Жуков Николай Николаевич
ассистент
ФГБОУ ВПО «Российский государственный
педагогический университет им. А.И. Герцена»
г. Санкт-Петербург

ПОДГОТОВКА БУДУЩИХ ИНЖЕНЕРОВ К ТЕСТИРОВАНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Аннотация: *в статье содержится описание процесса подготовки будущих инженеров к тестированию программного обеспечения с помощью системы автоматизированной проверки заданий и написания модульных тестов для Javascript-сценариев с использованием технологий Phantom.js, Mocha.js и Chai.*

Ключевые слова*: электронное обучение, e-learning, JavaScript, тестирование, система автоматизированного тестирования, модульное тестирование, юнит-тестирование, кодактор, бордкастинг, Mocha, Chai, Phantom.*

Тестирование является одним из этапов процесса разработки (software development) программного обеспечения. В течение этого процесса выявляются нештатные ситуации, при которых поведение создаваемой программы отличается от описанного в требованиях к ней (незапланированное поведение или возвращение ошибочного результата). Современный инструментарий программиста подразумевает активное использование модульных тестов (unit testing) для тестирования программных продуктов. Этот вид тестирования предполагает проверку корректности работы программного продукта, реализованного с использованием модульной архитектуры. При этом тесты необходимо написать для любой нетривиальной функции или метода. В последнее время широкое распространение получили методологии разработки, в которых заложена идея разработки через тестирование (например, TDD – test driven development и производная от неё BDD – behavior driven development). Процесс разработки кода в этих методологиях начинается с написания теста, определяющего требования к модулю. Этот процесс повторяется при добавлении нового функционала в программу.

Процесс автоматизации данного процесса привел к появлению множества библиотек и фреймворков, написанных для большинства современных языков программирования (например, Java, Javascript, PHP, Ruby, Python).

Автором данной статьи и его коллегами, обучающими веб-разработке на базе кафедры компьютерных технологий и электронного обучения РГПУ им. А.И. Герцена, активно используются элементы электронного обучения [1; 2] для преподавания различных дисциплин и различной педагогической деятельности. Необходимо отметить, что процесс организации проверки выполнения студентами лабораторных и практических работ вручную не представляется возможным, в силу большого их числа и разнообразия. Автоматизация этого процесса позволяет ускорить обратную связь между преподавателем и студентом, а также процесс корректировки и исправления кода.

В данной статье будет предложен подход к подготовке будущих инженеров к тестированию программного обеспечения, а также проверке выполненных ими практических и лабораторных работ по дисциплине «Программирование», обучение проводится на основе Javascript.

Данный подход в части проверки программ, написанных студентами, включает в себя: (1) использование инструмента «Кодактор» (кодактор.рф) для реализации заданий [3], (2) использование технологии Phantom.js [2] и (3) связки технологий Mocha.js и Chai для подготовки студентов к тестированию программ.

Опишем кратко указанные выше инструменты и алгоритм работы при их использовании.

1. Кодактор.рф – онлайн-редактор HTML, CSS и JavaScript-кода.

2. Phantom.js представляет собой Webkit-движок без графического интерфейса (headless) с открытым исходным кодом, реализованным в нем Javascript API и поддержкой современных веб-стандартов. Основная задача Phantom.js – имитация работы обычного браузера.

3. Mocha.js – современный фреймворк для тестирования, способный работать как с использованием отдельного сервера, так и в браузере. Chai – библиотека для организации различных типов тестов (например, равенство, эквивалентность, совпадение типов и т. д.)

Опишем процесс организации лабораторного или практического занятия.

Преподавателем формулируется задача, которую студенты должны решить. При этом тщательно описываются пары входных и выходных наборов значений (сигнатура или «кейс проверки») [2]. Пример варианта сигнатуры для задачи построения двоичного дерева приведен по ссылке: clck.ru/9cU22. Это позволяет обеспечить однозначность результата, ожидаемого от студента. Необходимое требование – сформулировать задачу таким образом, чтобы студент не имел возможности «подогнать» решение под ответ. Для каждой задачи преподаватель предоставляет шаблон борда, в котором предполагается выполнять решение.

Студент должен реализовать решение своей задачи в борде, созданным в инструменте «Кодактор» [3], преобразовать ответ в формат JSON и вернуть его в заранее оговоренное место внутри борда, а ссылку на борд представить преподавателю. Помимо реализации задания, студенту необходимо сопроводить решение набором тестов, реализованных с использованием связки Mocha и Chai.

Преподаватель, используя заранее заготовленные наборы сигнатур, Phantom.js и полученную от студента ссылку на борд тестирует правильность работы программы. Пример тестирующего скрипта приведен по ссылке: clck.ru/9cU24. Проверка с помощью Phantom.js различных вариантов сигнатур возможна при использовании GET-запросов к борду студента. В случае, если ответы студента и преподавателя для всех наборов сигнатур совпадают, а тесты проходят успешно, задание считается выполненным. В противном случае, преподаватель может осуществить анализ кода (code review) программы или тестов для того, чтобы дать какие-либо рекомендации по доработке.

Комбинация данных инструментов позволяет с одной стороны снизить нагрузку преподавателя, поскольку он не вынужден проверять множество решенных студентами лабораторных работ вручную, а с другой стороны, при выявлении неправильного или сомнительного решения, провести анализ кода.

Дальнейшая работа в данном направлении включает разработку системы (или надстройки над указанными выше фреймворками), позволяющую автоматизировать процесс сбора решений, их тестирования, отправки результатов оценивания студенту и выставления оценка. Такая система может быть интегрирована в LMS или представлять собой автономное решение с возможностью экспорта результатов в формате SCORM или Tin Can API.

Список литературы

1. Аксютин П.А. Опыт построения среды электронного обучения и ее использование для преподавания дисциплины «Информационные технологии» // Электронное обучение в вузе и в школе: Материалы сетевой международной научно-практической конференции. – СПб.: Астерион, 2014. – С. 28–31.

2. Государев И.Б. Электронное обучение веб-программированию на основе технологий тестирования JavaScript-сценариев / И.Б. Государев // Педагогический опыт: теория, методика, практика: Материалы III Междунар. науч.-практ. конф. (Чебоксары, 31 июля 2015 г.) / редкол.: О.Н. Широков [и др.]. – Чебоксары: ЦНС «Интерактив плюс», 2015. – С. 71–73.

3. Государев И.Б. Электронное обучение веб-разработке на основе бордкастинга и платформы edX // Современная педагогика: теория, методика, практика: Сборник материалов международной научной конференции. Россия, г. Москва, 27–28 февраля 2014 г [Электронный ресурс] / под ред. Д.Ю. Ануфриевой. – Электрон. текст. дан. (1 файл 1,3 Мб). – Киров: МЦНИП, 2014. – 150 с. – 1 электрон. опт. диск (CD-ROM). – С. 104–107.