Министерство образования и науки Российской Федерации

**Волжский политехнический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего**

**образования «Волгоградский государственный технический университет»**

**(ВПИ (филиал) ВолгГТУ)**

*Инженерно-экономический факультет*

Факультет «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

*Информатика и технология программирования*

Кафедра «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

**ЛАБОРАТОРНАЯ РАБОТА №6**

*Технология командной разработки ПС*

по дисциплине «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

**«Модульное тестирование. Построение приложений.»**

*(вариант №3)*

на тему\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Борзин Роман Юрьевич, Чурзина Екатерина Олеговна*

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(имя, отчество, фамилия)

*ВИП-408*

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(в баллах)

*доцент, к.т.н. Свиридова О.В.*

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (долж., инициалы и фамилия)

Волжский, 2018 г.

*Модульное тестирование* используется разработчиками для проверки правильности работы методов классов. Модульные тесты сохраняются в системе контроля версий и выполняются при каждом построении приложения. Кроме того, модульные тесты являются основой регрессионного тестирования, которое выполняется при добавлении новых возможностей или модификации приложения.

Тестирование класса Notification

Класс St реализует метод вставки addSt(). Данный метод реализуют размещение статьи в системе.

Спецификация данного тестового случая для тестирования метода addSt() на основе спецификации этого класса:

Название класса: addSt

Название тестового случая: add\_st\_test

Описание тестового случая: Тест проверяет правильность работы метода addSt() – добавление статьи на основе введенных пользователем данных.

Ожидаемый результат:

Перечисленным входным значениям должны соответствовать следующие выходные значения:

|  |  |  |  |
| --- | --- | --- | --- |
| Тест 1 | Тест 2 | Тест 3 | Тест 4 |
| Система добавила статью и продолжила работу | Система выдала сообщение: «Вы заполнили не все поля!» | Система выдала сообщение: «Вы заполнили не все поля!» | Система выдала сообщение: «Вы заполнили не все поля!» |

Код теста:

На основе спецификации был создан тестовый драйвер - метод order\_tester.

NSArray \*testCasesText = @[@" Название статьи ", @" Название статьи ", @"", @" Название статьи "];

NSArray \*testCasesZakazchik = @[@"Иванов Иван,", @"", @"Смирнов Дмитрий", @"Сидоров Петр"];

for (int i = 0; i < 4; i++) {

self.taskField.text = [testCasesText objectAtIndex:i];

self.stField.text = [testCasesSt objectAtIndex:i];

[self doneButton:nil];

}

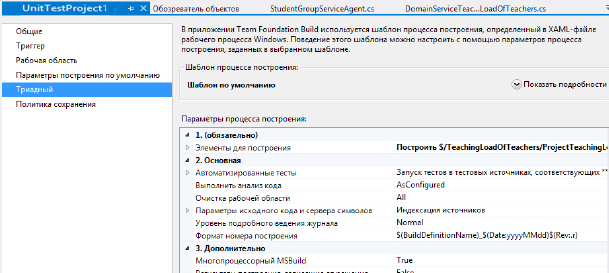
В методе st\_tester реализована вся функциональность теста. В данном случае для покрытия спецификации достаточно перебрать следующие комбинации входных данных: все данные введены корректно, ошибка в одной из составляющей входного массива, и получить соответствующие им сообщения с помощью метода add\_st(). Пары соответствующих значений заносятся в NSLog-файл для последующей проверки на соответствие спецификации.

Результаты тестирования формируются классом add\_st\_test, который производит сравнение ожидаемых результатов с фактическими.

После построения тестового проекта созданный тест будет отображаться в обозревателе тестов



Политика сохранения



Триадный определения построения

