12 Лучших инструментов рефакторинга кода для ваших DevOps-проектов

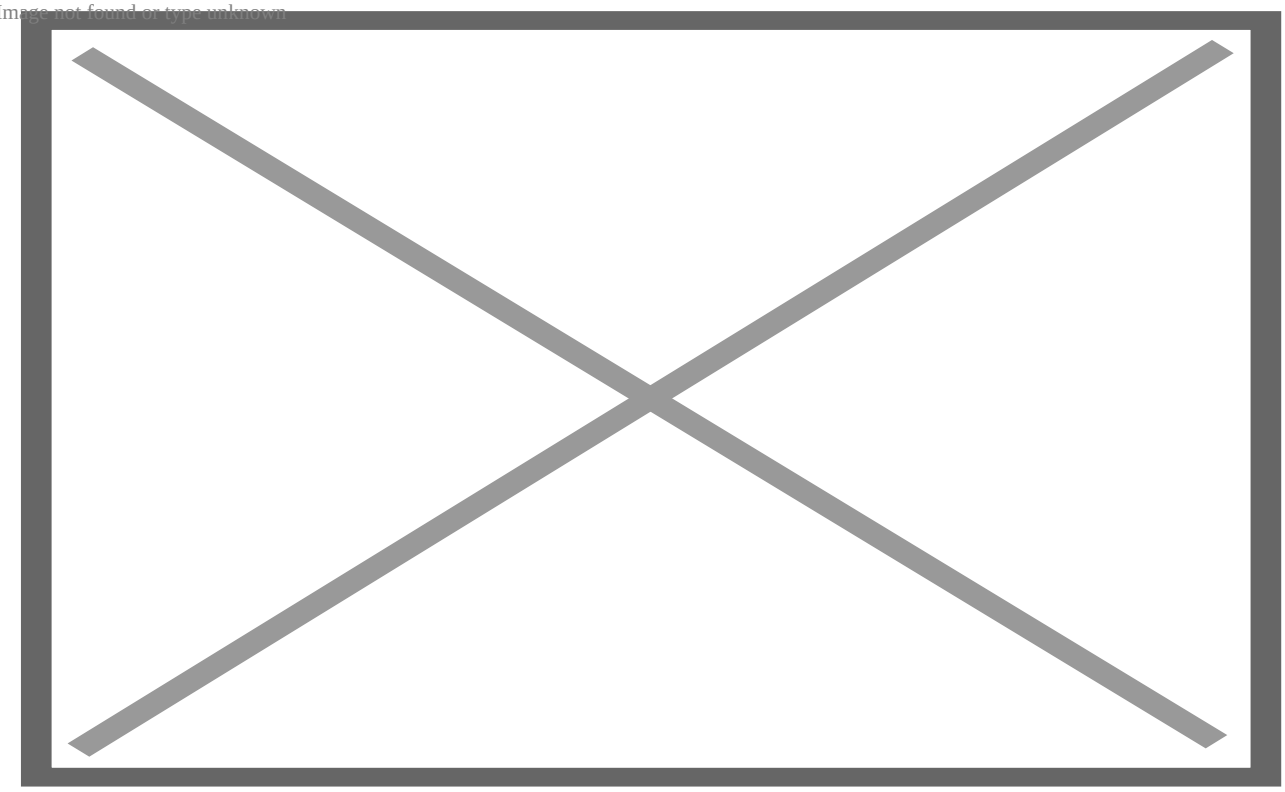
Описание

Вам необходимо отредактировать, очистить и реструктурировать код, чтобы сделать его более удобным для сопровождения и эффективным в рамках проекта по разработке программного обеспечения DevOps? Упомянутые в этой статье инструменты рефакторинга кода помогут вам! Agile и DevOps – наиболее успешные фреймворки для разработки программного обеспечения в условиях, когда быстрое развертывание высококачественного программного обеспечения имеет решающее значение для успеха вашего бизнеса. Согласно отчету Markets and Markets, текущий доход рынка DevOps составляет 10,4 млрд. долл. и должен вырасти до 25,5 млрд. долл. к 2028 году.

Это означает наводнение рынка DevOps настольными и мобильными приложениями от некачественных до премиальных разработчиков инструментов DevOps. На этом разросшемся рынке DevOps поиск лучших инструментов и приложений для запуска проекта разработки программного обеспечения должен быть непростой задачей. Эта статья поможет вам понять, что такое рефакторинг кода в DevOps, как получить правильные инструменты для этой цели, а также некоторые из лучших инструментов для рефакторинга кода, представленных на рынке.

Что такое рефакторинг кода в DevOps?

Image not found or type unknown



Рефакторинг кода – это процесс улучшения качества программного кода путем изменения некоторых его частей, дедупликации кодовой базы, устранения ненужных зависимостей и т.д. В DevOps рефакторинг кода выполняется сразу после цикла разработки, управляемой тестами (TDD), чтобы сделать код сопровождаемым и компактным без изменения внешнего поведения программы. Если вы следуете принципам разработки, ориентированной на поведение (BDD), или разработки, ориентированной на приемочное тестирование (ATDD), вам необходимо проводить рефакторинг кода.

Рефакторинг программного кода – неотъемлемая часть Agile- и DevOps-разработки программного обеспечения. Он позволяет разработчикам, ориентированным на бизнес, погасить технический долг перед выпуском продукта, чтобы избежать серьезных сбоев в работе программного обеспечения, когда оно станет достоянием общественности. Мартин Фаулер является родоначальником концепции рефакторинга кода. Он подробно объяснил рефакторинг кода для бизнеса и разработчиков программного обеспечения в своей книге “Рефакторинг: Улучшение дизайна существующего кода”. Если вы занимаетесь разработкой программного обеспечения, вам обязательно нужно прочитать эту книгу.

Когда следует рассматривать возможность рефакторинга кода

Большинство DevOps-проектов включает в себя график рефакторинга кода при добавлении новых функций или обновлении программного обеспечения. Также рефакторинг кода можно проводить при ежемесячном, полугодовом, ежегодном и т.д. пересмотре программного кода. Стоит отметить, что это последний шанс исправить и оптимизировать код перед запуском программы или сервиса в эксплуатацию. Существуют Agile-проекты разработки программного обеспечения, в которых также реализуется график частого рефакторинга кода. Ниже приведены некоторые советы о том, когда следует проводить рефакторинг программного кода:

- Наблюдаются логические повторы или кольцевые структуры кода.
- Несколько разработчиков сталкиваются с трудностями в понимании кода и его функциональности.
- Возникают проблемы с определенным участком кода.
- Процесс отладки занимает больше времени, чем ожидалось.
- Случайная отладка происходит из-за отсутствия комплексного метода решения проблем.
- Последний рефакторинг кода был проведен достаточно давно, и сейчас его необходимо обновить.
- Планируется добавить важную функцию, компонент, крупный блок или интегрировать со сторонним решением.

Лучшие практики рефакторинга кода

Ниже приведены удобные рекомендации по практике рефакторинга кода:

- Регулярно проводите рефакторинг кода для поддержания его качества, а также для сокращения технического долга.
- Сведите к минимуму риск добавления ненужных ошибок, рефакторя кодовую базу небольшими фрагментами.
- После рефакторинга не забывайте проверять функциональность кода на соответствие требуемому внешнему поведению.
- В проекте рефакторинга придерживайтесь подхода “съесть лягушку”. Это означает, что приоритет отдается тем областям, которые затрагивают

несколько частей кодовой базы или являются сложными для понимания.
- Используйте программу контроля версий или веб-приложение для поддержания различных версий рефакторингового кода и при необходимости возвращайтесь к наиболее известной версии.
- В проекте рефакторинга кода должны участвовать все сотрудники команды DevOps.
- Создайте документ или журнал рефакторинга кода и записывайте в него причины и подход к каждой сессии рефакторинга для дальнейшего использования.
- Рекомендуется проводить рефакторинг кода при проверке программного обеспечения, мобильного или веб-приложения в целях аудита.

Для ускорения процесса и поддержания согласованности можно использовать автоматизированные средства рефакторинга. Валидация рефакторингового кода с помощью комплексного тестирования и анализа производительности.

Преимущества рефакторинга кода

Без рефакторинга кода вы бы продолжали добавлять функциональные возможности в существующую кодовую базу программного обеспечения. Когда код становится очень сложным и не поддерживаемым, вы избавляетесь от него и начинаете работу с нуля. Однако, приступая к рефакторингу программного кода, вы делаете его эффективным для реализации текущих бизнес-ценностей и сохраняете его совместимость с будущими бизнес-ценностями без разработки с нуля. Ниже приведены некоторые общие преимущества рефакторинга кода в DevOps и Agile:

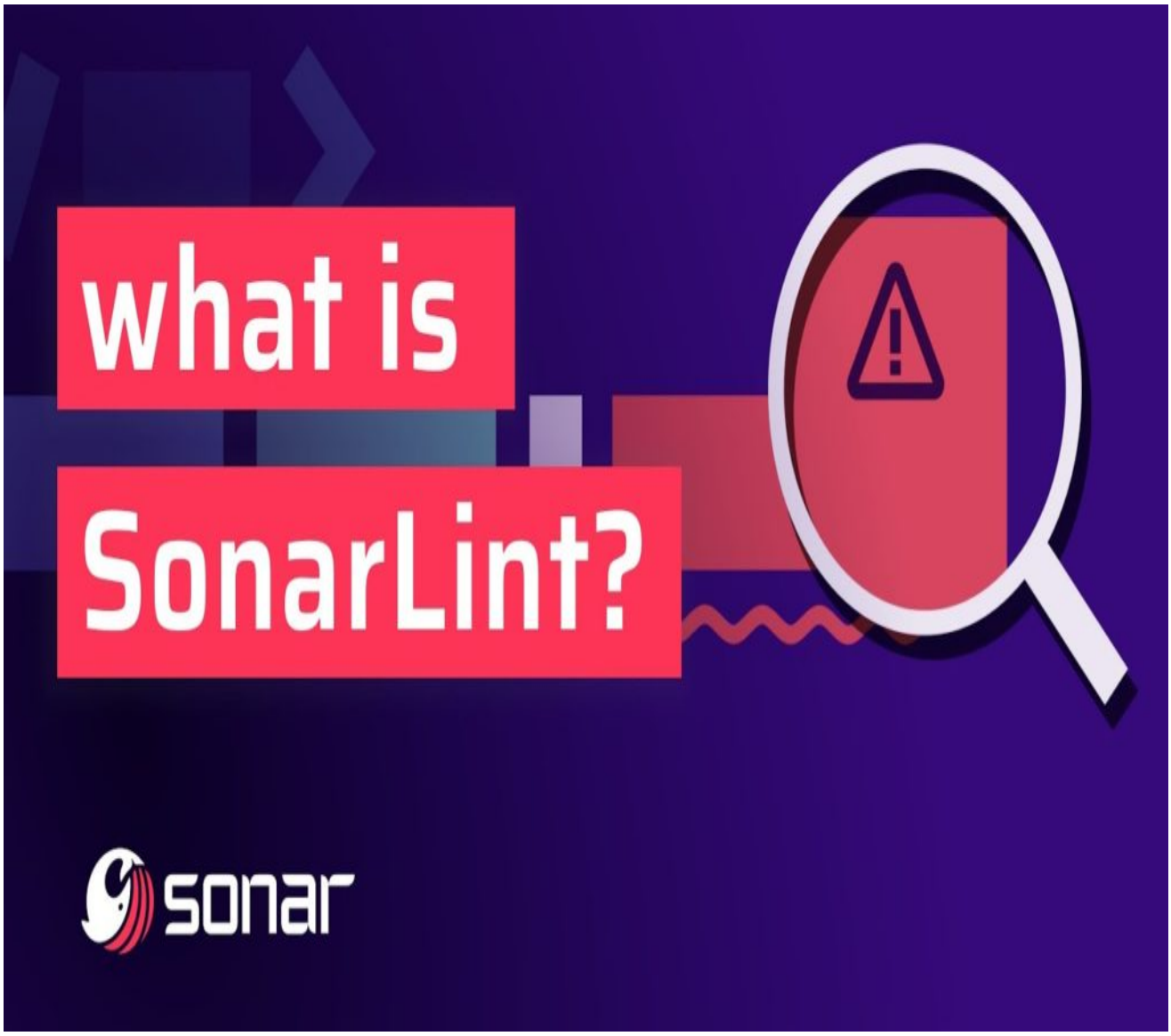
- Повышение качества и читабельности кода, что облегчает его сопровождение и устранение неполадок
- Сокращение технического долга и минимизация риска возникновения ошибок в будущем
- Повышение общей эффективности и производительности конечного продукта или сервиса
- Обеспечивает учет всех отзывов и комментариев всей команды DevOps посредством совместной работы
- Сохраняет модульность кодовой базы программного обеспечения, что позволяет в будущем легко добавлять и удалять функции.

- Процесс рефакторинга кода создает стандартную практику разработки
- Ваше программное обеспечение или мобильное приложение становится масштабируемым
- Это способствует развитию культуры непрерывного совершенствования в DevOps
- Новые разработчики могут легко разобраться в кодовой базе, когда существующие разработчики выходят из проекта.

Итак, ниже мы рассмотрим инструменты рефакторинга кода, которые используют наиболее успешные DevOps-проекты:

12 Лучших инструментов рефакторинга кода для ваших DevOps-проектов

SonarLint



SonarLint – популярный инструмент для рефакторинга кода, легко интегрируемый со многими интегрированными средами разработки (IDE). Он позволяет разработчикам выявлять и устранять проблемы с качеством кода в режиме реального времени. Анализируя код “на лету”, SonarLint обнаруживает ошибки, уязвимости в системе безопасности, запахи кода и проблемы с сопровождением. Такие отчеты о качестве кода помогают разработчикам незамедлительно вносить улучшения. Благодаря 5 000+ правилам кодирования и потоку данных о проблемах SonarLint обеспечивает стабильное качество кода во всех проектах.

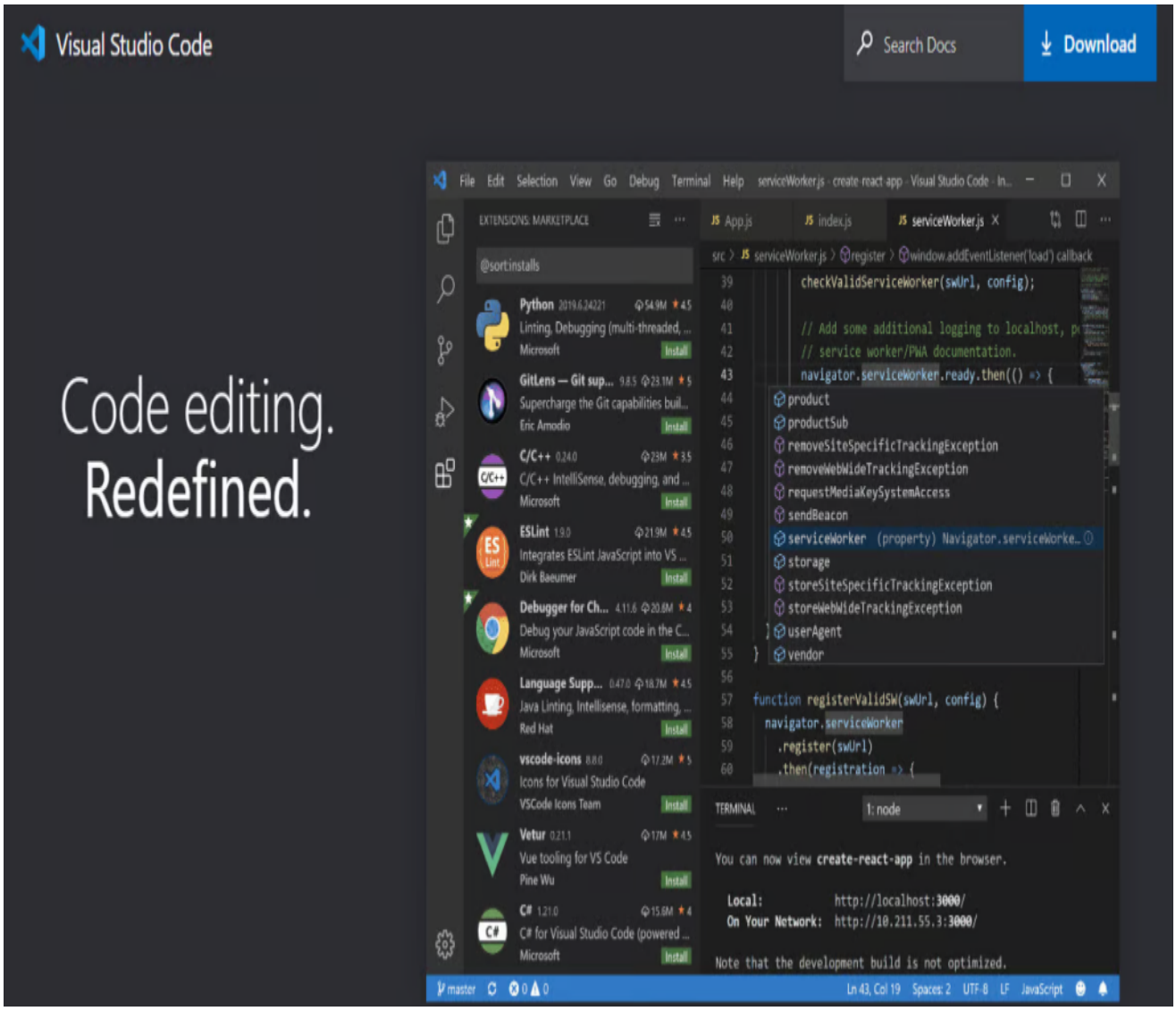
IntelliJ IDEA

IntelliJ IDEA предлагает широкие возможности рефакторинга кода для повышения его качества и удобства сопровождения. Интеллектуальный анализ кода позволяет мгновенно выявлять потенциальные проблемы и применять различные методы рефакторинга. Также в IntelliJ IDEA реализована автоматическая рефакторинговая обработка для таких задач, как переименование переменных, извлечение методов и введение переменных.

Кроме того, можно выполнять расширенные рефакторинги, такие как извлечение интерфейсов и перемещение членов в разные классы. Сохранение целостности и функциональности кода является реальной задачей при рефакторинге кода. IntelliJ IDEA предлагает средства для решения этой проблемы с помощью таких функций, как предварительный просмотр изменений при рефакторинге и разрешение

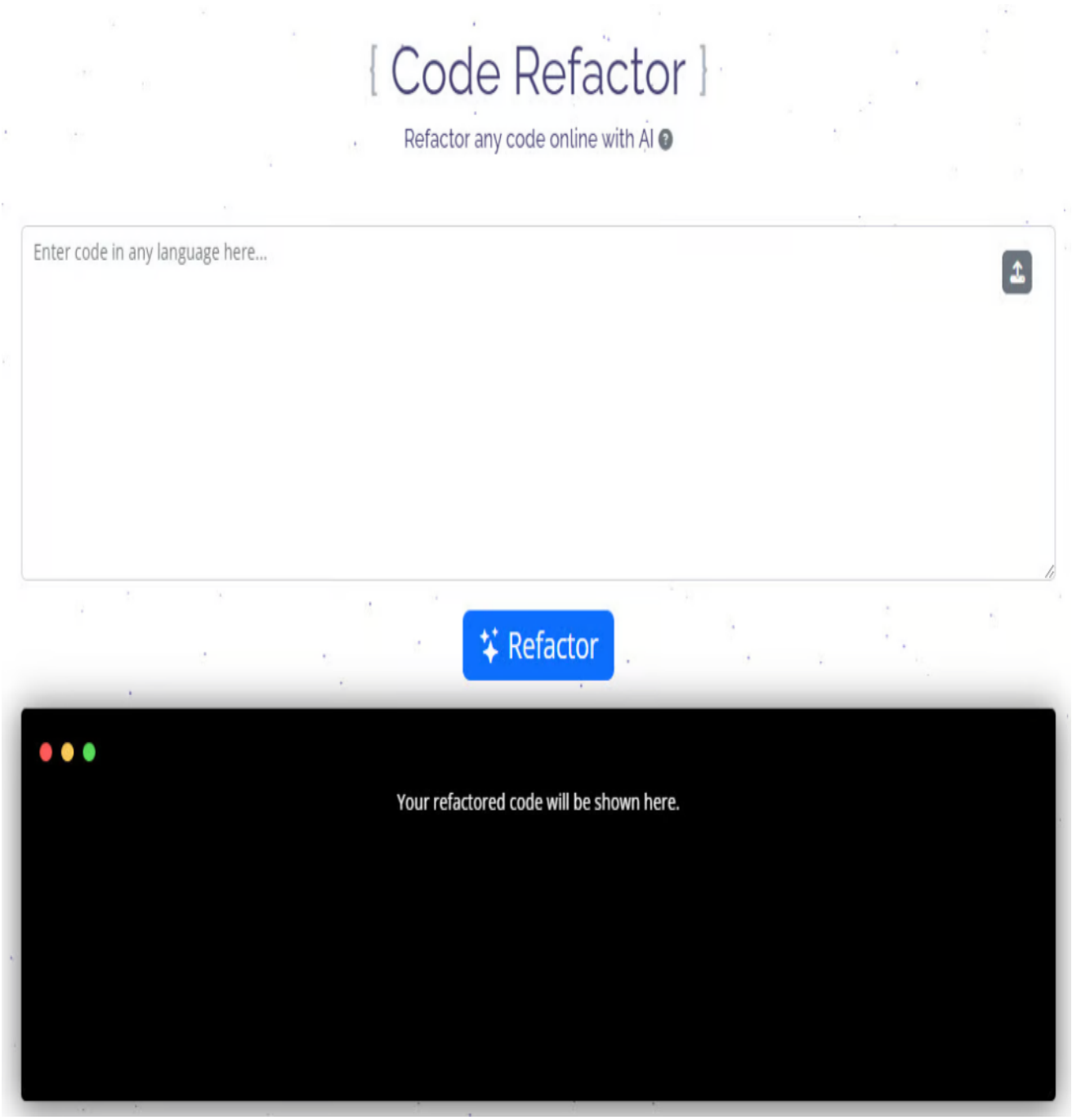
конфликтов.

Visual Studio Code



Доля Visual Studio Code компании Microsoft на рынке IDE составляет 41,16%. Это основная IDE, на которой разрабатываются кодовые базы большинства DevOps-проектов. Ее возможности по рефакторингу кода также не имеют себе равных. В ней имеется множество встроенных действий по рефакторингу, а также поддержка сторонних интеграций рефакторинга кода из рынка VS Code. Среди встроенных действий рефакторинга можно назвать извлечение метода, извлечение переменной, переименование символа и т.д.

CodePal



CodePal – это инструмент для рефакторинга кода, основанный на искусственном интеллекте и работающий в режиме DevOps. Он имеет две разновидности, описанные ниже:

- Веб-интерфейс для рефакторинга кода, в котором можно отправлять запросы на рефакторинг.
- API-сервис CodePal позволяет интегрировать систему рефакторинга кода в IDE или веб-сайты.

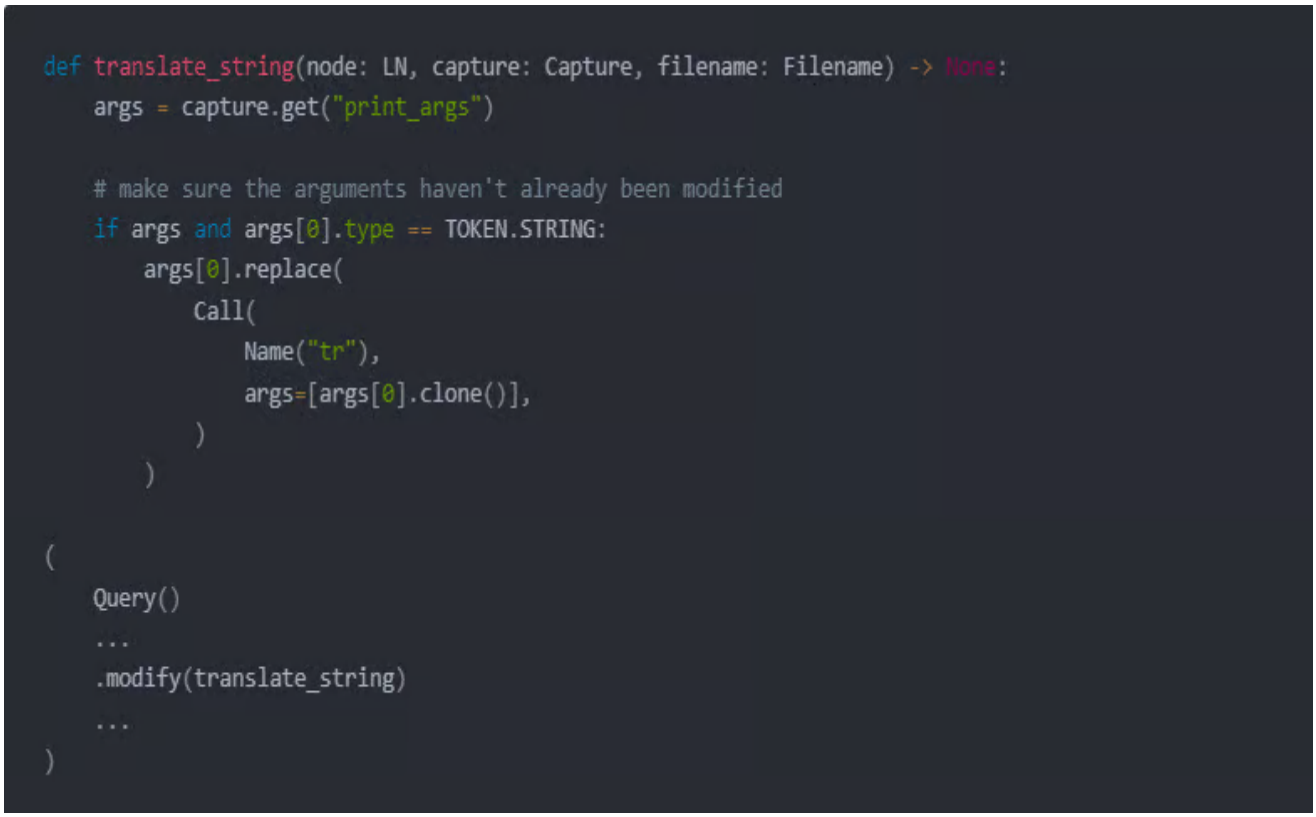
Вы можете воспользоваться бесплатным планом или оформить платную подписку. Бесплатный план позволяет делать меньшее количество запросов небольшого размера. Платные планы позволяют выполнять 250+ запросов к большим кодовым базам.

CodeRush

С помощью CodeRush вы получаете возможность улучшать читаемость кода,

изменять его и модифицировать внутреннюю структуру без изменения внешнего поведения. При работе над структурами кодирования, требующими наличия только одного типа для каждого файла, пригодится функция CodeRush по рефакторингу организации файлов. Используя директивы, инструмент может оптимизировать и сортировать код. Кроме того, он удаляет из кода ненужные и неиспользуемые элементы, делая его понятным и читабельным. CodeRush способен безопасно оптимизировать устаревший код для максимального использования новых возможностей языка.

Bowler

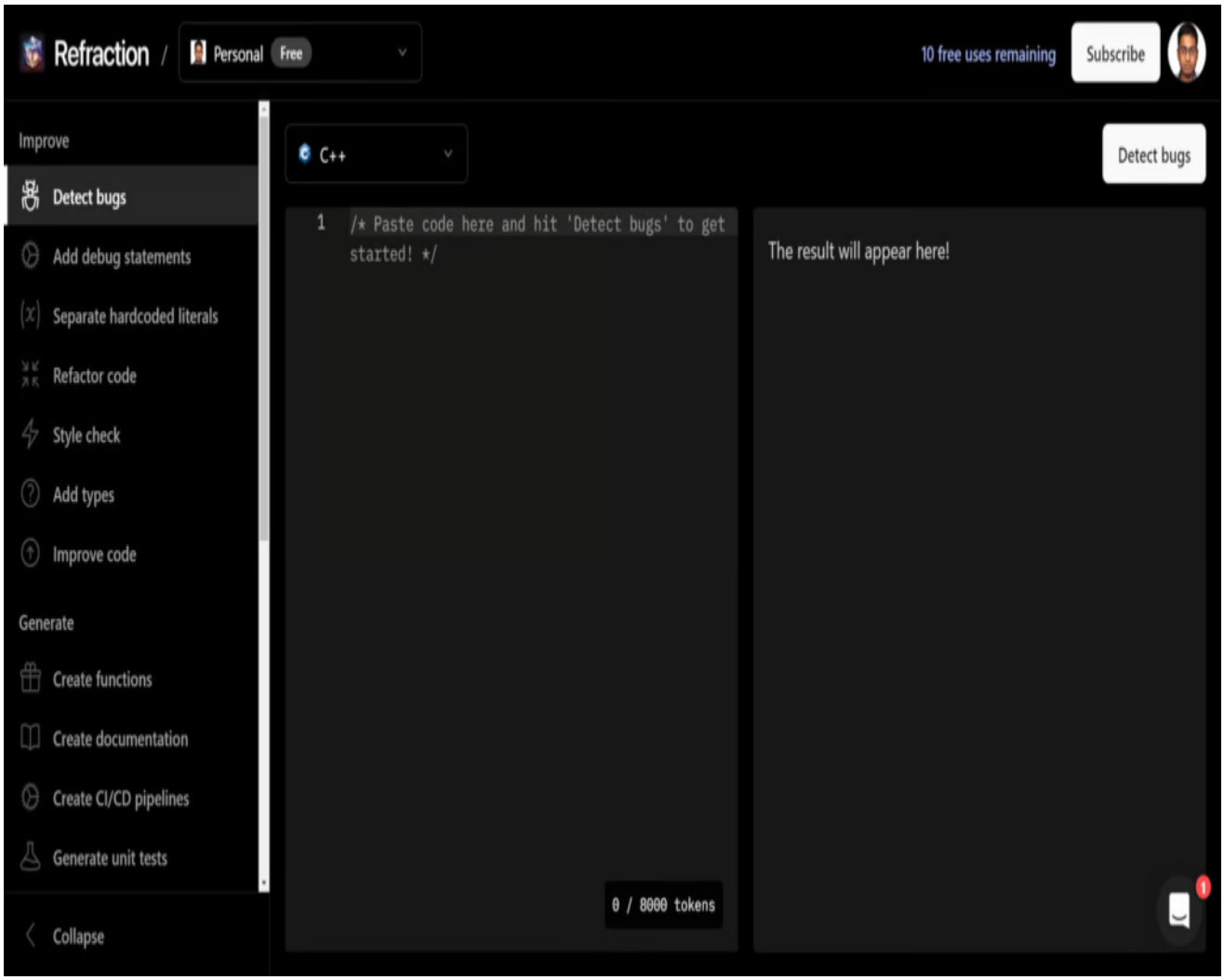


```
def translate_string(node: LN, capture: Capture, filename: Filename) -> None:
    args = capture.get("print_args")

    # make sure the arguments haven't already been modified
    if args and args[0].type == TOKEN.STRING:
        args[0].replace(
            Call(
                Name("tr"),
                args=[args[0].clone()],
            )
        )

(
    Query()
    ...
    .modify(translate_string)
    ...
)
```

Если вам нужен безопасный инструмент для рефакторинга современного Python-кода, то Bowler должен стать вашим лучшим выбором. Этот инструмент может использоваться разработчиками как для автоматизированных изменений, так и в качестве еще одного библиотечного компонента для редактора кода. Благодаря возможности создания композитных, многократно используемых и простых скриптов рефакторинга, он обеспечивает постоянную полезность и не требует выбрасывать скрипты после каждого использования. Этот инструмент построен на основе стандартных библиотек. Благодаря этому он поддерживает не только последние версии Python, но и обратно совместим со всеми предыдущими версиями

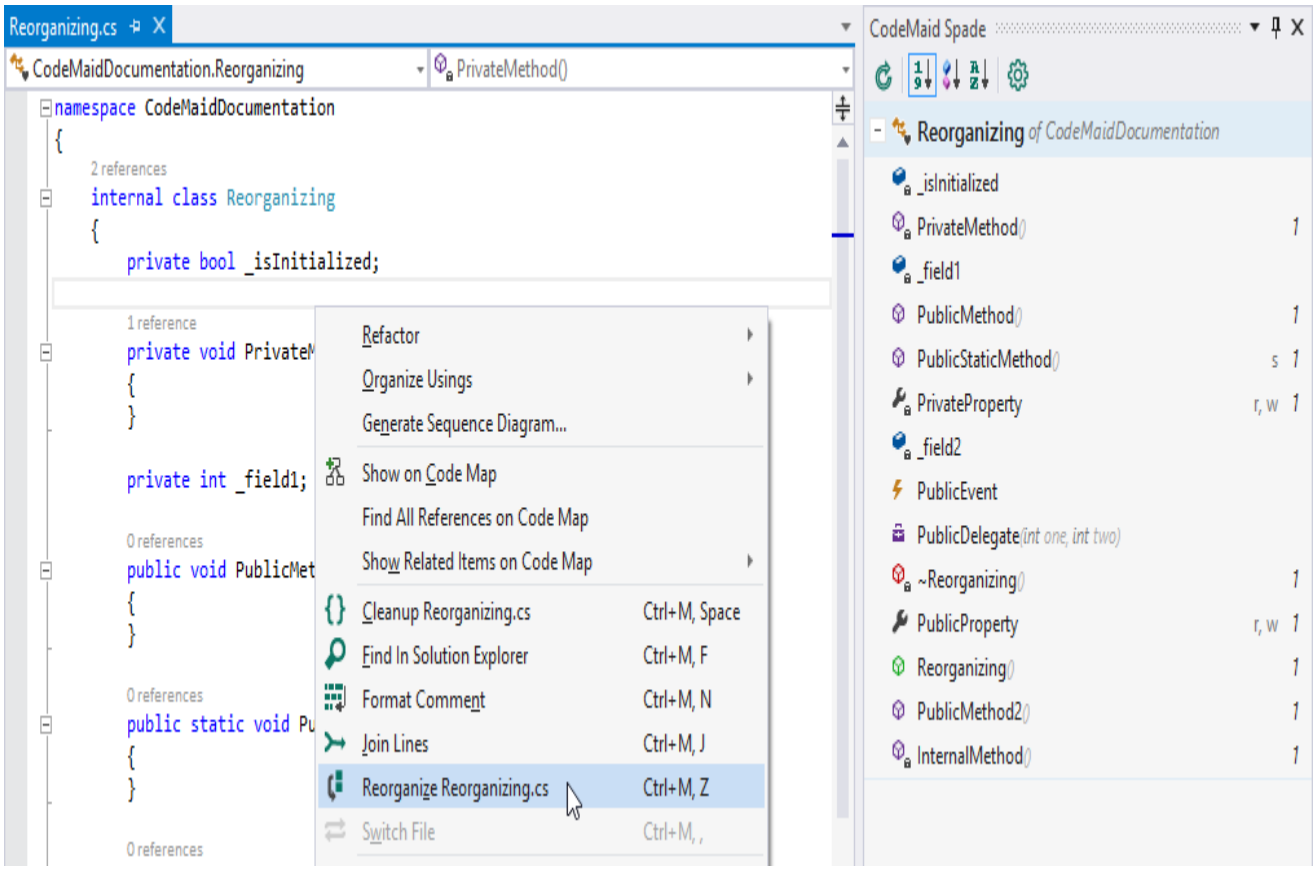
Python.

Refraction



Refraction выводит кодирование на основе ИИ на новый уровень, позволяя использовать ИИ для рефакторинга и документирования кода. Для этого достаточно зарегистрировать бесплатную учетную запись, которая позволяет использовать до 10 пользователей. Выберите язык программирования кодовой базы из обширного списка и ждите волшебства ИИ. На момент написания статьи инструмент поддерживал более 50 языков программирования, таких как Python, Scala, SAP ABAP, C++, GraphQL, Kotlin и другие. Кроме того, он позволяет выполнять рефакторинг кода в терминале Mac с помощью расширения The Terminal из Refraction. Имеются и другие расширения для VS Code, GitHub Autoreview и т.д.

CodeMaid

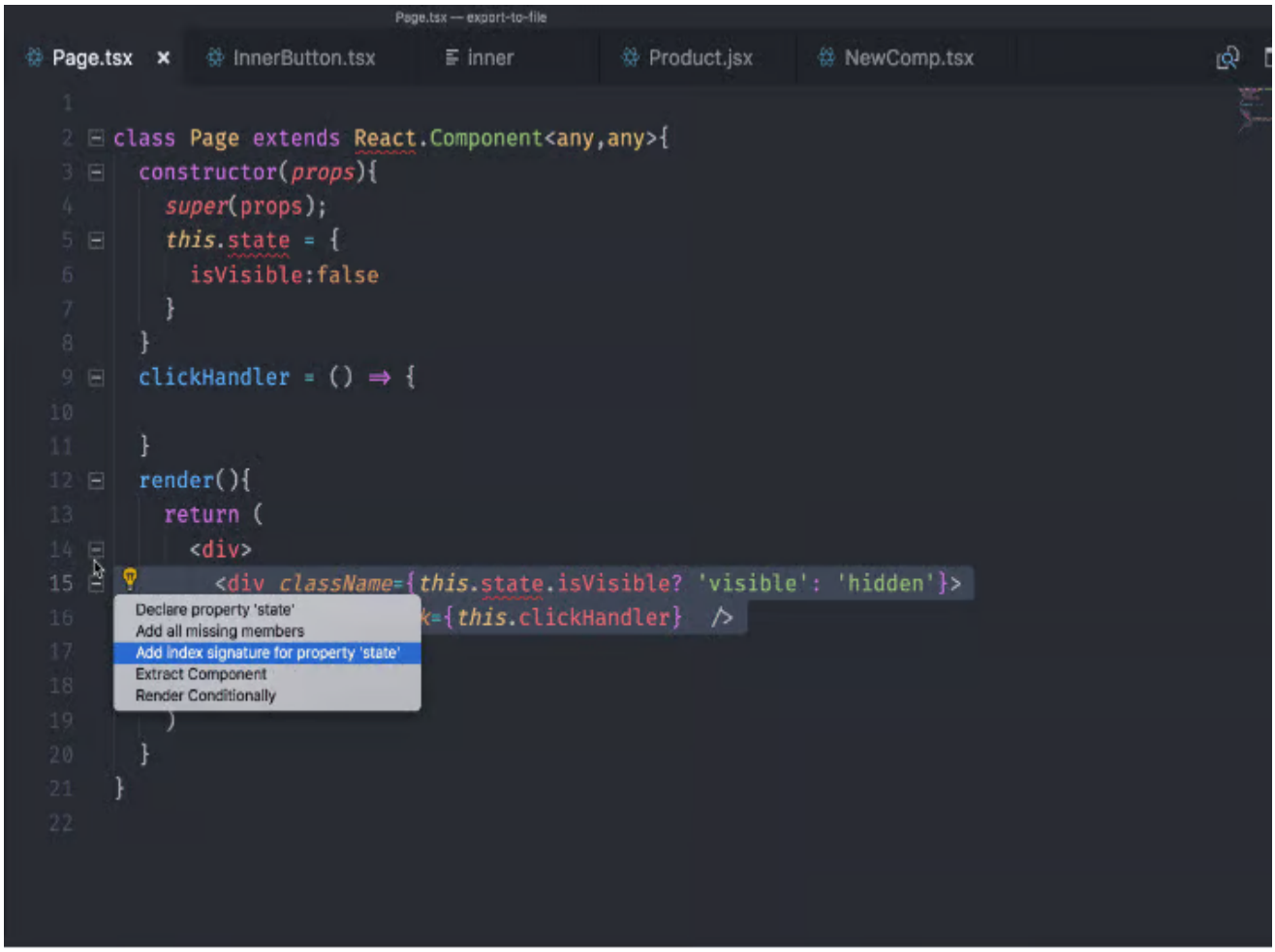


CodeMaid – это расширение Visual Studio с открытым исходным кодом, позволяющее рефакторить код, написанный на таких языках, как C#, C++, F#, JSON, JavaScript, TypeScript, XML, HTML, PHP, PowerShell, VB, R и многих других. С помощью этого инструмента разработчики могут удалять из кода случайные пробелы. Кроме того, с его помощью можно добавлять неопределенные модификаторы доступа, устранять и сортировать операторы использования, а также максимально использовать встроенные средства форматирования Visual Studio. Более того, все это может быть сделано автоматически или по запросу, причем как из одного файла, так и из всего кода. Кроме того, с его помощью можно решать такие задачи, как копание в коде, распознавание кода и форматирование кода.

ReSharper

ReSharper поставляется с набором рефакторингов, которые анализируют выбранный код для получения информации и затем обновляют существующий код на основе полученной информации с помощью своего интеллекта. Все функции рефакторинга можно использовать в коде на C#, однако некоторые из них могут применяться в таких языках, как VB.NET, ASP.NET, XAML, JavaScript, TypeScript и др. К атрибутам рефакторинга относятся извлечение суперкласса, введение параметра, изменение сигнатуры, преобразование интерфейса в абстрактный класс и наоборот, преобразование метода расширения в простой статический и наоборот, перемещение строки в ресурс, переименование, перемещение членов вверх или вниз и т.д.

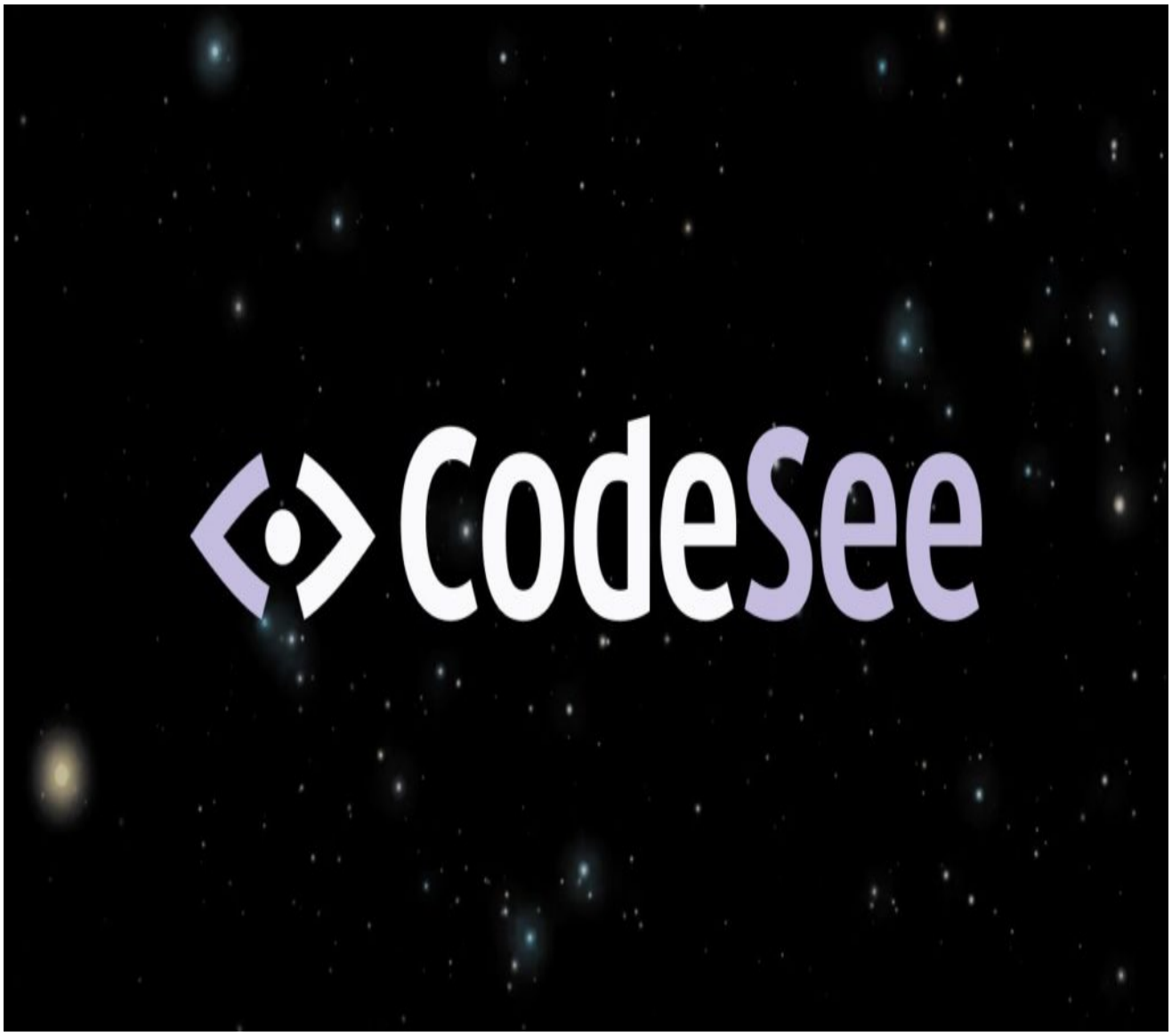
glean



glean позволяет запускать полнофункциональные проекты рефакторинга кода в Visual Studio Code для всех ваших проектов, использующих React для создания кодовых баз. Это надежное и проверенное расширение для Visual Studio Code, разработанное компанией Wix. Среди его возможностей по рефакторингу кода для кодовой базы React можно выделить следующие:

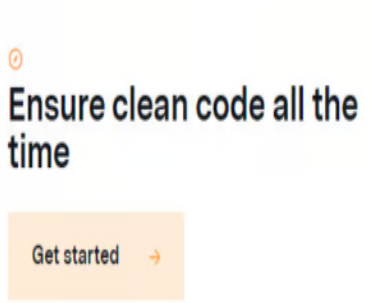
- Извлечение JSX (JavaScript Syntax Extension) в новый компонент React.
- Возможность преобразования компонентов класса в функциональные компоненты
- Переименование переменной состояния
- Обертывание с помощью хуков, таких как useCallback, useEffect, useMemo и т.д.

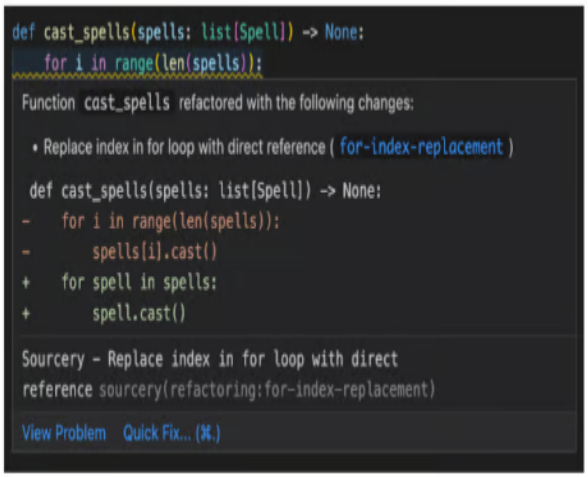
CodeSee



С помощью CodeSee можно визуализировать унаследованный код и понять его для целей рефакторинга. С его помощью можно автоматически создать точную визуальную модель для рефакторинга. CodeSee также может автоматически синхронизировать карты кодовой базы и автоматическое обнаружение сервисов для визуализации прогресса. Инженеры также могут использовать этот инструмент для выработки новых привычек. Например, автоматические комментарии могут служить напоминанием о необходимости внедрения файлов в микросервис, а не в конкретную папку. Кроме того, он может отображать и автоматизировать сервисы, изменения кода, каталоги и файловые зависимости вашего приложения, чтобы часто поставлять стабильный код.

Sourcery





Sourcery – это инструмент, который помогает постоянно улучшать код, просматривая его со всех устройств. Реализовав автоматические предложения, можно легко получить чистый и качественный код. Он также позволяет определить правило и включить его в CLI для исправления каждого случая конкретной проблемы. Поскольку с помощью этого инструмента можно выявлять проблемы в режиме реального времени, нет необходимости ждать этапа проверки кода. Sourcery уважает вашу конфиденциальность и использует шифрование AES256. Более того, код никогда не покидает ваших устройств, а значит, вы можете быть уверены в его безопасности.

Заключительные слова

Правильный выбор инструментов на начальном этапе – это ключ к успеху в DevOps. Вы не можете позволить себе терять время и бюджет на замену неэффективного инструмента в середине проекта. Поэтому выбирайте инструменты DevOps, такие как программы для рефакторинга кода, с умом и не теряйте продуктивности. Приведенный выше список должен помочь в этом.

Дата Создания
26.07.2023