



## ПРАКТИКА ПРИМЕНЕНИЯ UNIT-ТЕСТОВ ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Унучек Е.Н., Унучек Т.М., Комличенко В.Н., Пархименко В.А.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь,  
e.unuchek@gmail.com, t.unuchek@gmail.com, v.komlitchenko@gmail.com, uladimir.parkhimenka@gmail.com

Abstract. The possibilities of using Test Driven Development (TDD) techniques for students in teaching programming skills are considered.

Современное обучение становится массовым и требует все больших ресурсов для своей индивидуализации. Эта проблема характерна как для отдельных тренингов, так и для системы высшего образования в целом. Попытка сократить затраты на обучение приводит к поиску новых способов и технологий обучения с возможностью практически полностью отказаться от участия в нем преподавателей после разработки учебных курсов. Однако, сегодня это не представляется возможным в полной мере, поэтому все усилия по автоматизации обучения направлены лишь на его интенсификацию. Такие подходы призваны сократить затраты на одного обучаемого и обеспечить доступ большему количеству обучаемых к высококвалифицированным преподавателям.

В поиск возможных решений вовлечены и it-компании. Одна из интересных методик была предложена и апробирована компанией EPAM Systems, столкнувшейся с проблемой проверки огромного количества заданий кандидатов на участие и слушателей тренингов. Формат проведения обучения предполагал использование вебинаров, как средства доставки информационного контента. Лабораторные работы выполнялись каждым слушателем индивидуально. Аудитория участников тренинга охватывала больше тысячи человек. Однако, проверка выполнения лабораторных работ всех участников обучения, с точки зрения трудозатрат, представлялась немислимой. Данное ограничение удалось решить следующим способом: все участники получали заранее подготовленную единую структуру проекта, которая включает в себя коды функциональных тестов (unit-тесты), а также прототипы тестируемых методов (в виде классов или отдельных функций). Оказалось, что достаточно большое количество желающих принять участие в таком обучении не смогли справиться даже с самым первым и базовым заданием: настроить среду разработки и тестирования, несмотря на подготовленные заранее подробные инструкции. Это помогло отделить «случайных» слушателей тренинга.

Идея этого подхода позаимствована из техники разработки программного обеспечения «Разработка через тестирование» (англ. *test-driven development TDD*), суть которой состоит в повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемый функционал программного приложения, затем пишется код, который позволит пройти этот тест успешно [1].

Подобная практика с успехом может использоваться в практике выполнения лабораторных работ

студентами вузов по дисциплинам, связанным с изучением языков программирования. Наиболее результативным может быть применение данной методики в системе дистанционного обучения.

Однако заявленный подход имеет и ряд ограничений. Основные из них это:

- невозможность использования данного способа обучения для отработки навыков по проектированию архитектурных элементов систем, компьютерным сетям, базам данных и другим отдельным дисциплинам, связанным с ИТ и выходящими за изучение только лишь языков программирования;
- реальная возможность подачи чужих программных реализаций, выдаваемых за свои;
- большая трудоемкость составления заданий (разработки unit-тестов и каркаса приложений);
- сложность индивидуализации заданий;
- сложность обнаружения логических ошибок в реализации отдельных методов или функций;
- высока вероятность подбора реализации под контрольный тест;
- по-прежнему не представляется возможным отказаться от участия преподавателя, которое может проявляться в виде организации отдельных консультаций, как с использованием дистанционных коммуникационных технологий, так и реального общения в аудитории.

Познакомиться с реализацией данного подхода можно на таких ресурсах, где в игровой форме проводится обучение наиболее популярным языкам программирования, например:

- <https://www.codecademy.com/>;
- <https://checkio.org/>;
- <https://www.codewars.com/>.

Дальнейшее развитие такого формата проведения отдельных лабораторных работ по языкам программирования видится в создании специализированной инфраструктуры и программного обеспечения, которое интегрируется с системами непрерывной поставки новых версий программных приложений (Continuous integration system), с удаленными репозиториями-хранилищами исходного кода (например, Git), а также расширенными возможностями рассылки уведомлений и построения отчетности.

### Литература

1. Сонмез, Д. Путь программиста / Д. Сонмез. – СПб: Питер, 2016. – 448 с.