**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](#_Toc56715128)

[ЛАБОРАТОРНАЯ РАБОТА №1. СОСТАВЛЕНИЕ СХЕМЫ ТЕСТИРОВАНИЯ ПРЕДМЕТА 4](#_Toc56715129)

[ЛАБОРАТОРНАЯ РАБОТА №2.1. ТЕСТИРОВАНИЕ ПРЕДМЕТА. СОЗДАНИЕ УЧЕБНОГО ТЕСТА С ОТВЕТАМИ ПРИ ПОМОЩИ HTML И JAVASCRIPT 5](#_Toc56715130)

[ЛАБОРАТОРНАЯ РАБОТА №2.2. ТЕСТИРОВАНИЕ ПРЕДМЕТА. ТЕСТИРОВАНИЕ ТРЕУГОЛЬНИКА 6](#_Toc56715131)

[ЛАБОРАТОРНАЯ РАБОТА №4. СОЗДАНИЕ ПРОЕКТОВ В TESTCOMPLETE 9](#_Toc56715132)

[ЛАБОРАТОРНАЯ РАБОТА №5.1. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ. ТЕСТОВАЯ ДОКУМЕНТАЦИЯ 10](#_Toc56715133)

[ЛАБОРАТОРНАЯ РАБОТА №5.2. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ В TESTLINK 13](#_Toc56715134)

# **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

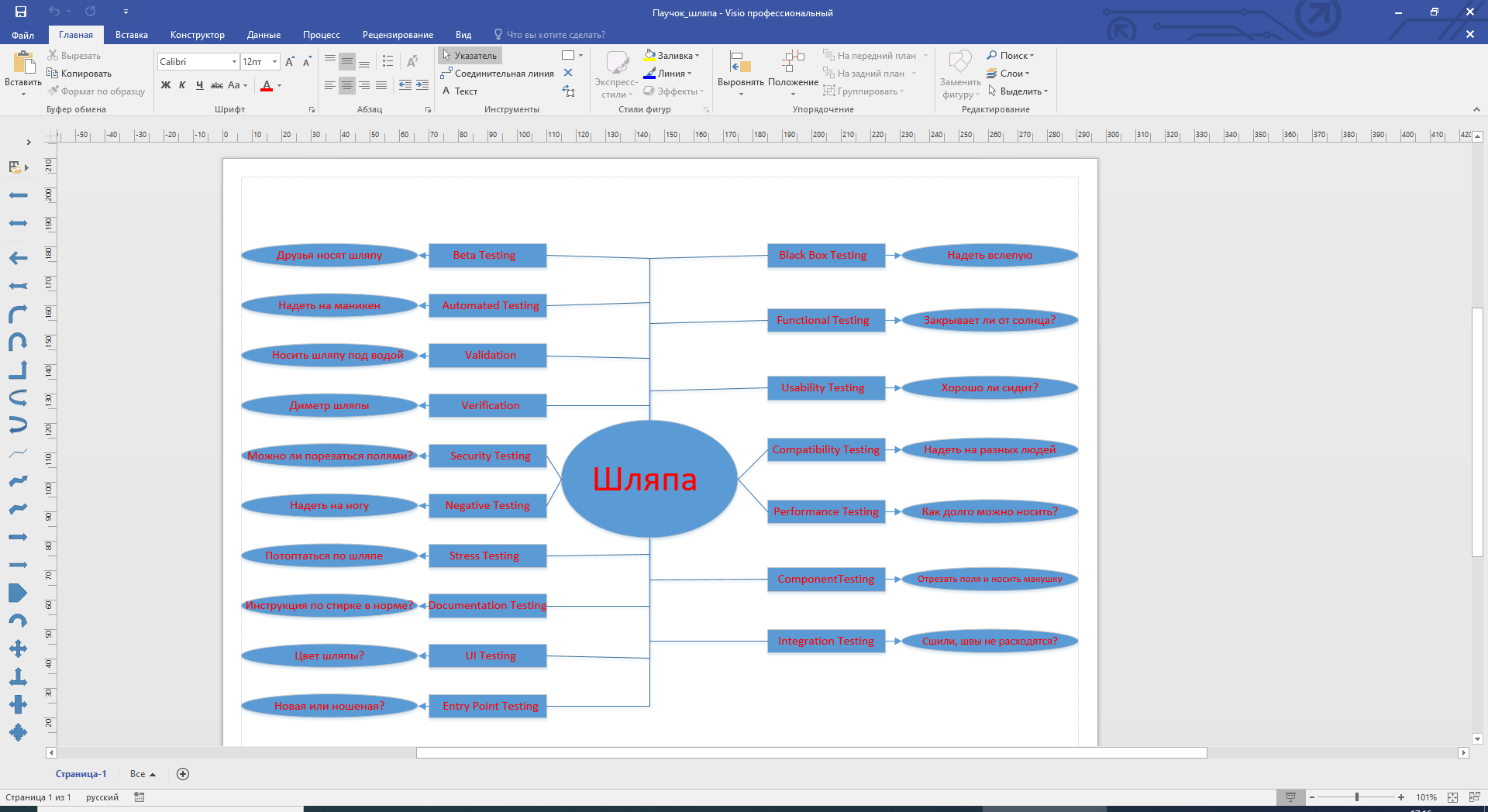
# **ЛАБОРАТОРНАЯ РАБОТА №1. СОСТАВЛЕНИЕ СХЕМЫ ТЕСТИРОВАНИЯ ПРЕДМЕТА**

**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.

.

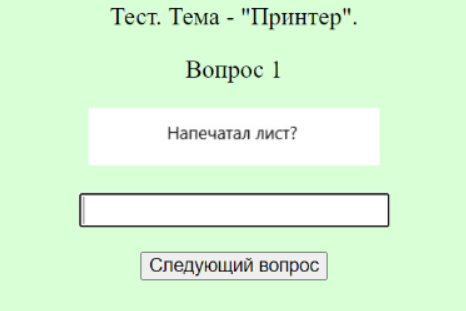
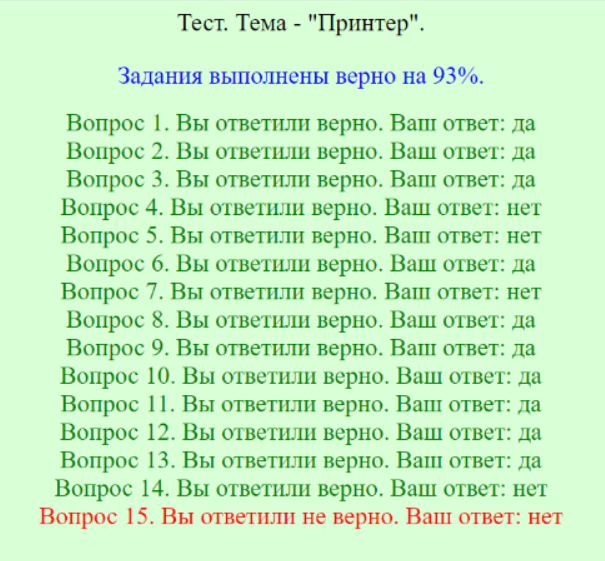
# **ЛАБОРАТОРНАЯ РАБОТА №2.1. ТЕСТИРОВАНИЕ ПРЕДМЕТА. СОЗДАНИЕ УЧЕБНОГО ТЕСТА С ОТВЕТАМИ ПРИ ПОМОЩИ HTML И JAVASCRIPT**

**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-sozdanie-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.



# **ЛАБОРАТОРНАЯ РАБОТА №2.2. ТЕСТИРОВАНИЕ ПРЕДМЕТА. ТЕСТИРОВАНИЕ ТРЕУГОЛЬНИКА**

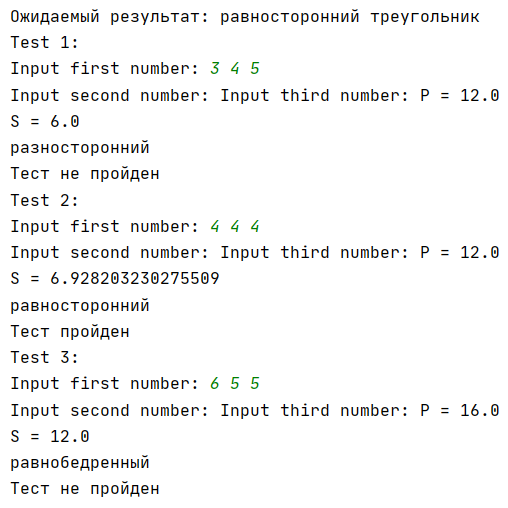
**Цель работы:** написать программу на Java (создание классов); cоздать тесты, покрывающие критерии по стратегии белого ящика (покрытие всех путей); cоздать тесты, покрывающие критерии по стратегии черного ящика (анализ граничных условий); в тестах указать входные данные, ожидаемые результаты; создать 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 результа ты прохождения теста – пройден тест или нет.

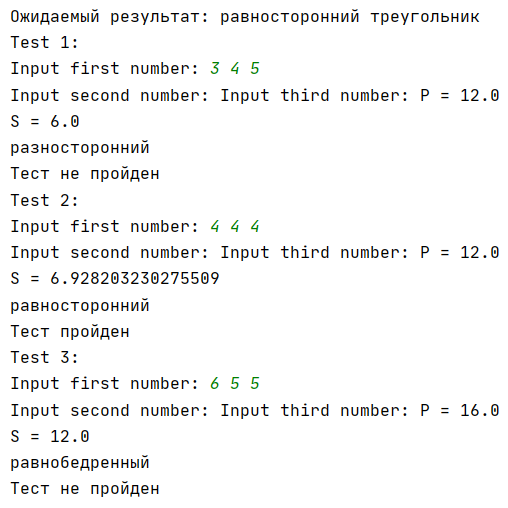
**Ход работы:**

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?»





**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 Applicaiton. 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.

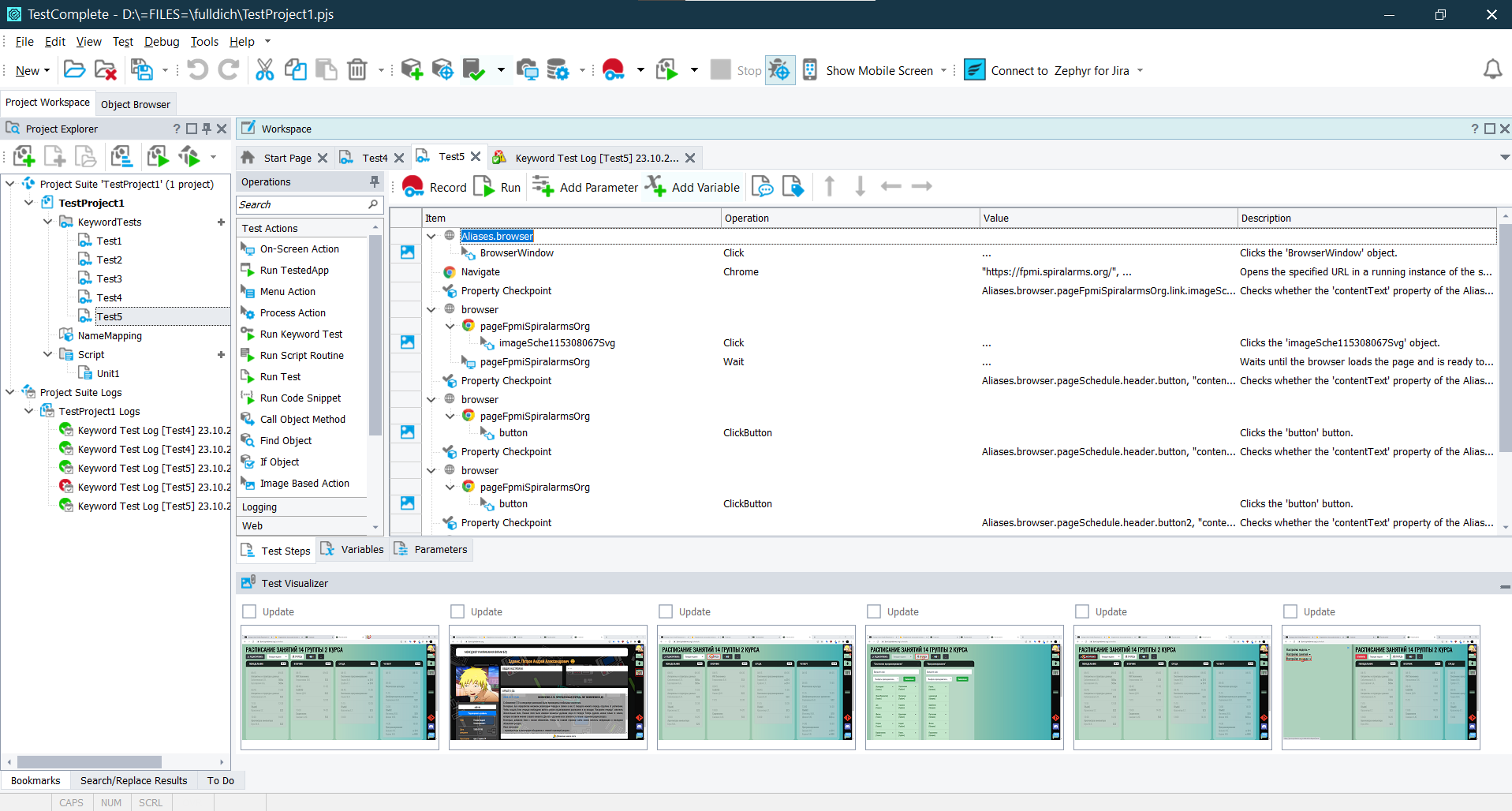


# **ЛАБОРАТОРНАЯ РАБОТА №4. СОЗДАНИЕ ПРОЕКТОВ В TESTCOMPLETE**

**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";

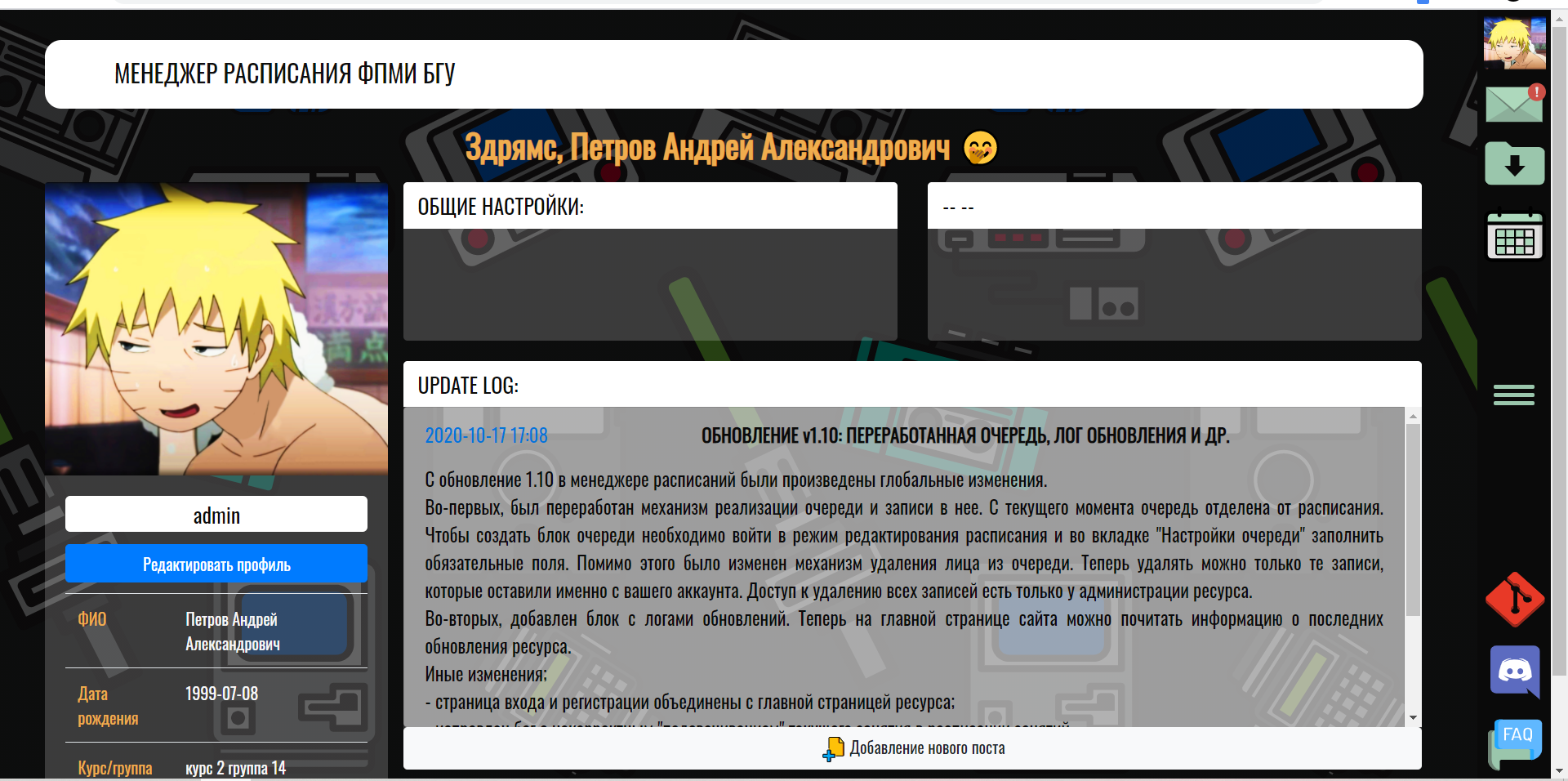
7. "работоспособность".

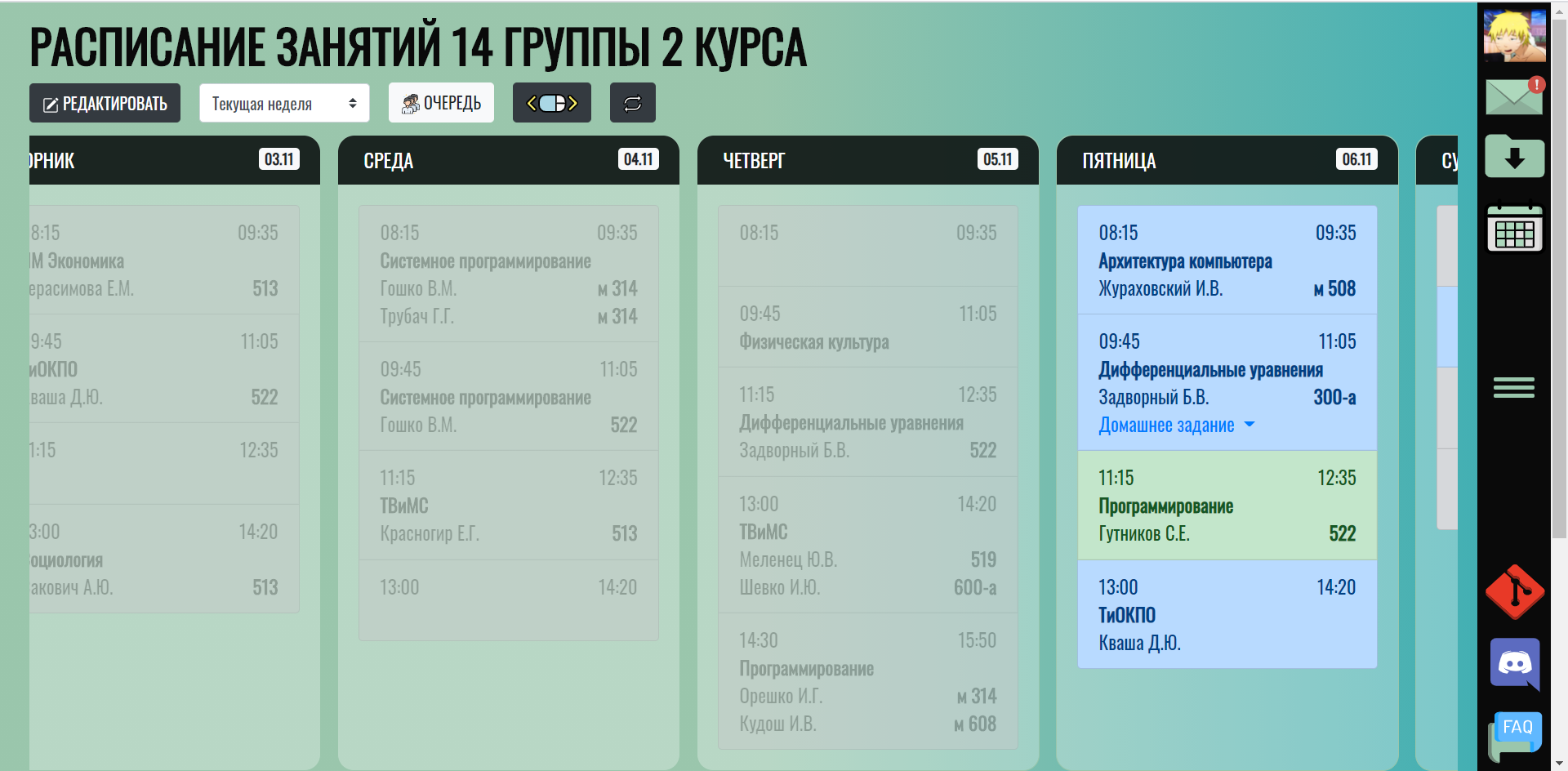
Исходя из этого, для выбранного мною веб-приложения была составлена следующая тестовая документация.

**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 Egde)

**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.**

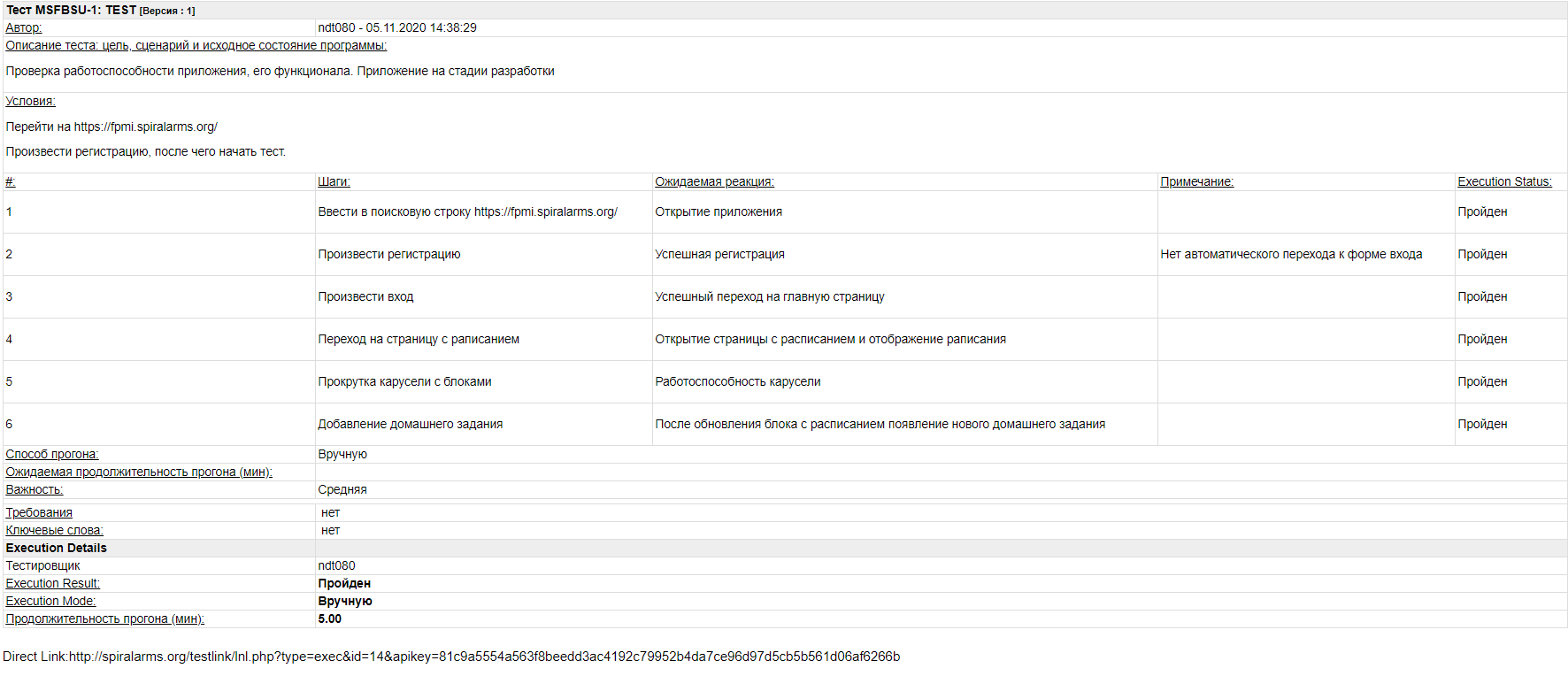
Приложение загружается очень быстро. Используется технология PWA и модульности компонентов, что позволяет пользователю догружать необходимый контент, а не загружать страницу полностью

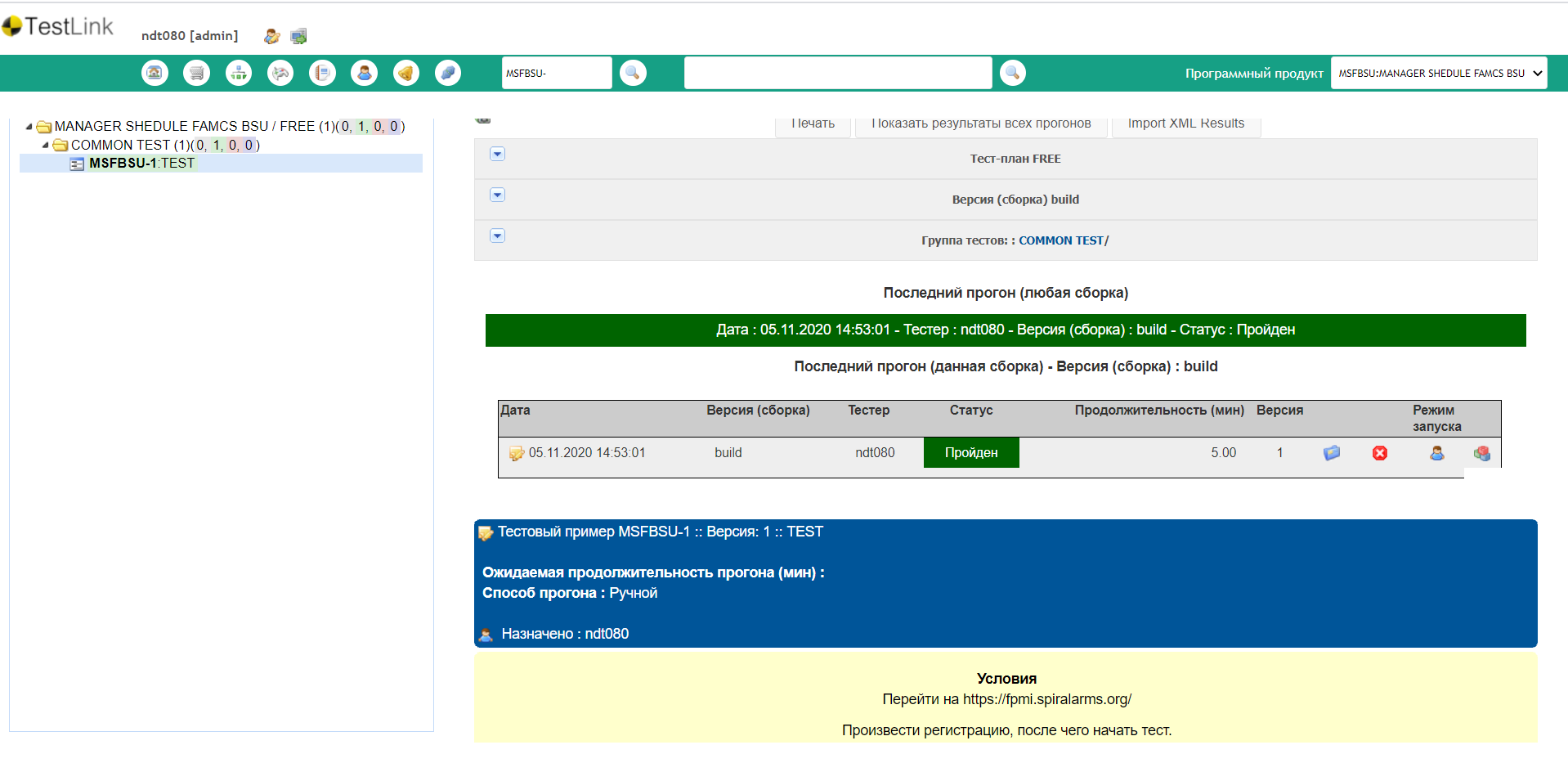
# **ЛАБОРАТОРНАЯ РАБОТА №5.2. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: РАЗРАБОТКА ТЕСТОВ В TESTLINK**

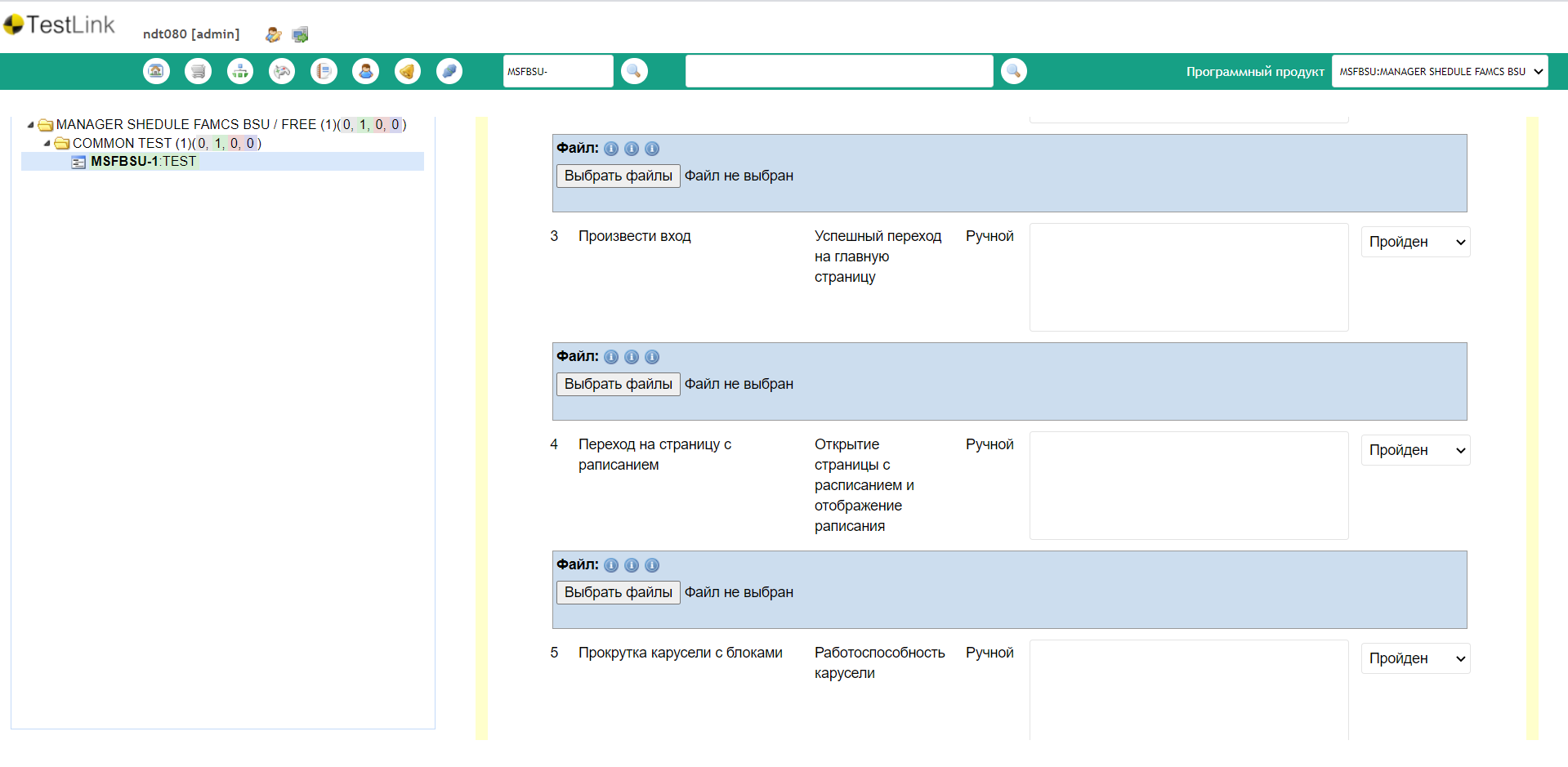
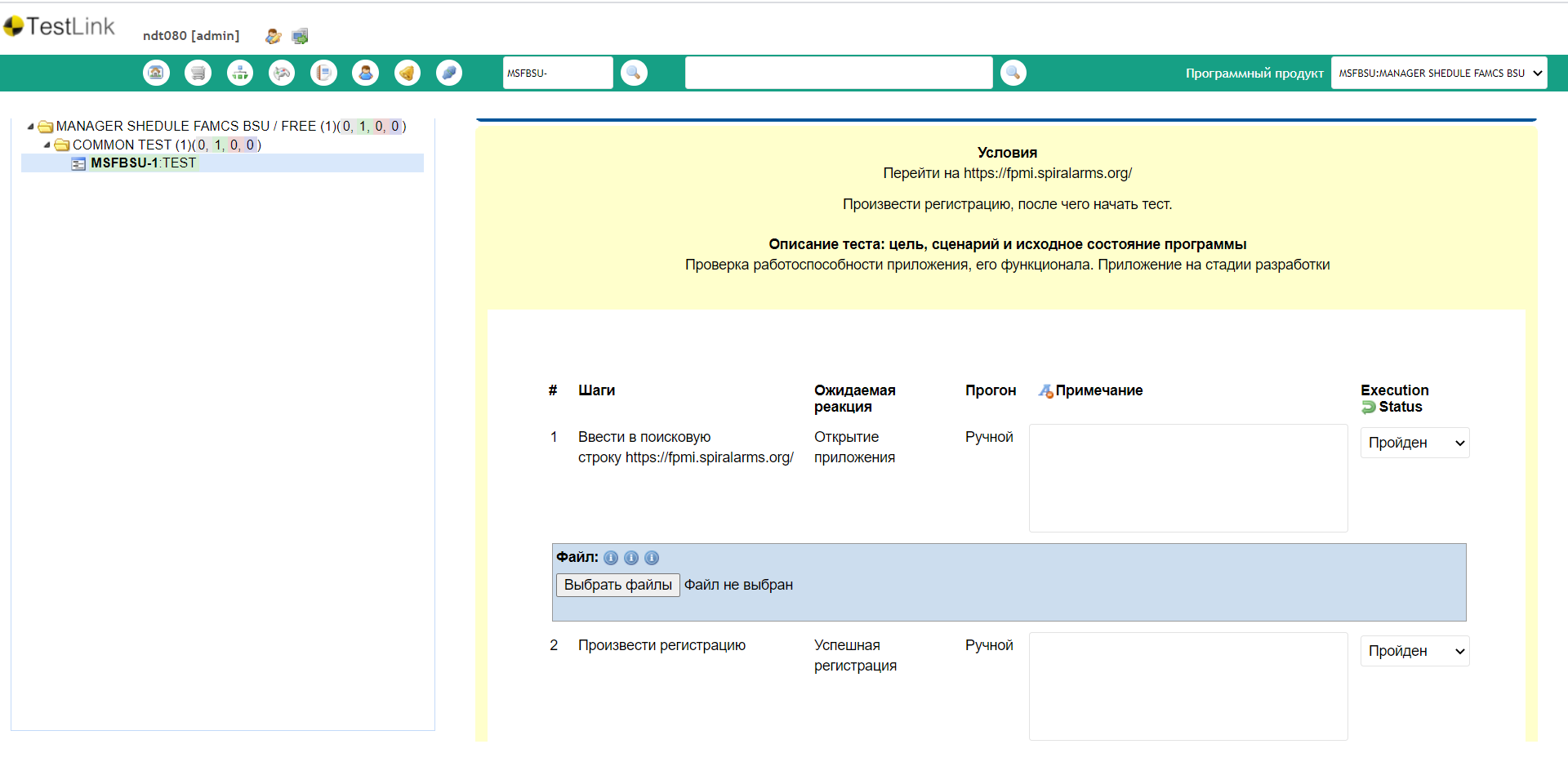
**Цель работы:** изучить порядок установки и произвести тестирование приложений в TestLink.

**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.







**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.

