



## ПОСТРОЕНИЕ ИНСТРУМЕНТА ДЛЯ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ ПРОГРАММ

Инструменты для автоматизированного тестирования приложений позволяют ускорить процесс ввода в эксплуатацию программных продуктов и повысить их качество. Многие из существующих приложений для тестирования программного обеспечения дороги и довольно сложны для начинающих специалистов. В статье описана созданная на кафедре АВТ сравнительно простая для освоения система для функционального тестирования приложений, предназначенная для применения в учебных целях.

Функциональное тестирование, автоматизированное тестирование, программный продукт, тест-кейс, синтаксический анализатор.

В настоящее время программное обеспечение (ПО) является неотъемлемой частью производственных процессов и иной хозяйственной деятельности. Численность организаций, занимающихся разработкой программных продуктов, существенно выросла. Созданные приложения постоянно совершенствуются: наращивается их функционал, улучшается интерфейс, повышается информационная безопасность и т.п. Тестирование – обязательный этап процесса разработки программного обеспечения. Оно необходимо не только для контроля соответствия всем требованиям добавленной функции, но и для проверки того, что внесенные изменения не снизили работоспособность системы в целом. В связи с этим объем тестирования резко увеличивается. Ускорить процесс ввода в эксплуатацию программного продукта и повысить его качество позволяет автоматизация тестирования [1]. Основопологающие идеи автоматизированного тестирования программного обеспечения, применяемые в ходе его приемы, методы и инструментальные средства, инструкции по эффективному их использованию, примеры успешных отраслевых реализаций автоматизации тестирования описаны в неоднократно переизданной и пользующейся спросом книге [2].

Вопросам обеспечения качества программных приложений и проблемам автоматизации тестирования посвящено множество публикаций. Например, в работе [3] рассматриваются вопросы тестирования программ логического управления электроавтоматикой станков. В работе [4] показана актуальность разработки методов информационной поддержки принятия решений в ходе тестирования программного обеспечения при их сертификации и автоматизации обработки экспертной информации, полученной в ходе испытаний при оценке характеристик качества программ.

Исследования подходов к тестированию ПО в организациях Российской Федерации регулярно проводит компания Перфоманс Лаб. На основе результатов опроса руководителей ИТ-подразделений порядка трехсот организаций, работающих в разных предметных областях, составляется аналитический отчет [5],

включающий комментарии экспертов о выявленных тенденциях и изменениях в области тестирования и обеспечения качества ИТ-продуктов в России. Опросы показывают постоянный рост количества запросов на тестирование ПО. В качестве цели работы отдела качества 80 % участников исследования 2020–2021 гг. указали повышение качества ИТ-продуктов, 69 % – повышение удовлетворенности пользователей. Большинство респондентов (82 %) используют для проверки качества разрабатываемых продуктов автотесты и уделяют внимание развитию компетенций по автоматизированному тестированию у своих специалистов в области функционального тестирования. По мнению специалистов Перфоманс Лаб, автоматизация позволяет обеспечить максимальное тестовое покрытие и эффективность применения тест-кейсов, приводит к оптимизации затрат, росту производительности труда и уменьшению сроков вывода продукта на рынок.

Об актуальности проблемы говорит и наличие на рынке множества инструментов автоматизированного тестирования. Например, для функционального и регрессионного тестирования применяются следующие:

- MercuryQuickTest использует концепцию тестирования на основе ключевых слов. К его достоинствам относят удобный и понятный пользовательский интерфейс для создания тестов без ручной правки скрипта;

- MercuryWinRunner направлен на тестирование графического интерфейса пользователя. Он автоматически собирает, проверяет и воспроизводит действия пользователей (таким образом создается тестовый сценарий), что позволяет выявлять дефекты;

- SegueSilkTest также позволяет тестировщикам без кодирования создавать автоматизированные тесты. Он работает с различными технологиями, включая AJAX, web, mobile web, Java, .NET и другие.

Популярны платные инструменты RationalFunctionalTester, TestComplete, Rational Robot [6]. Из бесплатных можно выделить SoapUI (программа с открытым исходным кодом, лицензией GNU и реализацией на языке Java) и Selenium (Java-приложение, анализирующее файлы определенной структуры для

нахождения в них команд манипуляции браузером и команд выполнения определенных действий и проверок).

По наблюдению исследователей, в российских компаниях для тестирования используют преимущественно зарубежные инструменты. Однако число компаний и организаций, которые ограничивают их использование, по сравнению с предыдущим отчетным периодом, увеличилось на 8 % [7]. Например, «М.Видео – Эльдorado» еще в 2021 году перешла на российское ПО для тестирования – Allure EE (Allure), которое полностью сравнимо по функционалу с западными аналогами при более низкой стоимости использования [8]. Собственные инструменты для тестирования ПО разрабатывает «Яндекс»: у компании есть свои TMS (Test Management System), трекер задач и ошибок, системы автоматизации функционального и нагрузочного тестирования и т.п. Создание инструментов автоматизированного тестирования ведется и на базе вузов [9, 10]. В 2022 году на фоне ухода из России иностранных разработчиков ПО острее стал вопрос замещения широко применяющихся импортных инструментов отечественными. Есть проблема и в недостатке специалистов для тестирования ПО. Особенно дефицит квалифицированных кадров наблюдается в области автоматизации функционального и нагрузочного тестирования. По состоянию на начало ноября 2022 года, на рекрутинговом портале HH.ru было размещено 2 282 вакансии в категории «Тестировщик» [8], при этом в 1 115 из них в список обязанностей тестировщика включено функциональное тестирование, в 720 – автоматизированное и в 375 – автоматизированное функциональное тестирование.

Будущий специалист по тестированию программных продуктов наряду с прочими во время обучения должен приобрести навыки применения методов и инструментов тестирования. На кафедре Автоматики и вычислительной техники Вологодского государственного университета неоднократно предпринимались попытки в лабораторном практикуме по дисциплине «Тестирование программного обеспечения» обучить студентов работе с каким-либо готовым

инструментом для функционального тестирования (например, с Selenium). Опыт оказался неудачным: студентам не хватало времени на освоение программы. Многие из существующих приложений для тестирования программного обеспечения довольно сложны для начинающих специалистов.

Таким образом, для целей обучения необходим достаточно простой инструмент для функционального тестирования приложений, обладающий основными возможностями перечисленных выше систем.

Задача функционального тестирования – поиск несоответствий полученного программного продукта функциональным требованиям, указанным в техническом задании на его разработку. Каждый сценарий включает исходные данные (и/или действия пользователя) и ожидаемые результаты и должен имитировать эксплуатацию программного продукта в реальных условиях. Не следует забывать проверку работоспособности приложения и на неправильных входных данных. В ходе подготовки тест-кейсов необходимо сразу определить, какие из них предназначены для автоматизированного тестирования. При функциональном тестировании автоматизация целесообразна в случаях тестирования часто используемых функций в различных ситуациях, рутинных операций, проверки математических расчетов и т.п.

Обычно функциональное тестирование проводится на двух уровнях: компонентное (модульное) и интеграционное (системы в целом). Так как времени на выполнение лабораторных работ отводится сравнительно мало, ограничим возможности создаваемого инструмента автоматизацией тестирования первого уровня. В список его функций должны войти:

- возможность создавать, редактировать и сохранять тест-кейсы;
- возможность загружать тестируемый код (подключаться к тестируемому приложению);
- возможность создавать сценарии тестирования (задавать последовательность выполнения тест-кейсов);
- возможность получать и просматривать отчеты по результатам тестирования.

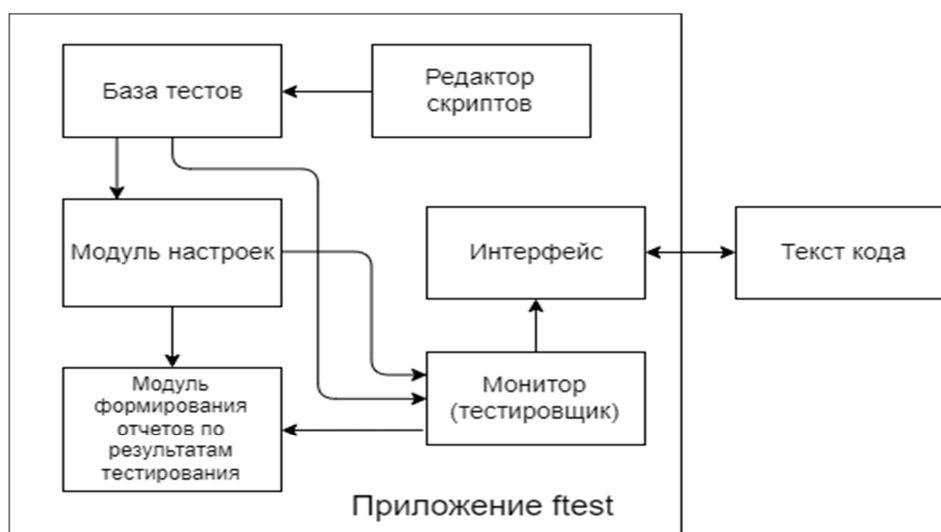


Рис. 1. Структурная схема инструмента для функционального тестирования

Предлагается следующая структура инструмента для функционального тестирования (рис. 1). Для создания и редактирования тест-кейсов предназначен Редактор скриптов. Созданные тест-кейсы будем сохранять в Базе тестов. Последовательность извлечения из Базы тестов и выполнения тест-кейсов можно будет задать с помощью Модуля настроек. Там же указываем, какую информацию следует включить в отчет по результатам тестирования. Компонент Интерфейс отвечает за взаимодействие создаваемой системы с тестируемым приложением и должен обеспечить подключение к приложению проверяемого кода. Задача Монитора – пропустить через проверяемую программу заданные тесты.

Важным элементом Монитора является синтаксический анализатор (парсер), который должен обработать входной скрипт и получить из него входные и выходные данные, в соответствии с которым будет проводиться тестирование. Процесс парсинга включает сканирование исходного массива информации, выделение семантически значимых единиц по заданным параметрам и конвертацию полученных данных в формат, удобный для изучения и дальнейшего использования. Объектом парсинга может быть любая грамматически структурированная система,

записанная на естественном языке, языке программирования, математическими выражениями и т.д.

Лексемами тестового скрипта будут цифры, числа, слова и словосочетания как в качестве входных данных, так и в качестве ожидаемых результатов. В инструменте для автоматизированного тестирования нет необходимости выполнять вычисления, но надо задать правильную последовательность входных данных. Поэтому запоминаем полученные значения в очередь – структуру данных, доступ к элементам которой контролируется по правилу FIFO («Первым пришел – первым ушел»). После запуска тестируемой программы получаем выходные данные в файл или считываем сразу в строку. Затем сравниваем выходные данные, выделенные из скрипта, и значения, полученные в результате работы тестируемой программы.

Первая версия инструмента для функционального тестирования реализована на языке программирования C++ и позволяет проверять программы, написанные на том же языке (имеют расширение \*.cpp). На рисунке 2 представлено главное окно разработанной программы. Оно содержит меню, поле для ввода тест-кейса, несколько кнопок и переключателей.

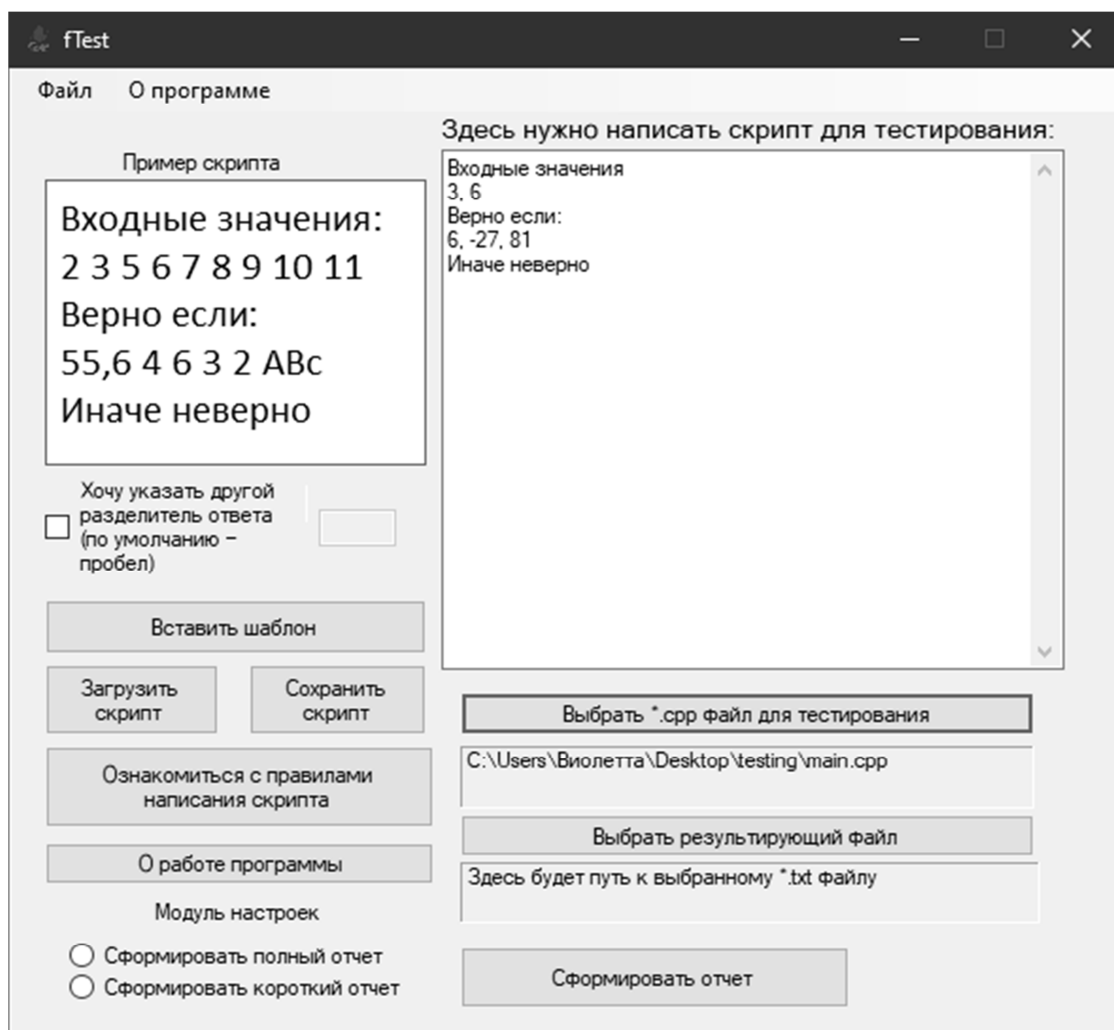


Рис. 2. Главное окно инструмента для функционального тестирования

После запуска приложения пользователь выбирает при помощи кнопок и всплывающих окон \*.cpp файл с программой, предназначенной для тестирования. Тест-кейс можно написать в окне редактора скриптов непосредственно перед началом тестирования или загрузить из хранилища тестов. В данной версии приложения готовые скрипты хранятся в виде файлов в формате \*.txt. Синтаксический анализатор обрабатывает заданный скрипт и получает из него входные данные и ожидаемый результат, в соответствии с которым будет проводиться тестирование. В настоящее время настройка отчета о тестировании осуществляется переключателем «Сформировать краткий (или полный) отчет». Во втором случае кроме информации о пройденном тесте и результате тестирования дополнительно выводится длительность тестирования и занимаемый объем памяти. Процесс тестирования запускается кнопкой «Сформировать отчет».

Например, нужно протестировать программу, в которой вводится два целых числа и вычисляется наибольшее из них, куб их разности и квадрат их суммы. Проверим все три функции. Предполагаем, что на входные значения 3, 6 программа вернет следующие результаты: 6, -27, 81. Тест-кейс представлен в окне редактора на рисунке 2. Следует заметить, что входные данные перечисляются в порядке присвоения значений переменным, а ожидаемый результат – в последовательности получения (вывода) результата. Файл со сформированным кратким отчетом, открытый в программе Блокнот, представлен на рисунке 3.

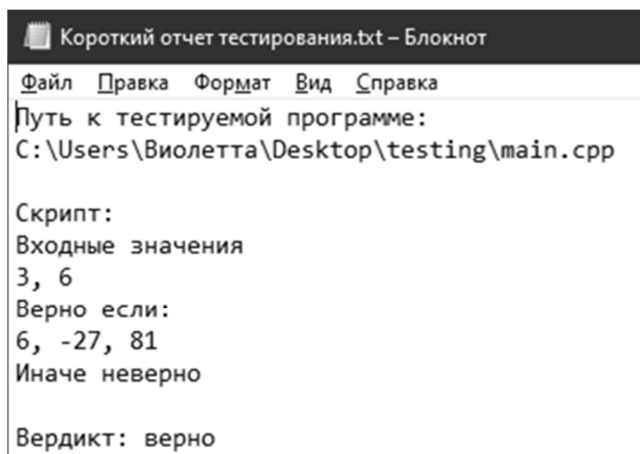


Рис. 3. Короткий отчет тестирования

В записи тест-кейса можно использовать логические выражения: знаком «+» обозначается логическое сложение, знаком «\*» – логическое умножение. Например, по скрипту «Входные значения 3, 6 + 6, 3 Верно, если: 6, -27, 81 + 6, 27, 81 Иначе неверно» через программу будет пропущено четыре теста.

Таким образом, программа показала свою работоспособность. Предполагается, что данный инструмент позволит студентам понять, как работают средства автоматизированного тестирования и освоить основные приемы работы с ними. Планируется расширить возможности программы и увеличить количество языков на которых могут быть написаны тестируемые программы, улучшить интерфейс.

### Литература

1. Сергушичева, А. П. Тестирование программно-го обеспечения : учебное пособие / А. П. Сергушичева. – Вологда : ВоГУ, 2022. – 88 с.
2. Дастин, Э. Автоматизированное тестирование программного обеспечения: Внедрение, управление и эксплуатация / Э. Дастин, Д. Рэшка, Д. Пол. – Москва : ЛОРИ, 2003. – 567 с.
3. Деркач, Е. В. Подход к проведению тестирования программ логического управления электроавтоматикой станков / Е. В. Деркач, Р. А. Нежметдинов // Вестник МГТУ «Станкин». – 2020. – № 4 (55). – С. 78–83.
4. Бурый, А. С. Тестирование качества программного обеспечения в процессе его сертификации / А. С. Бурый // Правовая информатика. – 2019. – № 1. – С. 46–55.
5. Russia Quality Report: аналитический отчет по российскому рынку тестирования программного обеспечения. – URL: <https://www.performance-lab.ru/rqr/> (дата обращения: 09.12.2022). – Текст: электронный.
6. Регрессионное тестирование: подборка инструментов. – URL: <https://testengineer.ru/regressionnoe-testirovanie-podborka-instrumentov/> (дата обращения: 23.11.2022). – Текст : электронный.
7. Тестирование программного продукта. – URL: <https://www.tadviser.ru/index.php/> (дата обращения: 06.12.2022). – Текст : электронный.
8. Как меняются инструменты и подходы к тестированию ПО в крупном российском бизнесе. Обзор TAdviser. Дата публикации 2022/12/09. – URL: <https://www.tadviser.ru/index.php/> (дата обращения: 09.12.2022). – Текст : электронный.
9. Кириллов, С. С. Ускорение автоматизированных тестов на базе Selenium WebDriver за счет внедрения системы параллельного запуска / С. С. Кириллов // Перспективы науки. – 2022. – № 7 (154). – С. 34–39.
10. Артамонова, А. А. Разработка системы автоматизированного тестирования UEFI-драйверов с пользовательским интерфейсом / А. А. Артамонова, А. В. Куров // Вестник РГГУ. Серия: Информатика. Информационная безопасность. Математика. – 2020. – № 1. – С. 25–38.

*A.P. Sergushicheva*  
*Vologda State University*

### **BUILDING A TOOL FOR FUNCTIONAL PROGRAM TESTING**

Automated application testing tools speed up the process of commissioning software products and improve their quality. Many of the existing software testing applications are expensive and quite difficult for beginners. The article describes a relatively easy-to-learn system for functional testing of applications created at the Department of AVT, designed for use in educational purposes.

Functional testing, automated testing, software product, test case, parser.