

разование: материалы XIII Международной научно-практической конференции, Екатеринбург, 1–5 марта 2021 г. Екатеринбург: Российский государственный профессионально-педагогический университет, 2020. С. 416–419.

УДК 378.016:004.415.53

**А. А. Штанюк**

**A. A. Shtanyuk**

***ФГБОУ ВО «Нижегородский государственный технический университет им. П. Е. Алексеева», Нижний Новгород  
Nizhny Novgorod State Technical University, Russia, Nizhny Novgorod  
ashtanyuk@nntu.ru***

## **КРАТКИЙ ОБЗОР ИНСТРУМЕНТОВ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ**

### **SYNOPSIS OF AUTOMATED TESTING TOOLS IN PROGRAMMING EDUCATION**

**Аннотация.** В статье рассматриваются возможности использования инструментов тестирования программного обеспечения в учебном процессе с упором на автоматизированное тестирование в рамках курсов программирования.

**Abstract.** The article examines the use of software testing tools in the educational process, with an emphasis on automated testing in programming courses.

**Ключевые слова:** учебный процесс; программирование; автоматизированное тестирование.

**Keywords:** education; programming; automated testing.

При подготовке специалистов в области информационных технологий, обучение программированию занимает ключевое место. Данное направление профессиональной деятельности не только расширяет набор профессиональных компетенций выпускников вузов, но и способствует развитию мышления для решения широкого спектра профессиональных задач [1].

Написание программного кода в рамках курсов программирования ставит неизбежный вопрос о проверки работоспособности этого кода, исследование его на корректность и полноту в свете поставленных задач. Современный профессиональный подход основан на обязательной процедуре тестирования программного кода, в связи с чем возникает вопрос о проведении такой процедуры.

Тестирование может производиться различными инструментами и в рамках разнообразных инфраструктурных решений. Самый простой вариант: ручной, - когда преподаватель собственноручно запускает код студента и пытается оценить его работу в процессе запуска. Такой подход можно назвать традиционным, но в современных условиях он не вызывает интереса по следующим причинам:

1. Высокая трудоёмкость ручного труда, особенно тяжёлая при больших групп студентов.

2. Невозможность покрыть функционал сложной программы в виду отсутствия пакета модульных тестов, что снижает вероятность обнаружения большого числа не синтаксических ошибок.

3. Длительное время ожидания результатов для обучаемого, необходимость информирования студентов преподавателем.

Профессиональный подход заключается в разработке пакета модульных тестов и автоматизации их запуска. При наличии пакета с тестами, сохраняется возможность их ручного применения, но это нежелательно, так как трудозатраты остаются на высоком уровне. Автоматизация процесса тестирования предполагает, что тестирование начинается автоматически при поступлении исходного кода программы на проверку, осуществляется без вмешательства человека и также самостоятельно завершается с генерированием финального отчета.

**Автоматизированное тестирование** предполагает использование специального программного обеспечения (помимо тестируемого) для контроля выполнения тестов и сравнения ожидаемого фактического результата работы программы. Этот тип тестирования помогает автоматизировать часто повторяющиеся, но необходимые для максимизации тестового покрытия задачи [2].

После создания автоматизированных тестов, их можно в любой момент запустить снова, причем запускаются и выполняются они быстро. Таким образом, если есть необходимость частого повторного прогона тестов, что характерно для учебного процесса, значение автоматизации для упрощения сопровождения проекта и снижения его стоимости трудно переоценить.

На рисунке приведены качественные зависимости стоимости процесса тестирования от числа необходимых тестов. В качестве основной единицы стоимости выступает время, затрачиваемое на тестирование программы.



Рисунок – Зависимость стоимости тестирования от числа тестов

Существует несколько основных видов автоматизированного тестирования:

– автоматизация тестирования кода (Code-driven testing) – тестирование на уровне программных модулей, классов и библиотек (модульное тестирование);

– автоматизация тестирования графического пользовательского интерфейса (Graphical user interface testing) – специальная программа (фреймворк автоматизации тестирования) позволяет генерировать пользовательские события – нажатия клавиш, клики мышкой, и отслеживать реакцию программы на эти действия – соответствует ли она спецификации.

– автоматизация тестирования API (Application Programming Interface) – программного интерфейса программы. Тестируются интерфейсы, предназначенные для взаимодействия, например, с другими программами или с пользователем. Здесь опять же, как правило, используются специальные фреймворки.

Мы рассмотрим модульное тестирование двух классов программ: классических консольных, не обладающих графическим интерфейсом (наиболее распространенный тип программ для обучающихся в вузе) и веб-приложений, использующих окно веб-браузера в качестве устройства ввода/вывода информации.

Необходимо отметить, что для организации процесса автоматизированного тестирования в рамках учебного курса обычно требуется использовать системы контроля версий, где будет храниться исходный код программ [3].

В качестве платформ для автоматизации процесса тестирования консольных программ рассмотрим следующие:

1. Yandex Contest
2. GitHub
3. GitLab

Yandex Contest [4] изначально разрабатывался как средство для организации и проведения соревнований по программированию. В нем имеются средства для формулировки задач, сценарии построения готовых программ и возможность загрузки тестов для проверки. Имеется встроенная поддержка более 20 языков программирования. Платформой пользуются более 3000 студентов каждый год и договоры о партнерстве с Яндекс заключили более 10 университетов.

Наиболее функционально богатой платформой из вышеперечисленных можно назвать GitHub [5]. Она является профессиональным инструментом для многих миллионов разработчиков и известна по всему миру.

Основным механизмом для построения программ и запуска тестов в GitHub является GitHub Actions. Сценарий для работы этого средства размещается в репозитории вместе с заданием и содержит простые команды. По этому сценарию GitHub создает виртуальный контейнер Docker, устанавливает в него необходимое программное обеспечение, выполняет построение учебной программы и ее запуск вместе с тестами. Вся процедура создания и использования контейнера с программой до получения результатов прогона занимает всего несколько минут и результаты тестирования доступны преподавателю и студентам.

Платформа GitLab [6] обладает схожими функциональными характеристиками с GitHub.

Для тестирования веб-приложений могут использоваться многочисленные фреймворки, среди которых наиболее популярным можно назвать Selenium [7]. Тестовые сценарии, которые можно писать на многих распространенных языках программирования, выполняют команды, позволяющие устанавливать соединение с серверной частью веб-приложения, производить авторизацию, обращаться к элементам веб-страницы и формам, проверяя функционал приложения. При этом может моделироваться деятельность пользователя, который через браузер обращается к приложению, вводит информацию и получает отклик от сервера. Автоматизация подобной деятельности существенно экономит время и устраняет неизбежную невнимательность человека, выполняющего ручной прогон тестов.

Нагрузочное тестирование веб-сайтов обычно выполняют с помощью специальной утилиты jMeter [8]. С помощью этого инструмента удобно проверять настройки веб-серверов и их общую производительность с целью оптимизации использования ресурсов. Это может быть использовано в курсах по веб-программированию и администрированию информационных систем.

Таким образом, в учебном процессе можно использовать достаточно богатый набор инструментов для тестирования программного кода и системных настроек.

### Список литературы

1. Винник В. К., Штанюк А. А. Информационно-проектный метод при подготовке будущих специалистов в сфере информационных технологий (с использованием системы MOODLE) // *Фундаментальные исследования*. 2015. № 2-23. С. 5183–5186.
2. Штанюк А. А. Системы управления проектами для поддержки учебного процесса // *Современные информационные технологии и ИТ-образование*. 2012. № 8. С. 380–385.
3. Мартюкова Е. С. О внедрении процесса автоматизации тестирования в различных методологиях разработки программного обеспечения // *Новые информационные технологии в автоматизированных системах*. 2019. № 22. С. 462–465.
4. Yandex.Contest. URL: <https://contest.yandex.ru/edu> (дата обращения: 10.03.2022).
5. GitHub. URL: <https://github.com> (дата обращения: 10.03.2022).
6. GitLab. URL: <https://gitlab.com> (дата обращения: 10.03.2022).
7. Selenium. URL: <https://www.selenium.dev> (дата обращения: 21.02.2022).
8. jMeter. URL: <https://jmeter.apache.org> (дата обращения: 28.03.2022).