 1, 0, 0)
MSFBSU-1-TEST

Файл: 1 1 1

Выбрать файлы

Файл не выбран

3

Произвести вход

Успешный переход на главную страницу

Ручной

Пройден

Файл: 1 1 1

Выбрать файлы

Файл не выбран

4

Переход на страницу с расписанием

Открытие страницы с расписанием и отображение расписания

Ручной

Пройден

Файл: 1 1 1

Выбрать файлы

Файл не выбран

5

Прокрутка карусели с блоками

Работоспособность карусели

Ручной

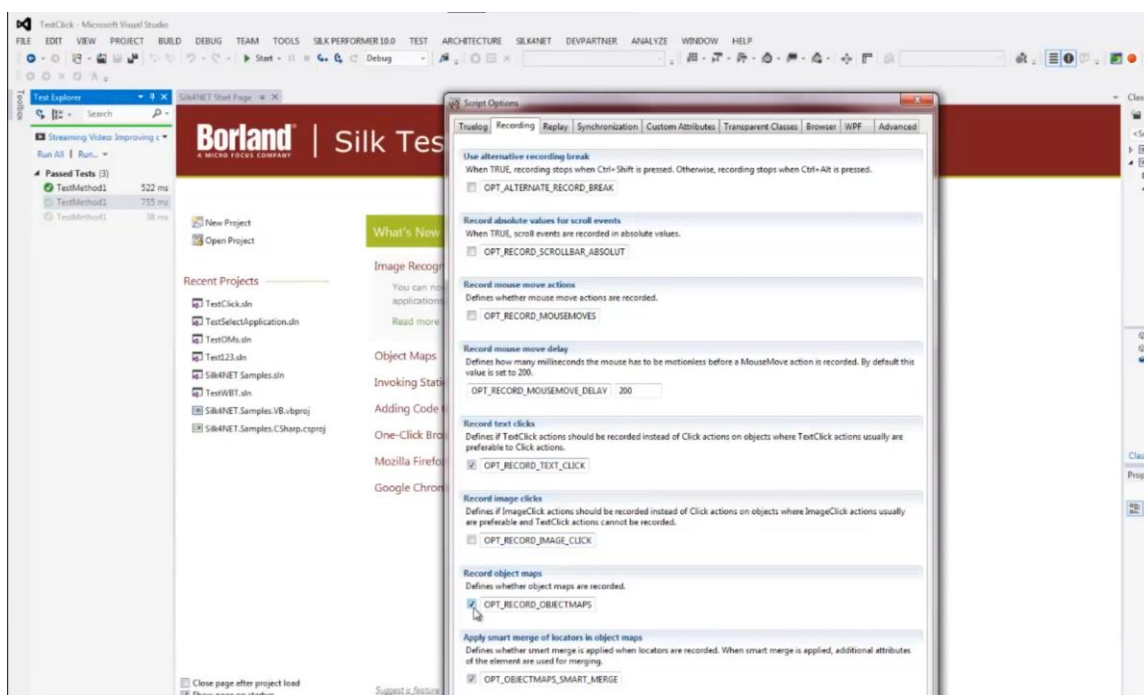
Пройден

ЛАБОРАТОРНАЯ РАБОТА №6. РАБОТА В СРЕДЕ SILKTEST

Цель работы: изучить порядок установки и произвести тестирование приложений в SilkTest.

Ход работы:

В ходе выполнения данной лабораторной работы были изучены возможности и функционал программного обеспечения SilkTest. Предметом тестирования было выбрано стандартное приложение Microsoft Windows – «Калькулятор». После исследования возможностей SilkTest был создан проект тестирования приложения, создан тест план, в котором в нескольких местах вызывается тест кейс. Осуществлен запуск проекта и произведено тестирование приложения.



The screenshot displays the Visual Studio IDE with a Selenium test project named 'TestClick'. The main code editor shows a C# test script for a calculator application. The script imports necessary namespaces and defines a test class 'TestCalc'. It includes a 'TestInitialize' method to set up the test environment and a 'TestMethod1' method that performs a series of clicks on the calculator interface to calculate the cosine of 45 degrees.

```

1 Imports System
2 Imports System.Collections.Generic
3 Imports Microsoft.VisualStudio.TestTools.UnitTesting
4 Imports SilkTest.Ntf
5
6 <SilkTestClass()> Public Class TestCalc
7
8     Private ReadOnly _desktop As Desktop = Agent.Desktop
9
10    <TestInitialize()> Public Sub Initialize()
11        Dim baseState = New BaseState()
12        baseState.Execute()
13    End Sub
14
15    <TestMethod()> Public Sub TestMethod1()
16        With _desktop.Window("@caption='Calculator'")
17            .SetActive()
18            .PushButton("[@priorlabel='0']{4}").Select()
19            .PushButton("[@priorlabel='4']{10}").Select()
20            .PushButton("[@priorlabel='45']{40}").Select()
21            .PushButton("[@priorlabel='0,70710678118654752440084436210485']{8}").Select()
22            .TextClick("4")
23            .TextClick("5")
24            .TextClick("cos")
25            .TextClick("CE")
26        End With
27    End Sub
28
29 End Class

```

The Test Explorer on the left shows four passed tests, all named 'TestMethod1', with execution times ranging from 11 ms to 522 ms. The background shows a calculator application window with the result '0,70710678118654752440084436210485' displayed.