

СЦЕНАРИЙ

Глава Первая. Акт 1.

«Свежо предание»

Играем PS2 startup theme.

– (с энтузиазмом) Творческий коллектив «Союз юмористов» представляет. Обзор IDE Rider!

На экране медленно скроллится главная страница Rider.

– (с предвкушением) Rider был выпущен в 2017 году компанией JetBrains. В основе IDE лежит идея создания среды с широким набором рефакторинга кода и ускорения технологических процессов создания продукта. В основе Rider лежит другое решение компании - ReSharper, плагин для повышения продуктивности работы в Microsoft Visual Studio.

– (заигрываяще) Rider — это кроссплатформенная IDE, поддерживающая операционные системы Windows, macOS и Linux. Она обеспечивает расширенную поддержку редактирования и анализ кода для языков, используемых при разработке .NET.

На экране скроллится раздел «возможности» сайта

– (со знанием дела) Программа распространяется по проприетарной лицензии, широко используется в крупных IT-компаниях, например Adobe. Rider позволяет разрабатывать .NET, ASP.NET, .NET Core, Xamarin или Unity приложения. Может использоваться для разработки игр (Unity, Unreal Engine), работы в облаке (AWS, Azure), .NET и WEB нативных приложений. Так же поддерживаются технологии Entity Framework, Docker. Несмотря на большой набор функций, Rider — быстрая и отзывчивая IDE, которая делает упор на инспекцию и рефакторинг кода.

Глава Вторая. Акт 1.

«Тень хищника»

– () Чтобы испытать данное чудо бесплатно, нет необходимости иметь богатую родословную, ведь вы получите триал на месяц бесплатного использования. Бесплатность также гарантируется студентам и Opensource проектам, а как закончится – еще и 25% скидки подкинут! С другими оферами можно ознакомиться на сайте JetBrains. Счастливым обладателям Rider же предстоит установка и первоначальная настройка продукта.

– () Следите за руками: входим в аккаунт, изо всех сил стараемся доказать, свою человечность. На странице Licenses находим IDE Rider, скачиваем exe`шник. Так же предлагают скачать некий лаунчер JetBrains Toolbox, через который можно скачивать и обновлять весь доступный софт от компании, открывать недавние проекты. В общем, рассмотрим оба варианта и сравним, какой быстрее.

На экране расположены несколько вариантов установки среды, где-то маячит менеджер установки от JetBrains, где-то устанавливается exe`шник, где-то располагаются и радуют глаз гифки.

– (доверительно) Пока ждем победителя, давайте поговорим о требованиях. Когда-то программисты писали на бумаге и гордо называли этот процесс программированием. Разумеется, ни автодополнения кода, ни рефакторинга на перфокартах нет, все они – в Rider. Но даже райдеру нужна своя бумага...

– (наизусть) 2 GHz`ый процессор, 4 GB оперативной памяти и 2.5 GB места на жестком диске – Rider запустится даже на чайнике с такой конфигурацией, если на чайнике стоит Windows 8.1 и старше, конечно. IDE кроссплатформенная, так что не стоит ограничивать ваш чайник в выборе операционной системы, Rider поддерживает все: macOS 10.14+, Debian 9+, Ubuntu 16.04+. Стоит также понимать, что для корректной работы на вашем чайнике должен быть установлен .NET Framework.

Акт 2.

«Явление»

На экране демонстрируется окончание установки IDE.

– () Итак, ехе`шник победил по скорости, заняв почти ровно 5 минут. Без учета установки самого Toolbox, Rider устанавливался целых 13 минут. Теперь нас ждет настройка.

Демонстрация настройки шорткатов и привязки аккаунта к среде.

– () Если до Rider вы долго работали в других средах, то сможете импортировать настройки горячих клавиш, либо же применить существующие схемы известных IDE в Rider. Тут же можно настроить подсветку синтаксиса и установить интересные плагины.

– () В теории, теперь можно создать новый проект, открыть существующий и начать программировать. Но после первого запуска проекта наступает самый главный этап работы каждого программиста – настройка стиля IDE под себя. Продукты JetBrains славятся огромным количеством плагинов, выполняющих самые разные функции, одна из которых – оформление! Обращаю внимание на то, что это самый важный этап настройки, так что уделите этому необходимое время. Листайте, скачивайте, применяйте, перезагружайте – вуаля!

Нереальный монтаж настройки окружения, заканчивающийся ВАУ звуками и аниме девочками с мотоциклами.

Глава 3. Акт 1.

«Навигация – север»

Открыт любой проект.

- () Теперь давайте поговорим о функциях, за которые все так любят IDE от JetBrains.
- () Навигация в самом Rider может осуществляться графически, всегда есть возможность найти и нажать интересующую вас кнопку, скрывающуюся во множестве вкладок и окон среды. Рабочее пространство, разумеется, можно кастомизировать, дублировать и изменять.

Навигация с демонстрацией.

- () Навигация в проекте осуществляется с помощью окна Explorer. Дополнительно навигация может также осуществляться с помощью инструмента TODO – встроенной в среду системы управления задачами.
- () Ориентация в исходном коде может осуществляться как по символам (структурным элементам по типу класса), так и по имени или же по текстовому содержанию
- () Универсальный инструмент навигации Navigate To можно вызвать Shift+Shift или Ctrl+T. Все сочетания клавиш будут для предустановленной раскладки Visual Studio. Содержание диалогового окна Navigate To зависит от типа элемента, на который указывает каретка.
- () Необходимо найти и заменить конкретное слово или текст в проекте? Ctrl+F / Ctrl+N
- () Хотите сделать это только в одном файле или другой области видимости? Ctrl+Shift+F / Ctrl+Shift+N
- () Даже закомментированный код можно отыскать, применив полный поиск. Ctrl+F3 для того, чтобы найти далее выделенное слово, F3 – чтобы начать поиск из начала файла.
- () Навигация по символам полностью реализована с помощью Navigate To, но даже здесь есть некоторые полезные хоткеи. Переходим к объявлению – F12. Найти реализацию объявленного – Ctrl+F12. Я совершенно серьезен, эти штуки спасают нервы.

Акт 2.

«Автодополнение, автозавершение (англ. autocomplete) — функция в программах, предусматривающих интерактивный ввод текста (редакторы, оболочки командной строки, браузеры и т. д.), по дополнению текста по введённой его части. Источник википедия (дата обращения 8 марта 2014 года)»

Автодополнение и генерация кода с демонстрацией.

- () Rider поддерживает несколько видов автодополнения кода, генерацию типовых конструкций и дополнение отсутствующих идентификаторов.
- () Список предложенных дополнений появляется с началом ввода символов, остается только стрелочками выбирать подходящий вариант и нажимать на Tab или Enter. Допустим, у нас есть `public void`, который будет устанавливать позицию по X и Y. Еще до начала ввода можно нажать `Ctrl+Space`, вызвав контекстное меню с предложениями и совсем не думать.
- () Генерация базовых конструкций осуществляется с помощью `Alt+Insert`, нужно просто выбрать интересующую конструкцию из списка. При этом Rider может объявить используемый символ, если вы только что его придумали, или дополнить его объявления идентификаторами, все это доступно через быстрое исправление `Alt+Enter`.
- () Автодополнение поддерживает сокращения ключевых слов. Просто введите «ху..» – и вот наша функция, подобранная по заглавным буквам! Для расширения ассортимента выбора дополнений следует нажать `Ctrl+Alt+Space` или `Alt+Shift+Space`. Для принятия дополнения нажмите Enter и все!

Акт 3.

«Закодированный прораб»


– () Рефакторинг в Rider осуществляется универсальным инструментом Refactor This... сочетанием Ctrl+Shift+R. Все варианты переработки кода, которых, кстати, больше 50, доступны именно в этом окне и ранжируются в зависимости от символа или выделенного элемента, на котором стоит каретка.

– () Refactor This позволяет:

- переименовать любой символ, обновляя его по всему проекту; Ctrl+R, R
- перемещать сущности разных типов в типы, пространства имен, файлы и папки; Ctrl+R, O
- создавать новый метод или функцию, основанную на выделенном выражении; Ctrl+R, M
- создать новую локальную переменную или константу на основе выбранного выражения и так далее. Ctrl+R, V

Акт 4.

«Всевидающее око»

- () Rider начинает анализировать код в момент открытия редактора и продолжает это делать автоматически вплоть до закрытия IDE. Результат автоматического анализа проблем – проблемы разного уровня проблемности. Ошибки, предупреждения, предложения и подсказки.
- () IDE подсвечивает изъяны и бельма кода прямо в редакторе → . Но ориентироваться в них можно с помощью окошка «Проблемы» («Problems») сочетанием Alt+6.
- () Чтобы вас не мучали ошибки прошлого по ночам, проблемы можно отмечать в TODO списке, с чистой совестью доверяя их решения вам из будущего. Или просто исправлять их с помощью Alt+Enter.
- () Настроить фильтры анализа кода можно в настройках Rider Ctrl+Alt+S, Editor > Inspection Settings. Отключить проверку вашего безупречного кода или отдельных выдающихся файлов или потребовать больше подсказок? Вам туда...
- () К проблемам также относятся и несоблюдение требований Coding conventions или общепринятых паттернов Code Style, грамматические ошибки и опечатки. Если вы неграмотный, вам в Alt+Enter, для остального есть быстрое форматирование Ctrl+Alt+Enter.
- () Если вы думаете, что код безупречен и ошибок не может быть, а IDE сошел с ума и лоббирует свои интересы, в настройках можно отключить анализ кода и жить спокойно, не чувствуя на себе чуткого осуждающего взгляда.

Акт 5.

«Дезинсекция жуков»

- () Все знают, что такое отладка, говорить не о чем. Установить Breakpoints можно с помощью f9, а запустить отладку с помощью Alt+f5. Переход к следующей точке остановки в режиме отладки осуществляется с помощью f5.
- () В режиме отладки доступно управление ходом программы: используются горячие клавиши или интерфейс отладки. Пока программа работает, возможно отследить значение каждой переменной в точке остановки, а также проанализировать эффективность и затраты памяти программы с помощью встроенных плагинов MSTest, xUnit, bUnit и так далее, но в нашей кричащей тестовой программе таких заморочек нет.
- () Rider поддерживает Hot Reload, и исправления, внесенные в режиме отладки автоматически синхронизируются с этим режимом.

Акт 6.

«Скатерть самобранка»

– () Плагины для Rider не просто дополнительные функции, это основной строительный кирпичик IDE, так как сама среда была создана на основе плагина ReSharper для Microsoft Visual Studio. С помощью плагинов в IDE реализована подсветка синтаксиса, поддержка языков и фреймворков, кастомизация окружения и решение практических задач, таких, управление версиями и расширенная отладка.

Демонстрация прикольных плагинов.

Тема закрыта.

– () Если видеоролика оказалось для вас мало, можете проверить описание, где вы найдете ссылку на документ, составленный творческим коллективом «Союз Юмористов», а также ссылку на документацию от самих JetBrains, правда она на английском. В любом случае, всех благ.

ТИТРЫ!

В титрах отображается список участников: режиссёр, монтажер, оператор, звукорежиссер, продюсер, сценарист, актер озвучки. Всего 7 человек, 6 из которых – Лотуга Данила.