

Мясоедова Виктория Алексеевна, студент, Хакасский государственный университет им. Николая Федоровича Катанова, Россия, г. Абакан

Голубничий Артем Александрович, научный руководитель, старший преподаватель кафедры ПОВТиАС, Хакасский государственный университет им. Николая Федоровича Катанова, Россия, г. Абакан

РАЗРАБОТКА ИНТЕРАКТИВНОГО ЛИНТЕРА ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ R

Аннотация: В статье дается краткий исторический экскурс по теме линтеров. Анализируются особенности и назначение линтеров. Рассматриваются инструменты для создания интерактивного линтера для языка программирования R. Представлен итоговый программный интерфейс разработанного программного обеспечения с использованием пакетов shiny и lintr.

Ключевые слова: статический анализ кода, линтер, язык R, пакет lintr, пакет shiny.

Abstract: The article provides a short historical excursion on the topic of linter. The features and purpose of linter are analyzed. The tools are considered for creating an interactive linter for the programming language R. The final software interface for the developed software using the shiny and lintr packages is presented.

Keywords: static code analysis, linter, R Language, package lintr, package shiny.

Для создания эффективной программы и для дальнейшего совершенствования программного продукта, необходимо грамотно и корректно

написать сам код для понимания работы с ним. Правильно написанный код необходим и полезен во многих случаях, приведем их краткий перечень:

- для новых сотрудников, которых включили в проект по созданию программного продукта;
- для легкого внесения правок в уже созданный код;
- для быстрого поиска ошибок.
- на корректный код достаточно просто посмотреть, чтобы понять основную логику его работы.

Статический анализ кода – это процесс поиска ошибок в исходном программном коде без запуска (компиляции или интерпретации) самой программы. Статический анализ кода лучше всего можно описать, как метод отладки путем автоматической проверки исходного кода перед запуском программы.

Чтобы программист не закливал внимание на правильности написания программного кода, был разработан инструмент линтер. Линтер не проверяет работоспособность кода на этапе компиляции или интерпретации. Линтер анализирует статичный код на ошибки [1].

Линтеры помогают приводить код к единому стилю и избегать ошибок. Дальнейшие правила линтинга, помогают улучшить код, так как линтер уже действует как предварительный обзорщик кода, проверяя все основные проблемы, такие как синтаксические ошибки, неправильные (некорректные) имена, пропуски табуляции и пробелов и т.д. Это повышает ценность проверки кода, поскольку, люди тогда более охотно проверяют реализацию программного продукта.

Линтер разработан как компилятор-аналитик и, по сути, часто фокусирует внимание на частях кода, которые могут привести к различному поведению в зависимости от конкретного используемого компилятора (в случае компилируемых языков программирования).

Механизм работы линтеров состоит в том, что они производят анализ статичного кода в соответствии с заданными шаблонами, эти шаблоны уже написаны в функциях линтера.

Для работы с линтером, в разных языках программирования необходимо подключить библиотеку, которая содержит функции линтера.

Первый линтер был написан для языка программирования C. Стивен Джонсон, ученый-компьютерщик из Bell Labs, придумал линтер в 1978 году, отлаживая грамматику, которую он писал для C, и занимаясь проблемами переносимости, связанными с переносом Unix на 32-битную машину. В дальнейшем эту тему развивал нидерландский ученый Франс Кунст, он выпустил книгу «Lint, программа проверки C» [2].

В данный момент у большинства языков программирования есть специальное программное обеспечение для статического анализа кода.

- Python – Pylint, MyPy, PyFlakes;
- JavaScript – JSLint;
- R – lintr.

Для создания интерактивного линтера для языка программирования R необходимо использовать основные пакеты, а именно lintr и shiny.

lintr – это пакет языка R, который проверяет исходный код на всевозможные ошибки [3].

Примеры некоторых функций этого пакета:

- `assignment_linter` – проверяет, что в качестве знака присваивания используется «<-» (фактически можно использовать и знак «=>», но что является не правильным по канонам языка);
- `camel_case_linter` – проверяет, чтобы стиль написания соответствовал camelCase синтаксису;
- `trailing_blank_lines_linter` – проверяет на наличие пустых завершающих строк.

Для придания интерактивности разрабатываемому линтеру необходимо использовать сторонние средства, так как в чистом виде язык

программирования R не предназначен для сложных визуализаций. Однако система пакетов, имеющаяся в официальной репозитории CRAN [4], помогает в решении задач визуализации, в том числе интерактивной. Для создания интерактивного линтера используется пакет shiny.

Shiny – это технология и пакет языка R, который позволяет создавать мощные интерактивные веб приложения, плюс этого пакета состоит в том, что нет необходимости владения JavaScript или HTML [5]. Shiny включает в себя средство для построения пользовательского интерфейса и серверной части.

UI – отвечает за расположение элементов управления и внешним видом приложения. User interface (UI) часто представляет собой нечто большее, чем просто фиксированный набор элементов управления, которые влияют на фиксированный набор выходов (результатов рендеринга данных). В зависимости от состояния другого входа может потребоваться, чтобы входы отображались или скрывались. Созданный интерфейс был сделан специально для удобства пользователя. При запуске приложения появляется окно (рисунок 1).

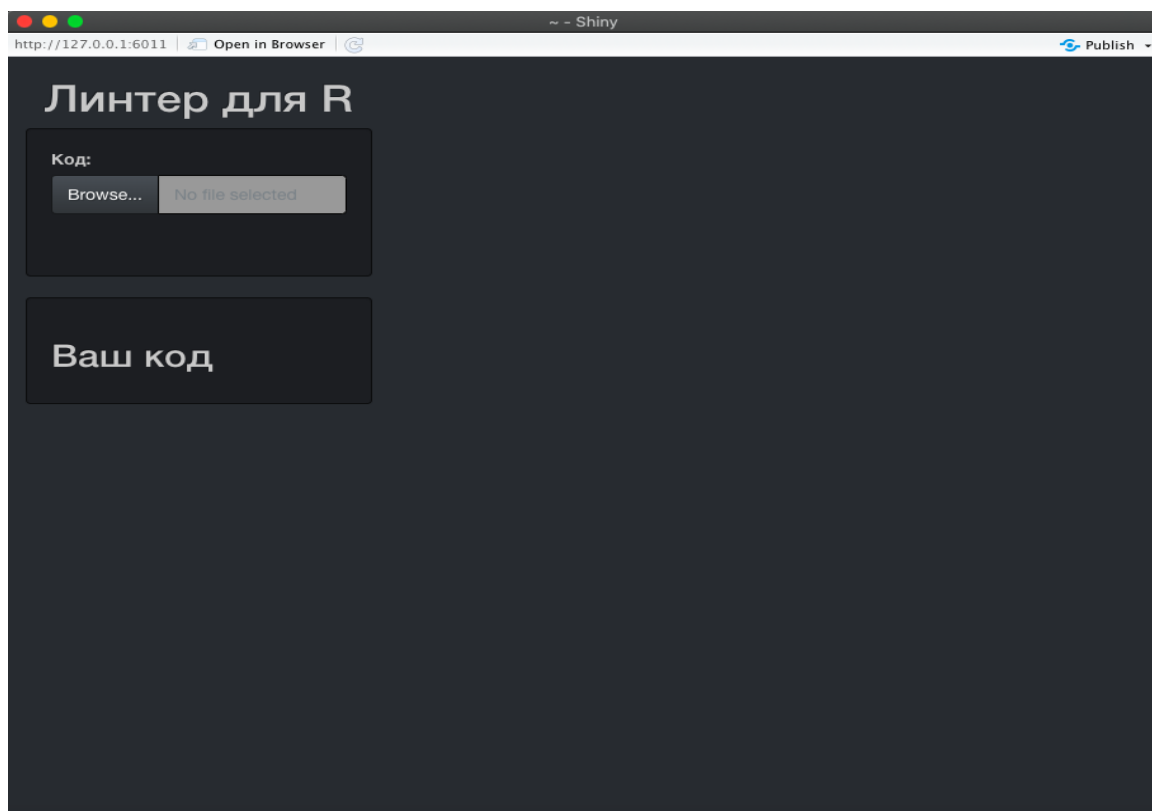


Рисунок 1 – Запуск приложения

Пользователю необходимо загрузить файл, который ему нужно проверить, это происходит с помощью кнопки-загрузки «Browse». Затем идет проверка кода на ошибки, проверку кода выполняет серверная часть приложения.

Скрипт `server.R` содержит инструкции, которые необходимы интерпретатору, чтобы создать приложение и управлять его поведением.

После проверки кода в серверной части, пользователю выводится соответствующие результаты (рисунок 2), эти результаты включают в себя:

- первое окно, содержащее исходный код, который содержался в загружаемом документе;
- второе окно, содержащее метаинформацию: сколько было строк в исходном файле, количество синтаксических ошибок, количество предупреждений, а также количество других видов ошибок;
- третье окно выводит непосредственно найденные ошибки, указывается строка, где была найдена ошибка, тип ошибки и название самой ошибки, название ошибок в сравнении с исходным пакетом `lintr` были переведены на русский язык.

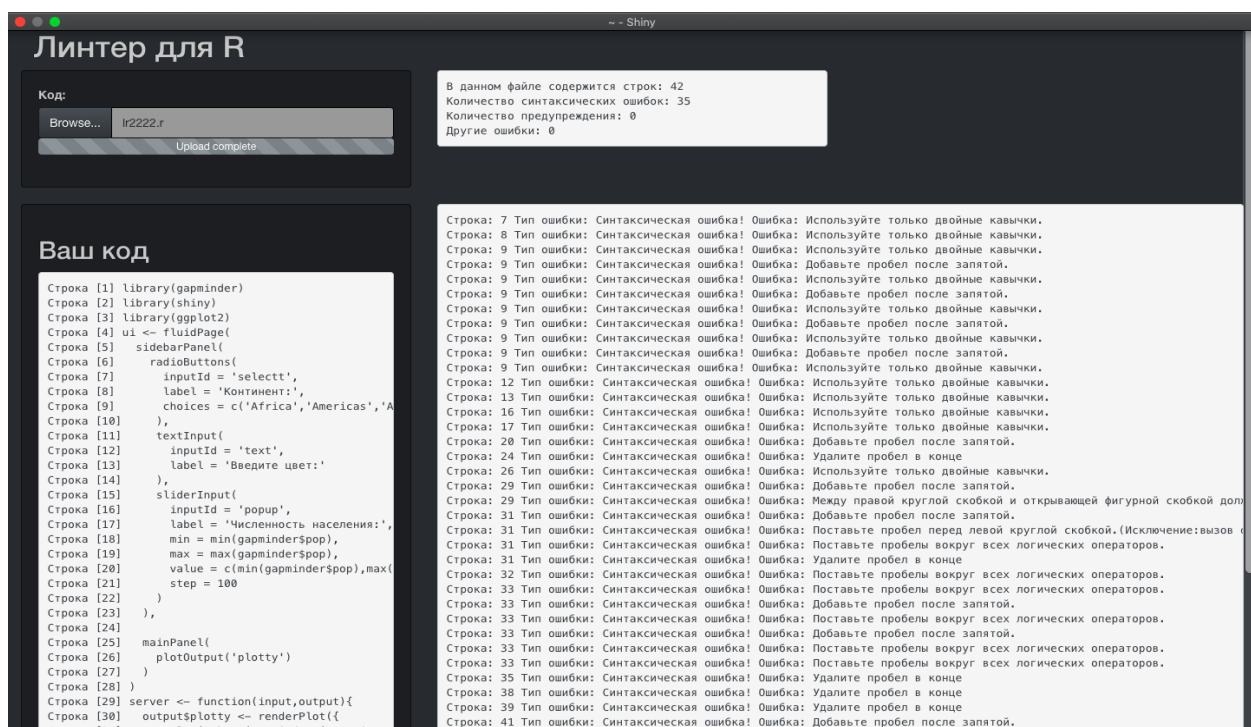


Рисунок 2 – Вывод результата проверки

Удобство созданного приложения состоит в том, что пользователю нет необходимости находить ошибки в консоли RStudio, также если пользователь не владеет в должной мере техническим английским языком, то перевод будет полезен, не нужно переводить ошибки на русский язык, так как иногда переводчик может не совсем правильно перевести ошибку, тем самым потерять смысл переведенного предупреждения или ошибки.

Для работы пользователя был разработан интерактивный линтер языка R, пользователь загружает файл и получает всю необходимую ему информацию для преобразования кода в более корректный вид.

Библиографический список:

1. What is “Linting”? – Stack Overflow [Электронный ресурс] URL: <https://stackoverflow.com/questions/8503559/what-is-linting> (дата обращения: 03.01.2020).
2. Johnson, S.C. Lint, a C program checker – Bell Laboratories, 1978 г. –16 с.
3. CRAN – Package linter [Электронный ресурс] URL: <https://CRAN.R-project.org/package=linter> (дата обращения: 03.01.2020).
4. The Comprehensive R Archive Network [Электронный ресурс] URL: <https://cran.r-project.org> (дата обращения: 03.01.2020).
5. CRAN – Package shiny [Электронный ресурс] URL: <https://CRAN.R-project.org/package=shiny> (дата обращения: 03.01.2020).