

# Руководство пользователя Kate

Anders Lund

Seth Rothberg

Dominik Haumann

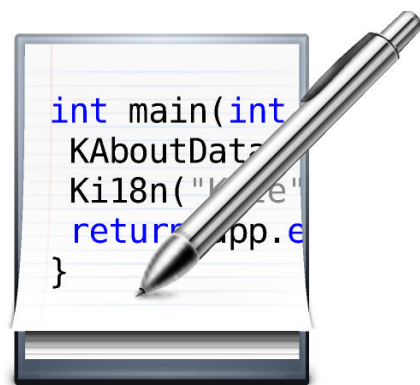
T.C. Hollingsworth

Перевод на русский: Андрей Балагута

Редактирование перевода: Николай Шафоростов

Перевод на русский: Алексей Опарин

Обновление перевода: Мария Шикунова



## Руководство пользователя Kate

# Оглавление

1	Введение	10
2	Основы	11
2.1	Запуск Kate	11
2.1.1	Из меню запуска	11
2.1.2	Из командной строки	11
2.1.2.1	Параметры командной строки	12
2.1.3	Перетаскивание	13
2.2	Работа в Kate	13
2.2.1	Начало работы	13
2.2.2	Комбинации клавиш	13
2.3	Работа с многодокументным интерфейсом Kate	14
2.3.1	Обзор	14
2.3.1.1	Главное окно	14
2.3.2	Область редактирования	14
2.4	Работа с сеансами	15
2.5	Быстрый переход	15
2.5.1	Работа с быстрым переходом	16
2.5.2	Настройка быстрого перехода	16
2.6	Получение справки	16
2.6.1	Получение справки в Kate	16
2.6.2	Получение справки из внешних файлов	17
2.6.3	Статьи про Kate	17
3	Работа с редактором Kate	18
4	Работа с модулями	19
4.1	Модули Kate	19
4.2	Внешние программы	20
4.2.1	Настройка внешних программ	21
4.2.2	Расширение переменных	22
4.2.3	Список стандартных программ	24
4.3	Модуль просмотра стека вызовов	27
4.3.1	Работа с модулем «Просмотр стека вызовов»	27

## Руководство пользователя Kate

4.3.2	Настройка . . . . .	27
4.4	Модуль для сборки ПО . . . . .	28
4.4.1	Введение . . . . .	28
4.4.2	Работа с модулем для сборки ПО . . . . .	28
4.4.2.1	Вкладка «Параметры цели» . . . . .	29
4.4.2.2	Вкладка «Вывод» . . . . .	30
4.4.3	Структура меню . . . . .	30
4.4.4	Благодарности . . . . .	31
4.5	Модуль «Закрыть кроме/только...» . . . . .	31
4.5.1	Введение . . . . .	31
4.5.2	Работа с модулем «Закрыть кроме/только...» . . . . .	31
4.5.3	Структура меню . . . . .	31
4.6	Модуль выбора цвета . . . . .	32
4.6.1	Введение . . . . .	32
4.6.2	Настройка . . . . .	32
4.7	Цветные скобки . . . . .	32
4.7.1	Введение . . . . .	32
4.7.2	Настройка . . . . .	32
4.8	Модуль CTags . . . . .	33
4.8.1	Введение . . . . .	33
4.8.2	Настройка . . . . .	33
4.8.2.1	Общее индексирование . . . . .	33
4.8.2.2	Индексирование текущего сеанса . . . . .	33
4.8.3	Работа с модулем CTags . . . . .	34
4.8.4	Структура меню . . . . .	34
4.9	Модуль предварительного просмотра документа . . . . .	35
4.9.1	Введение . . . . .	35
4.9.2	Структура меню . . . . .	35
4.9.3	Интерфейс . . . . .	35
4.10	Модуль переключателя документов . . . . .	35
4.10.1	Структура меню . . . . .	35
4.11	Модуль обзора файловой системы . . . . .	36
4.11.1	Структура меню . . . . .	36
4.11.2	Интерфейс . . . . .	36
4.11.3	Настройка . . . . .	38
4.12	Модуль списка документов . . . . .	38
4.12.1	Введение . . . . .	38
4.12.2	Структура меню . . . . .	39
4.12.3	Настройка . . . . .	39
4.13	Модуль GDB . . . . .	40
4.13.1	Введение . . . . .	40
4.13.2	Структура меню и панели инструментов . . . . .	40

## Руководство пользователя Kate

4.13.3	Панель «Отладка» . . . . .	41
4.13.4	Локальные переменные и стек . . . . .	43
4.13.5	Благодарности . . . . .	43
4.13.6	Настройка . . . . .	43
4.13.6.1	Настройка среды выполнения . . . . .	45
4.14	Модуль проектов . . . . .	46
4.14.1	Введение . . . . .	46
4.14.2	Структурированный просмотр файлов . . . . .	46
4.14.3	Переключение между проектами . . . . .	47
4.14.4	Поиск и замена в проектах . . . . .	47
4.14.5	Простое автодополнение . . . . .	48
4.14.6	Поддержка сборки проекта . . . . .	49
4.14.7	Создание проектов . . . . .	49
4.14.7.1	Автоматическая загрузка проектов . . . . .	49
4.14.7.2	Создание проектов вручную . . . . .	49
4.14.8	Текущий проект . . . . .	51
4.14.9	Меню «Проекты» . . . . .	52
4.15	Модуль «Клиент LSP» . . . . .	52
4.15.1	Структура меню . . . . .	52
4.15.2	Поддержка перехода к идентификаторам . . . . .	54
4.15.2.1	Настройка структуры идентификаторов клиента LSP . . . . .	54
4.15.2.2	Поддержка глобального перехода к идентификаторам . . . . .	54
4.15.3	Другие возможности . . . . .	54
4.15.4	Настройка . . . . .	54
4.15.4.1	Конфигурация сервера LSP . . . . .	57
4.15.4.2	Форматирование при сохранении сервера LSP . . . . .	58
4.15.4.3	Подавление диагностики сервера LSP . . . . .	58
4.15.4.4	Устранение неполадок сервера LSP . . . . .	58
4.15.4.5	Настройка среды выполнения . . . . .	59
4.16	Модуль поиска и замены . . . . .	62
4.16.1	Введение . . . . .	62
4.16.2	Интерфейс . . . . .	62
4.16.2.1	Поисковый запрос . . . . .	62
4.16.2.2	Параметры поиска в папке . . . . .	63
4.16.2.3	Результаты поиска . . . . .	64
4.16.3	Структура меню . . . . .	64
4.17	Модуль «Сценарии и фрагменты» Kate . . . . .	65
4.17.1	Введение . . . . .	65
4.17.2	Структура меню . . . . .	65
4.17.3	Панель фрагментов . . . . .	65
4.17.3.1	Загрузка файлов комплектов фрагментов . . . . .	66
4.17.3.2	Создание и редактирование комплектов . . . . .	66

## Руководство пользователя Kate

4.17.3.3	Создание и редактирование фрагментов . . . . .	67
4.17.4	Использование фрагментов . . . . .	70
4.17.5	Благодарности . . . . .	70
4.18	Модуль клавиатурных макросов . . . . .	70
4.18.1	Введение . . . . .	70
4.18.2	Основные функции . . . . .	70
4.18.2.1	Чтобы начать запись клавиатурного макроса: . . . . .	70
4.18.2.2	Чтобы завершить запись: . . . . .	71
4.18.2.3	Чтобы отменить запись: . . . . .	71
4.18.2.4	Чтобы воспроизвести текущий макрос: . . . . .	71
4.18.3	Именованные макросы . . . . .	71
4.18.3.1	Чтобы сохранить текущий макрос: . . . . .	71
4.18.3.2	Чтобы загрузить сохранённый макрос в качестве текущего: . . . . .	71
4.18.3.3	Чтобы воспроизвести сохранённый макрос без его загрузки: . . . . .	71
4.18.3.4	Чтобы стереть (то есть удалить) сохранённый макрос: . . . . .	72
4.18.3.5	Советы по командам: . . . . .	72
4.18.4	Ограничения . . . . .	72
4.19	Модуль SQL . . . . .	72
4.19.1	Введение . . . . .	72
4.19.2	Подключение к базе данных . . . . .	72
4.19.3	Выполнение запросов . . . . .	73
4.19.3.1	INSERT/DELETE/UPDATE . . . . .	73
4.19.3.2	SELECT . . . . .	73
4.19.4	Просмотр . . . . .	74
4.19.5	Структура меню . . . . .	74
4.19.6	Благодарности . . . . .	74
4.20	Модуль просмотра идентификаторов . . . . .	74
4.20.1	Работа с модулем «Закрыть кроме/только...» . . . . .	74
4.20.2	Структура меню . . . . .	75
4.20.3	Настройка . . . . .	75
4.21	Модуль встроенного терминала . . . . .	75
4.21.1	Структура меню . . . . .	76
4.21.2	Настройка . . . . .	76
4.22	Модуль текстового фильтра . . . . .	77
4.22.1	Работа с модулем текстового фильтра . . . . .	77
4.22.2	Структура меню . . . . .	77
4.23	Модуль проверки XML . . . . .	78
4.23.1	Структура меню . . . . .	78
4.23.2	Благодарности . . . . .	78
4.24	Модуль автодополнения для XML . . . . .	78
4.24.1	Использование . . . . .	79
4.24.2	Возможности и ограничения . . . . .	79

## Руководство пользователя Kate

4.24.3	Структура меню . . . . .	79
4.24.4	Благодарности . . . . .	80
4.25	Модуль просмотра компиляции . . . . .	80
4.25.1	Использование . . . . .	80
4.26	Модуль форматирования . . . . .	80
4.26.1	Использование . . . . .	81
4.26.2	Поддерживаемые языки и средства форматирования . . . . .	81
4.26.3	Настройка . . . . .	82
4.26.4	Временно отключить форматирование при сохранении . . . . .	82
5	Дополнительные возможности редактирования	83
6	Расширение функциональных возможностей Kate	84
6.1	Введение . . . . .	84
6.2	Работа с подсветкой синтаксиса . . . . .	84
6.3	Написание сценариев JavaScript . . . . .	84
6.4	Модули Kate (C++) . . . . .	84
7	Режим ввода Vi	85
8	Команды меню	86
8.1	Меню «Файл» . . . . .	86
8.2	Меню «Правка» . . . . .	87
8.3	Меню «Вид» . . . . .	90
8.4	Меню «Закладки» . . . . .	93
8.5	Меню «Сеанс» . . . . .	94
8.6	Меню «Сервис» . . . . .	94
8.7	Меню «Настройка» и «Справка» . . . . .	95
9	Настройка Kate	97
9.1	Обзор . . . . .	97
9.2	Основной диалог настройки . . . . .	98
9.3	Настройка приложения Kate . . . . .	98
9.3.1	Общие . . . . .	98
9.3.2	Сеансы . . . . .	99
9.3.3	Модули . . . . .	100
9.3.4	Настройка компонента редактирования . . . . .	100
9.3.5	Настройка с помощью переменных документа . . . . .	101
10	Авторские права и лицензия	102
A	Регулярные выражения	104
B	Предметный указатель	105

## Список таблиц

4.1	Некоторые доступные модули KParts . . . . .	35
-----	---	----

## Аннотация

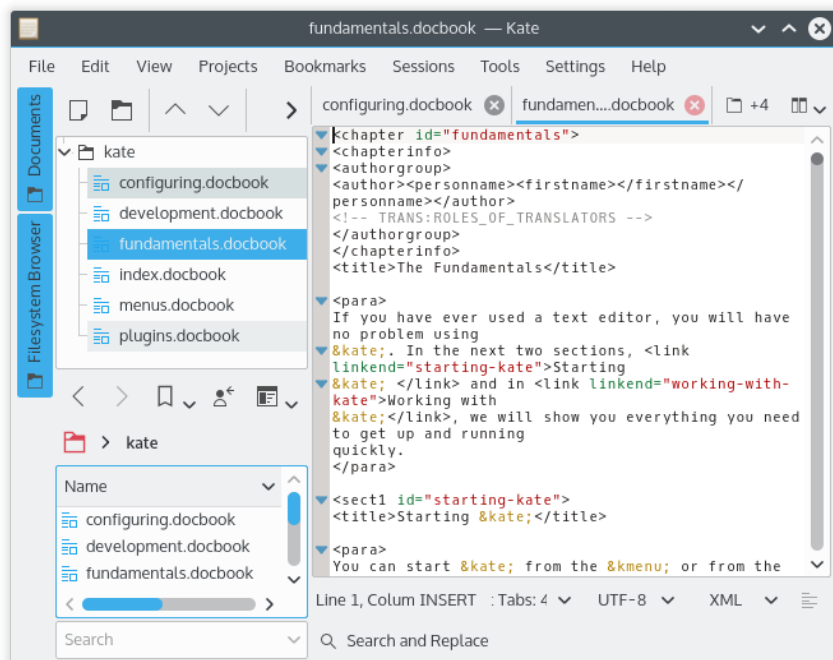
Kate — это текстовый редактор для программистов, разработанный KDE.  
В данном руководстве описан редактор Kate версии 22.08

## Глава 1

## Введение

Добро пожаловать в Kate — улучшенный текстовый редактор для программистов, который работает в среде KDE. Настраиваемая подсветка синтаксических конструкций для широкого диапазона языков программирования (начиная от C, C++ и HTML и заканчивая сценариями bash), возможность создавать и разрабатывать проекты, многодокументный интерфейс (MDI), встроенный эмулятор терминала — вот лишь небольшая часть возможностей Kate.

Kate — это больше, чем текстовый редактор для программистов. Многодокументный интерфейс позволяет открывать и редактировать сразу несколько конфигурационных файлов UNIX®. Это руководство пользователя было создано с помощью Kate.



Редактирование этого руководства...

## Глава 2

# ОСНОВЫ

Перевод на русский язык: Андрей Балагута  
Обновление перевода: Николай Шафоростов  
Обновление перевода: Мария Шикунова  
Обновление и редактирование перевода: Олеся Герасименко

У пользователей, имеющих опыт использования текстовых редакторов, проблем с работой в Kate возникнуть не должно. Следующие два раздела [Запуск Kate](#) и [Работа в Kate](#) позволят узнать всё, что нужно для начала продуктивной работы с редактором.

### 2.1 Запуск Kate

Запуск Kate выполняется из меню запуска приложений или из командной строки.

#### 2.1.1 Из меню запуска

Откройте меню программ KDE, щёлкнув левой кнопкой мыши по значку на панели инструментов, который расположен в левом нижнем углу экрана. В появившемся меню запуска выберите пункт Приложения → Служебные → Kate (Улучшенный текстовый редактор) или просто введите «kate» в строке поиска.

#### 2.1.2 Из командной строки

Чтобы запустить Kate, введите «kate» в командной строке. Если указать имя файла в качестве параметра, как это сделано в приведённом ниже примере, то редактор откроет или создаст этот файл.

```
%kate myfile.txt
```

При наличии доступа в Интернет появляется возможность открывать и сохранять удалённые файлы благодаря реализации «прозрачной» работы с сетью в KDE.

```
%kate ftp://ftp.kde.org/pub/kde/README
```

Чтобы изменить каталог для сохранения временных файлов (по умолчанию /tmp), укажите переменную среды TMPDIR перед запуском Kate, например:

```
%mkdir /tmp/kate -p && export TMPDIR=/tmp/kate && kate
```

### 2.1.2.1 Параметры командной строки

При запуске Kate из командной строки возможно указать следующие ключи:

`kate --help`

Выводит список всех доступных команд.

`kate -s --start имя_сеанса`

Запускает Kate и открывает сеанс `имя_сеанса`. Если указанный сеанс не существует, он будет создан. Если программа Kate с данным сеансом уже запущена, то переданные в командной строке файлы будут открыты в уже существующем окне программы.

`kate -p --pid PID`

Открывает файл в экземпляре с указанным PID (идентификатор процесса).

`kate -e --encoding кодировка URL`

Использует определённую кодировку для документа.

`kate -l --line номер_строки URL`

Переходит к строке с указанным номером после открытия документа.

`kate -c --column позиция URL`

Переходит в указанную позицию строки после открытия документа.

`kate -i --stdin`

Получает текст из стандартного ввода. Аналогично указанию `-`, используемому во многих программах с наличием командной строки, и позволяет направить вывод команды в Kate.

`kate --startanon`

Запускает Kate с новым безымянным сеансом, подразумевает параметр `-n`.

`kate -n --new`

Принудительно запускает новый экземпляр Kate (игнорируется, если используется `start` и указанный сеанс уже открыт в другом экземпляре Kate); данный параметр применяется автоматически, если не указаны какие-либо параметры или ссылки.

`kate -b --block`

Если используется уже запущенный экземпляр Kate, выполняет блокировку до выхода из него, если указаны ссылки для открытия.

С этой опцией Kate возможно использовать в качестве редактора для набора сообщений фиксации для таких систем контроля версий, как Git или Subversion. Для работы с этими системами необходима блокировка редактора до того момента, пока пользователь не напишет сообщение, потому что затем они открывают временный файл, который в противном случае был бы пустым.

Эта опция также используется в КИО (ввод/вывод KDE) при открытии сетевого файла (загруженного в качестве временного), который требует повторной отправки после сохранения.

`kate --tempfile`

Использует указанные файлы как временные и удаляет их (если они локальные, а у пользователя достаточно прав) при закрытии, если только они не были изменены с момента их открытия.

`kate --desktopfile имя_файла`

Имя файла `.desktop` (без пути) для этого приложения.

Это особенно полезно для приложений-оболочек и приложений, имеющих, как правило, несколько файлов рабочей области. У каждого файла рабочей области может быть своя командная строка для записи `Exes`.

- `kate --author`  
Показывает список авторов Kate в окне терминала.
- `kate -v --version`  
Показывает сведения о версии Kate.
- `kate --license`  
Показывает информацию о лицензии.

### 2.1.3 Перетаскивание

Kate поддерживает протокол перетаскивания мышью KDE. Файлы возможно перетащить в Kate с рабочего стола, из диспетчера файлов Dolphin или с какого-либо сетевого FTP-сайта, открытого в окне Dolphin.

## 2.2 Работа в Kate

В разделе [Начало работы](#) рассказано о четырёх опциях, которые позволят быстро настроить основные возможности Kate. Раздел [Комбинации клавиш](#) содержит описание некоторых комбинаций клавиш для тех, кто не может или не желает работать с мышью.

### 2.2.1 Начало работы

В этом разделе описываются несколько подпунктов меню Вид, которые помогут быстро настроить Kate необходимым образом.

При первом запуске Kate появятся две области с белым фоном. Над ними будет находиться панель инструментов с наиболее часто используемыми командами. Ещё выше — строка меню.

Область слева — это боковая панель. В ней доступны список открытых файлов (вкладка Документы) и проводник (Просмотр файловой системы) в виде «служебных панелей». Переключаться между ними возможно с помощью ярлычков, которые находятся слева от боковой панели.

Если программа Kate запущена для открытия файла (например, из Dolphin), то в правой области появится текст редактируемого файла, а в списке файлов, который находится на боковой панели, — его имя. С помощью селектора файлов возможно просматривать файловую структуру и открывать файлы для редактирования.

Включить и отключить все служебные панели возможно либо с помощью меню Вид → Боковые панели, либо с помощью комбинации клавиш `Ctrl+Alt+Shift+F`. При использовании этого меню становятся очевидны широта возможностей и гибкость Kate.

В меню Боковые панели отображается список всех подключённых модулей. Чтобы показать или скрыть служебную панель, установите флажок напротив соответствующего пункта списка или щёлкните левой кнопкой мыши по соответствующей кнопке на боковой панели.

### 2.2.2 Комбинации клавиш

Почти все комбинации клавиш Kate возможно перенастроить с помощью меню [Настройка](#). По умолчанию комбинации клавиш в Kate настроены следующим образом:

F1	Справка
Shift+F1	Что это?
Ctrl+N	Создать документ
Ctrl+L	Сохранить всё
Ctrl+O	Открыть документ

Ctrl+Alt+O	Быстрый переход
Ctrl+Shift+F	Полноэкранный режим
Ctrl+Shift+,	Настроить Kate
Ctrl+W / Ctrl+Esc	Заккрыть
Ctrl+Q	Выход — закрыть редактор
Ctrl+Alt+Shift+F	Показать боковые панели
Ctrl+Shift+T	Разделить по горизонтали
Ctrl+Shift+L	Разделить по вертикали
F8	Следующая створка
Shift+F8 / Ctrl+Esc	Предыдущая створка
Ctrl+Shift+R	Заккрыть текущую створку
Alt+Стрелка вправо	Следующая вкладка
Alt+Стрелка влево	Предыдущая вкладка
Ctrl+Shift+T	Открыть последние закрытые документы заново

Кроме того, возможно использовать комбинации клавиш, предоставляемые компонентом [KatePart](#) и всеми подключёнными модулями [модулями Kate](#).

## 2.3 Работа с многодокументным интерфейсом Kate

### 2.3.1 Обзор

Окно, представление, документ, панель, редактор... Что означают эти понятия в терминологии Kate? Эта глава даст ответы на все вопросы, связанные с окнами Kate.

#### 2.3.1.1 Главное окно

Главное окно Kate представляет собой стандартное окно приложения KDE, в которое добавлены боковые панели с представлениями инструментов. Главное окно содержит строку меню с расположенными на ней стандартными меню и панель инструментов, на которой находятся кнопки часто используемых команд.

Самая важная часть окна — это область редактирования; по умолчанию она содержит один компонент для редактирования текста, в котором происходит работа с документами.

Возможности закрепления окна используются для окон инструментов любого модуля, включённого в диалоге настройки параметров.

Размещение служебных панелей возможно изменить. Для этого щёлкните правой кнопкой мыши по вкладке панели и выберите край экрана, в который следует её перенести.

Служебная панель может иметь Фиксированное размещение (устанавливается вызовом контекстного меню с помощью щелчка правой кнопкой мыши). В этом случае при переходе к другой служебной панели последней будет выделено не всё пространство, а при её скрытии будет показана служебная панель, имеющая фиксированное размещение.

Если у подключаемого модуля есть параметры конфигурации, первый пункт контекстного меню позволяет открыть соответствующую страницу в диалоге настройки Kate.

### 2.3.2 Область редактирования

В Kate возможно редактировать несколько документов одновременно, разделив область редактирования на панели (наподобие тех, что используются в `Konqueror` и `emacs`). С помощью

панелей (фреймов) возможно просматривать в области редактирования несколько документов (или несколько экземпляров одного документа). Например, если в начале файла расположены важные определения, с которыми нужно часто сверяться, эту возможность сложно оценить по достоинству. Ещё один пример: при написании программы удобно будет в одной панели открыть заголовочный файл, а в другой — файл реализации.

Если один и тот же документ открыть в нескольких панелях редактора, то все изменения (редактирование и выделение текста), сделанные в одной панели, будут немедленно отражены во всех остальных. Операция поиска и перемещение курсора действуют только в активной панели.

В текущей версии Kate нельзя открыть один и тот же документ несколько раз в том смысле, что редактироваться будет только один экземпляр, и изменения не будут отражаться во втором.

После разделения редактора на две панели они имеют одинаковую ширину и отображают текущий документ этого редактора. Новая панель будет снизу (в случае разделения по горизонтали) или справа (в случае разделения по вертикали). Активной становится новая панель — об этом говорит мигающий текстовый курсор.

## 2.4 Работа с сеансами

Сеанс позволяет открывать несколько документов и настраивать расположение элементов интерфейса на своё усмотрение. Поддерживается возможность создания бесконечного количества именованных сеансов, а также использования безымянных и анонимных сеансов для файлов, открываемых только один раз. Благодаря реализации концепции сеансов появилась возможность запуска нескольких экземпляров Kate одновременно — с разными сеансами в каждом.

Поддержка сеансов присутствует в трёх областях:

- [Ключи командной строки](#) — возможность выбора сеанса при открытии Kate с помощью командной строки.
- [Меню «Сеанс»](#) — позволяет переключаться между сеансами, сохранять и настраивать их.
- [Окно настройки](#) — позволяет настроить параметры работы с сеансами.

При создании нового сеанса используется конфигурация Сеанс по умолчанию. Чтобы сохранить конфигурацию окна в этом стандартном сеансе, включите сохранение конфигурации окна на странице параметров сеансов (в окне настройки Kate), загрузите стандартный сеанс, настройте параметры окна и сохраните сеанс снова.

При загрузке любого сохранённого сеанса в заголовке окна Kate отображается его имя. Следовательно, заголовок окна выглядит следующим образом: «имя\_сеанса: имя\_документа или ссылка — Kate».

Если при открытии файлов из командной строки указывается ключ `--start имя_сеанса` или выбирается сеанс в диалоге выбора сеансов, то перед открытием указанных в командной строке файлов загружаются файлы соответствующего выбранного сеанса. Kate позволяет всегда открывать файлы в новом сеансе. Для этого установите соответствующий флажок на странице параметров сеансов диалога настройки программы или укажите ключ `--start` с пустым аргументом: `''`.

Начиная с Kate 2.5.1 PID текущего экземпляра экспортируется в переменную окружения `KATE_PID`. Для открытия файлов из встроенного терминала используется текущий экземпляр программы, если явно не указано иного.

## 2.5 Быстрый переход

В Kate встроен диалог для быстрого открытия файлов и переключения между ними. Чтобы вызвать этот диалог, воспользуйтесь комбинацией клавиш `Ctrl+Alt+O`.

В окне этого диалога можно просмотреть все открытые в Kate документы, а также все файлы в открытых проектах. Для просмотра файлов проектов следует включить [Модуль проектов](#).

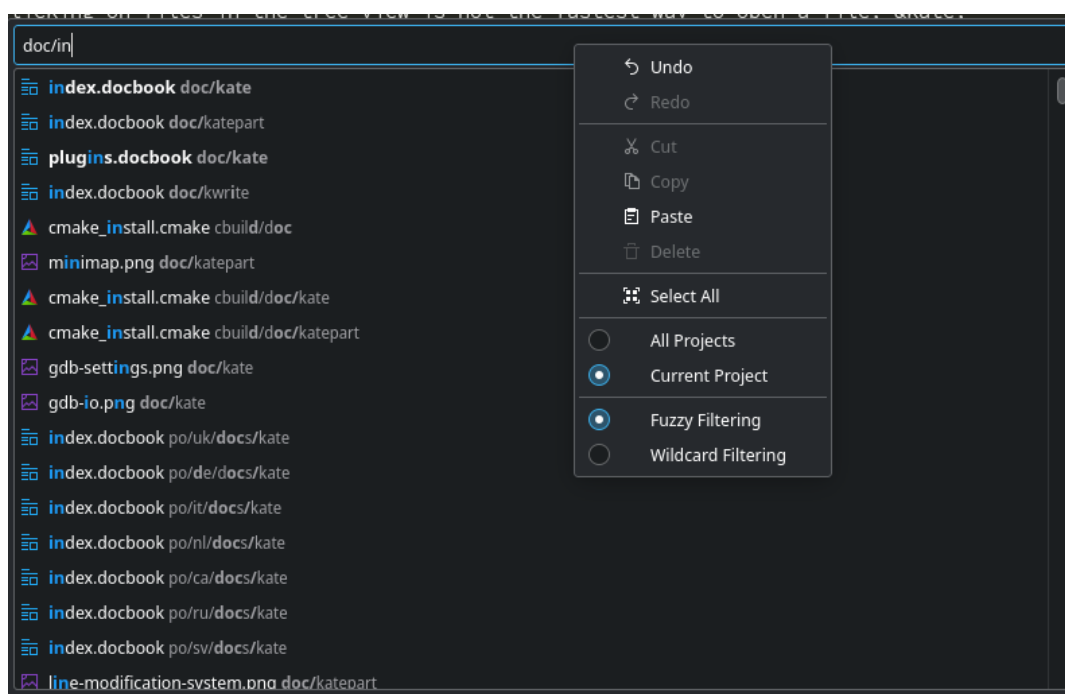
## 2.5.1 Работа с быстрым переходом

Использовать быстрый переход очень просто: откройте соответствующее диалоговое окно и введите (полностью или частично) имя файла, который требуется открыть. Показанный в окне список файлов будет отфильтрован. Выбранный файл открывается по нажатию клавиши ввода, а при нажатии клавиши Esc диалог быстрого открытия будет скрыт.

По умолчанию при фильтрации поиск ведётся только по имени файла. Чтобы учитывался и путь, в набираемом тексте должен быть символ «/». Например: «doc/index» позволит найти все файлы, в имени которых присутствует текст «index», в папке «doc».

Уже открытые документы выделяются полужирным шрифтом и отображаются в верхней части диалогового окна. Кроме того, при открытии диалогового быстрого перехода будет выбран предыдущий открытый документ, поэтому для перехода к нему достаточно нажать клавишу ввода.

## 2.5.2 Настройка быстрого перехода



В диалоге быстрого перехода предусмотрено несколько параметров настройки. Чтобы получить доступ к ним, щелкните правой кнопкой мыши в строке ввода.

В текущей версии программы доступны следующие параметры:

- Текущий проект — показать файлы только из текущего проекта
- Все проекты — показать файлы из всех открытых проектов
- Нечёткая фильтрация — использовать нечёткое сопоставление для фильтрации файлов
- Фильтрация по подстановочным знакам — использовать сопоставление по подстановочным знакам для фильтрации файлов

## 2.6 Получение справки

### 2.6.1 Получение справки в Kate

## Руководство пользователя Kate

### Данное руководство

Здесь детально описаны все команды меню, параметры настройки, инструменты, диалоговые окна, модули. Также здесь описано главное окно Kate, сам редактор и различные понятия и концепции, которые используются в приложении.

Чтобы открыть руководство, нажмите клавишу F1 или вызовите пункт меню Справка → Руководство пользователя Kate.

### Контекстная справка («Что это?»)

Контекстная справка позволяет сразу получить сведения о каком-либо элементе графического окна, будь то кнопка или любая другая область окна.

Разработчики постарались обеспечить контекстной справкой все элементы, для которых это необходимо. Окно настройки Kate полностью снабжено контекстной справкой, также как и многие другие диалоговые окна.

Чтобы вызвать контекстную справку, воспользуйтесь комбинацией клавиш Shift+F1 или выберите пункт меню Справка → Что это?. Указатель мыши примет вид стрелки со знаком вопроса; теперь достаточно просто щёлкнуть левой кнопкой мыши по нужному элементу окна — небольшое окно с контекстной справкой тут же появится на экране (если, конечно, такая справка доступна).

### Кнопки справки в диалоговых окнах

В некоторых диалоговых окнах присутствует кнопка Справка. При нажатии этой кнопки будет открыт соответствующий раздел документации в KHelpCenter.

## 2.6.2 Получение справки из внешних файлов

В настоящее время Kate не поддерживает чтение документации, относящейся к редактируемому документу. В зависимости от типа редактируемого файла, может быть полезен [встроенный эмулятор терминала](#), например, для просмотра руководств UNIX® (map-страницы) или документации в формате info. Также возможно использовать Konqueror.

## 2.6.3 Статьи про Kate

Дополнительная информация доступна на [домашней странице Kate](#).

## Глава 3

# Работа с редактором Kate

Более подробные сведения об основах работы с компонентом редактирования Kate доступны в разделе «Работа с редактором KatePart» руководства пользователя KatePart.

## Глава 4

# Работа с модулями

Anders Lund

Перевод на русский язык: Андрей Балагута

Обновление перевода: Мария Шикунова

Обновление и редактирование перевода: Олеся Герасименко

Все модули подключаются в [диалоге настройки параметров](#). В этом диалоге также возможно настроить сами модули, если это предусмотрено.

### 4.1 Модули Kate

Модули Kate позволяют воспользоваться дополнительными функциями редактора. Подключение того или иного модуля расширяет функционал Kate, иногда даже добавляя новые пункты меню и ярлыки. Количество одновременно подключённых модулей не ограничено. Откройте окно настройки параметров: [Настройка](#) → [Настроить Kate...](#) — выберите пункт [Приложение](#) → [Модули](#) и установите флажки для подключения соответствующих модулей.

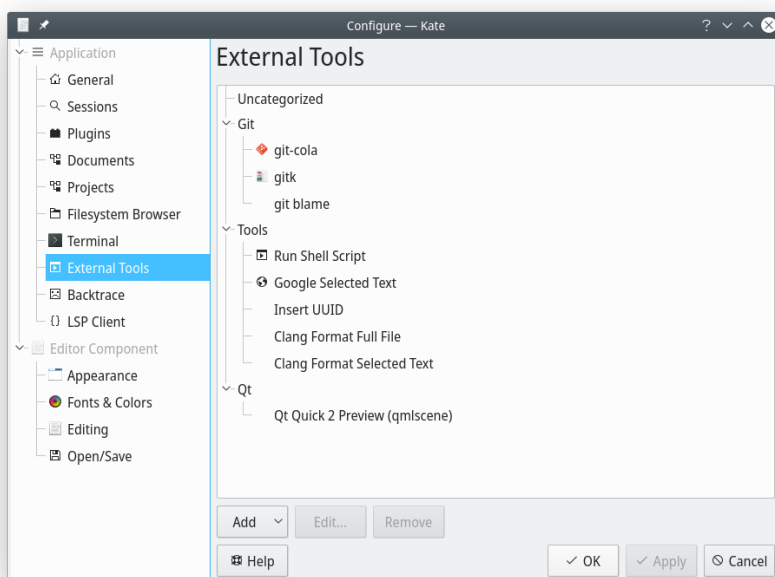
Доступны следующие модули:

- [Внешние программы](#) — запуск внешних программ
- [Просмотр стека вызовов](#) — инструмент для просмотра стека вызовов C и C++
- [Модуль для сборки ПО](#) — сборка и обработка сообщений об ошибках
- [Закреть кроме/только...](#) — закрывает группу документов, в зависимости от пути или расширения
- [Выбор цвета](#) — предпросмотр известных цветов
- [Цветные скобки](#) — выделение скобок цветом для удобочитаемости
- [CTags](#) — поиск определений и объявлений с помощью индекса CTags
- [Предварительный просмотр документа](#) — предварительный просмотр документа в целевом формате
- [Переключатель документов](#) — переключатель документов с поведением как у переключателя окон по [Alt+Tab](#)
- [Обзор файловой системы](#) — панель для просмотра файловой системы
- [Иерархический просмотр документов](#) — просмотр открытых документов в дереве папок
- [GDB](#) — простой интерфейс к отладчику GDB
- [Модуль проектов](#) — интеграция с Git, а также с другими системами управления версиями

- Replicode — конструктивистский язык программирования ИИ и исполняемые модули
- Клиент LSP — поддержка Language Server Protocol
- Поиск и замена — поиск и замена в открытых документах, папках и проектах на диске
- Сценарии и фрагменты — создание или загрузка фрагментов кода и сценариев редактора
- Клавиатурные макросы — запись и воспроизведение клавиатурных макросов (последовательностей действий на клавиатуре)
- Модуль SQL — выполнение запросов SQL к базам данных
- Просмотр идентификаторов — извлекает и показывает идентификаторы C/C++ из исходного кода
- Встроенный терминал — быстрый вызов терминала, основанного на виджете Konsole
- Текстовый фильтр — выполнение операций с текстом с помощью командной строки
- Автодополнение для XML — перечисляет элементы XML, их атрибуты, значения атрибутов и записи, разрешённые DTD
- Проверка XML — проверка правильности XML с использованием xmllint
- Просмотр компиляции — интерфейс к средству просмотра компиляции
- Форматирование — модуль для форматирования кода

## 4.2 Внешние программы

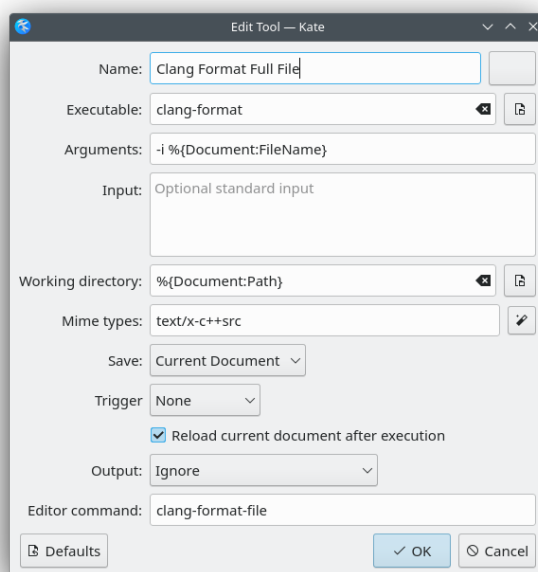
Модуль Внешние программы позволяет вызывать внешние приложения, передавая им данные, которые связаны с текущим документом (например, его URL-адрес, каталог, текст или выделенный фрагмент). После включения модуля появляется страница настройки (она показана далее), на которой возможно изменить или удалить существующие программы. Также возможно добавить новые программы. После добавления они появятся в подменю Внешние программы меню Сервис.



На странице настройки возможно добавить новые внешние программы, нажав кнопку **Добавить**. После этого появится всплывающее меню, которое позволяет добавить новую внешнюю программу, добавить существующую программу из предустановленного списка или добавить новую категорию для организации внешних программ. Существующие программы возможно изменить либо по двойному щелчку, либо нажатием кнопки **Изменить....** Кнопка **Удалить** позволяет удалить выбранные программы.

### 4.2.1 Настройка внешних программ

При редактировании программы открывается диалог, который позволяет выполнить тонкую настройку:



Возможно задать много параметров, в частности:

**Название** — название программы, которое затем появится в меню.

**Значок** — необязательный значок, который показан в меню.

**Исполняемый файл** — название исполняемого файла. Следует указать либо полный путь, либо просто название программы (если исполняемый файл хранится в одном из каталогов, описанных переменной среды PATH).

**Аргументы** — необязательные аргументы, которые передаются исполняемому файлу.

**Ввод** — необязательные входные данные, которые передаются процессу через stdin.

**Рабочий каталог** — рабочий каталог, из которого будет запущена программа. Если значение не указано, рабочим каталогом будет каталог текущего документа.

**Типы MIME** — если значение указано, программа становится активной только в том случае, если тип MIME текущего документа соответствует указанному.

**Сохранение до запуска** — при вызове не сохраняется ничего, сохраняется текущий документ или сохраняются все документы.

**Активация, триггер для выполнения программы**. Влияет только на текущий активный документ и срабатывает только в том случае, если тип MIME текущего активного документа соответствует типу MIME внешней программы.

Допустимы следующие триггеры:

Нет — значение по умолчанию, означает, что у программы нет переключателя.

До сохранения, этот триггер запускает программу непосредственно перед сохранением документа.

После сохранения, этот триггер запускает программу после сохранения документа.

Перечитать текущий документ после выполнения — эта опция полезна в тех случаях, когда текущий файл был изменён на диске.

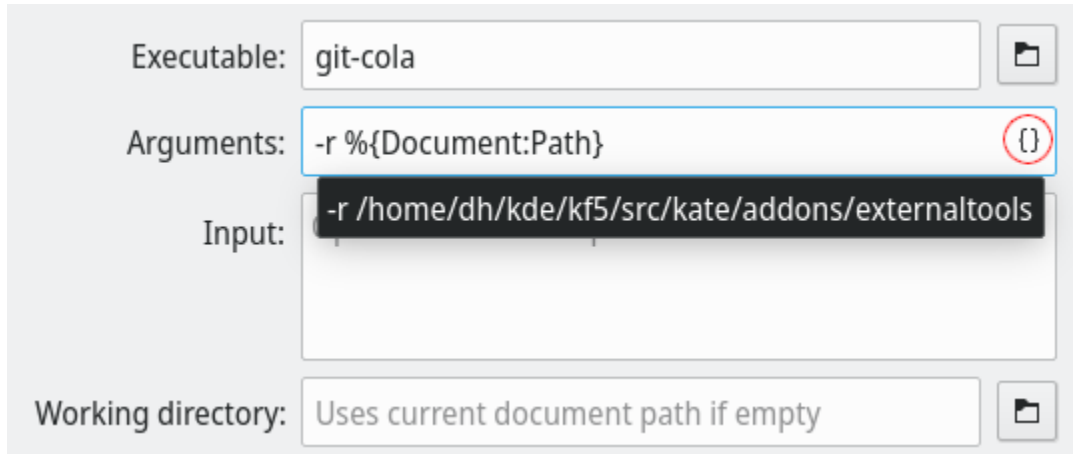
Выходные данные — вывод определяет цель stdout. Доступные варианты: Игнорировать, Вставить в текущей позиции, Заменить выбранный текст, Заменить текущий документ, Добавить в текущий документ, Вставить в новый документ, Копировать в буфер обмена или Вывести в панели.

Командная строка редактора — необязательная команда, которая будет использоваться для вызова внешней программы с помощью встроенной [командной строки](#).

Кнопка По умолчанию отображается только для программ, которые поставляются с Kate. При её нажатии все параметры программы возвращаются к стандартным (заводским) значениям.

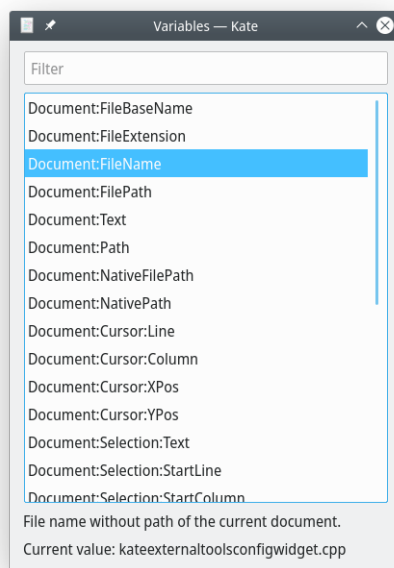
#### 4.2.2 Расширение переменных

Некоторые поля ввода, например Исполняемый файл, Аргументы, Ввод и Рабочий каталог, поддерживают переменные, которые расширяются при вызове программы. Это обозначается значком {}, который появляется при получении фокуса одним из этих полей ввода (см. красный кружок):



При наведении курсора на одно из таких полей ввода также отображается всплывающая подсказка с текущим расширенным текстом. Кроме того, при щелчке по значку {} открывается диалог с перечнем всех доступных переменных:

## Руководство пользователя Kate



Эта опция позволяет сделать настройку внешней программы очень гибкой, так как все переменные вида `%{...}` расширяются при вызове этой программы. Поддерживаются переменные двух типов:

- `%{имя-переменной}`
- `%{имя-переменной:<значение>}`

Если применяется первый тип `%{имя-переменной}`, программа просто заменяет переменную его содержимым. Например, переменная `%{Document:FileName}` заменяется именем файла текущего документа без пути к нему. Если применяется второй тип, `%{имя-переменной:<значение>}`, в качестве содержимого используется `<значение>`. Например, этим можно воспользоваться для расширения переменной среды с помощью `%{ENV:HOME}`. Или можно получить текущую дату в предпочитаемом формате: `%{Date:yyyy-MM-dd}`.

Среди поддерживаемых переменных:

`Document:FileName`: базовое имя файла текущего документа без пути и суффикса.

`Document:FileExtension`: расширение файла текущего документа.

`Document:FileName`: имя файла текущего документа без пути.

`Document:FilePath`: полный путь к текущему документу, включая название файла.

`Document:Text`: содержимое текущего документа.

`Document:Path`: полный путь к текущему документу, исключая название файла.

`Document:NativeFilePath`: полный путь к документу, включая название файла, с разделителем пути операционной системы (в Windows — обратная косая черта).

`Document:NativePath`: полный путь к документу, исключая название файла, с разделителем пути операционной системы (в Windows — обратная косая черта).

`Document:Cursor:Line`: номер строки позиции текстового курсора в текущем документе (нумерация начинается с 0).

`Document:Cursor:Column`: номер столбца позиции текстового курсора в текущем документе (нумерация начинается с 0).

`Document:Cursor:XPos`: компонент X глобальной позиции курсора на экране.

Document:Cursor:YPos: компонент Y глобальной позиции курсора на экране.  
Document:Selection:Text: выбранный текст в текущем документе.  
Document:Selection:StartLine: начальная строка выбранного текста в текущем документе.  
Document:Selection:StartColumn: начальный столбец выбранного текста в текущем документе.  
Document:Selection:EndLine: конечная строка выбранного текста в текущем документе.  
Document:Selection:EndColumn: конечный столбец выбранного текста в текущем документе.  
Document:RowCount: количество строк в текущем документе.  
Document:Variable:<переменная>: развернуть произвольные [переменные документа](#).  
Date:Locale: текущая дата в формате текущей локали.  
Date:ISO: текущая дата (ISO).  
Date:<значение>: текущая дата (в формате, указанном строкой формата QDateTime).  
Time:Locale: текущее время в формате текущей локали.  
Time:ISO: текущее время (ISO).  
Time:<значение>: текущее время (в формате, указанном строкой формата QDateTime).  
ENV:<значение>: доступ к переменным среды.  
JS:<выражение>: обработка простых инструкций JavaScript.  
PercentEncoded:<текст>: текст в процентной кодировке.  
UUID: создать новый UUID.

### 4.2.3 Список стандартных программ

По умолчанию поставляются несколько программ. Если у пользователя имеются ещё какие-либо полезные инструменты, рекомендуется поделиться ими через [проект на GitLab](#) (чтобы разработчики смогли добавить их в список стандартных). Все стандартные программы по умолчанию показаны в списке. Все эти программы возможно изменить в соответствии с потребностями пользователя (например, изменить категорию или даже удалить программу из списка). Удалённые программы возможно вернуть в список нажатием кнопки [Добавить](#) на странице настройки.

Git Cola

Git Cola — это графический клиент git, который позволяет легко подготавливать и фиксировать изменения. Если этот клиент установлен, доступ к нему также возможно получить путём ввода команды git-cola в командной строке.

Название: Git Cola  
Значок: git-cola  
Исполняемый файл: git-cola  
Аргументы: -r %[Document:Path](#)  
Команда редактора: git-cola

gitk

gitk — также клиентская программа git, позволяет визуализировать журнал изменений в git.

Название: gitk  
Значок: git-gui  
Исполняемый файл: gitk  
Рабочий каталог: %[Document:Path](#)  
Команда редактора: gitk

## git blame

Запускает git blame для облегчения слежения за изменениями текущего файла в git.

Название: git blame  
Исполняемый файл: git  
Аргументы: gui blame `{Document:FileName}`  
Сохранение до запуска: текущий документ  
Рабочий каталог: `{Document:Path}`  
Команда редактора: git-blame

## Запуск сценария оболочки

Запускает внешнюю консоль, в которой выполняется текущий документ. Сценарий должен объявлять в первой строке свой интерпретатор с помощью следующей последовательности символов: `#!/путь/интерпретатор`.

Название: запуск сценария оболочки  
Значок: system-run  
Исполняемый файл: konsole  
Аргументы: `-e sh -c "cd {Document:Path} && pwd && chmod -vc a+x {Document:FileName} && ./{Document:FileName} ; echo Press any key to continue. && read -n 1"`  
Сохранение до запуска: текущий документ  
Рабочий каталог: `{Document:Path}`  
Команда редактора: run-script

## Поиск выделенного текста в Google

Поиск выделенного текста с помощью поисковой системы Google.

Название: поиск выделенного текста в Google  
Значок: globe  
Исполняемый файл: xdg-open  
Аргументы: `"https://www.google.com/search?q={Document:Selection:Text}"`  
Команда редактора: google

## Вставка UUID

Вставляет новый UUID при каждом вызове этого действия.

Название: вставка UUID  
Исполняемый файл: echo  
Аргументы: `{UUID}`  
Выходные данные: вставить в текущей позиции  
Команда редактора: uuid

## Форматирование Clang всего файла

## Руководство пользователя Kate

Выполняет clang-format для текущего файла на диске. После обработки программа повторно загружает документ.

Название: форматирование Clang всего файла

Исполняемый файл: clang-format

Аргументы: -i `{Document:FileName}`

Рабочий каталог: `{Document:Path}`

Сохранение до запуска: текущий документ

Перечитать: да

Команда редактора: clang-format-file

Форматирование Clang выделенного текста

Выполняет clang-format только для выбранного текста в текущем документе.

Название: форматирование Clang выделенного текста

Исполняемый файл: clang-format

Аргументы: -assume-filename: `{Document:FileName}`

Рабочий каталог: `{Document:Path}`

Ввод: `{Document:Selection:Text}`

Выходные данные: заменить выбранный текст

Команда редактора: clang-format-selection

Предварительный просмотр Qt Quick 2 (qmlscene)

Предварительный просмотр текущего файла QML в qmlscene.

Название: предварительный просмотр Qt Quick 2 (qmlscene)

Исполняемый файл: qmlscene

Аргументы: `{Document:FileName}`

Сохранение до запуска: текущий документ

Рабочий каталог: `{Document:Path}`

Команда редактора: qml-preview

Форматирование JSON всего файла

Форматировать весь файл кода JSON.

Название: форматирование JSON всего файла

Значок: application-json

Исполняемый файл: jq

Аргументы: `{Document:FileName}`

Сохранение до запуска: текущий документ

Рабочий каталог: `{Document:Path}`

Выходные данные: заменить текущий документ

Команда редактора: json-format-file

Форматирование XML всего файла

Форматировать весь файл кода XML.

Название: форматирование XML всего файла

Значок: application-xml

Исполняемый файл: xmllint

Аргументы: --format %`{Document:FileName}`

Сохранение до запуска: текущий документ

Рабочий каталог: %`{Document:Path}`

Выходные данные: заменить текущий документ

Команда редактора: xml-format-file

## 4.3 Модуль просмотра стека вызовов

### 4.3.1 Работа с модулем «Просмотр стека вызовов»

Этот модуль предназначен для разработчиков и, скорее всего, не потребуется обычным пользователям. Он отображает стек вызовов, предоставленный GDB, в виде списка в панели Kate. При щелчке по элементу списка выбранный файл будет открыт на строке с нужным номером. Этот модуль работает со стеком вызовов, сгенерированным на пользовательском компьютере, но также может работать и со стеком вызовов других людей, например, с помощью `/home/dushmu/qt-cory/.../qwidget.cpp`. Чтобы это сработало, требуется проиндексировать каталоги, в которых находится исходный код.

Иногда может существовать несколько файлов с одинаковыми именами, например

```
kdegraphics/okular/generators/dvi/config.h
```

```
kdepim-runtime/resources/gmail/saslplugin/config.h
```

Чтобы сделать правильный выбор, модуль учитывает последние две части в адресе пути, в таком случае это

```
dvi/config.h
```

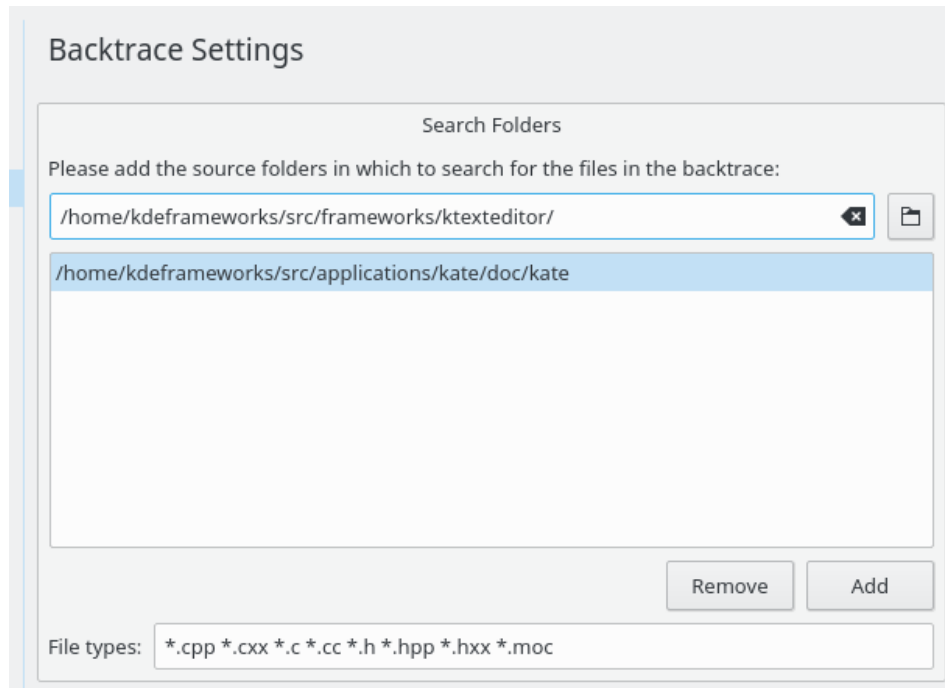
```
saslplugin/config.h
```

И затем модуль обычно находит нужный файл.

Индексирование основной ветки и побочной приведёт к конфликту.

### 4.3.2 Настройка

На странице настройки добавьте каталоги, содержащие исходный код.



Настройка путей в панели просмотра стека вызовов

Чтобы начать индексирование, нажмите кнопку ОК. Когда индексирование завершится, откройте панель Просмотрщик стека вызовов.

Теперь возможно загрузить стек вызовов из буфера обмена (например, после нажатия Скопировать в буфер обмена в DgKonqi) или из файла.

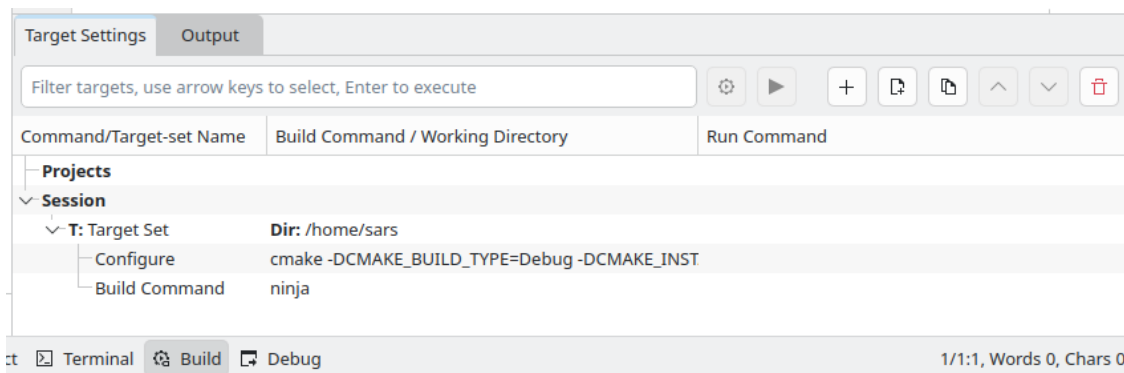
## 4.4 Модуль для сборки ПО

### 4.4.1 Введение

Модуль для сборки ПО позволяет выполнять сборку, очистку и компиляцию проекта. Кроме того, можно автоматически запускать созданные приложения. По сути, модуль предоставляет способ настройки наборов команд для запуска, а также позволяет выполнить синтаксический анализ вывода, чтобы найти ссылки на файлы и определённые строки и позиции в этих файлах. Этот модуль изначально был создан для компиляции кода на языке C/C++ и в основном тестировался в связи с этой задачей, но его также можно использовать для других задач и языков.

### 4.4.2 Работа с модулем для сборки ПО

Этот модуль добавляет панель Сборка в нижней части окна, а также меню Сборка в строке меню. Панель позволяет выбирать и настраивать цели сборки, а меню и его комбинации клавиш возможно использовать для выбора и выполнения настроенных команд оболочки.



На панели Сборки доступны две статические вкладки:

- Параметры цели
- Вывод

#### 4.4.2.1 Вкладка «Параметры цели»

Вкладка «Параметры цели» используется для настройки различных целей сборки и определения наборов целей.

Набор целей — это группа команд, которые можно запускать в указанном рабочем каталоге. Наборы целей можно добавлять либо на панели Проекты, либо на панели Сеанс. Пользовательские наборы целей, добавленные на панели Проекты, хранятся в файле `.kateproject.build` в корневом каталоге проекта и восстанавливаются при повторном открытии проекта. Наборы целей, добавленные на панели Сеанс, хранятся в конфигурации сеанса Kate.

Первая строка в наборе целей содержит название набора (в первом столбце) и каталог, в котором должны выполняться команды (во втором столбце). В каждой последующей строке находятся название команды (в первом столбце), команда сборки (во втором столбце), команда запуска (в третьем столбце). Чтобы выполнить редактирование, сделайте двойной щелчок по нужной записи или нажмите связанную клавишу (часто это F2).

##### Рабочий каталог

Второй столбец первой строки в наборе целей позволяет настроить рабочий каталог, в котором выполняется компиляция и запускаются команды. Если модуль проектов включён, строка рабочего каталога также может содержать заменители для пути к основному каталогу проекта: `%B` и имя: `%b`






##### Команда сборки

Второй столбец в «не первых строках» содержит команду оболочки для запуска в рабочем каталоге. Обратите внимание на слово «оболочка». Подойдёт почти любая команда оболочки. Команда сборки может содержать заменители. `%f` для текущего файла, `%d` для каталога текущего файла и `%n` для базового имени текущего файла (имя файла без суффикса).

##### Команда запуска

Третий столбец в «не первых строках» может содержать команду оболочки для запуска в самом терминале в рабочем каталоге набора целей. Терминал открывается в виде вкладки. Модуль попытается повторно использовать вкладку терминала, если выполняется та же самая команда и предыдущее приложение завершило работу.

В верхней части вкладки Параметры цели находится панель инструментов с фильтром целей и следующими кнопками:

-  Собрать выбранную цель
- ▶ Собрать и запустить выбранную цель
-  Добавить цель
-  Создать набор целей
-  Скопировать команду или набор целей
-  Удалить текущий набор целей

#### 4.4.2.2 Вкладка «Вывод»

На вкладке Вывод показаны выходные данные консоли, сгенерированные запущенной командой (сборки). Если строка содержит расположение файла, по ней можно щёлкнуть. Если в строке вывода также показана ошибка или предупреждение, такая строка будет другого цвета.

Если на странице модуля в параметрах настройки Kate включена опция Выводить ошибки и предупреждения, на панели диагностики также будут показаны ошибки и предупреждения. Чтобы перейти к предыдущей ошибке на панели диагностики, нажмите **Alt+Shift+Стрелка влево**. Чтобы перейти к следующей ошибке, нажмите **Alt+Shift+Стрелка вправо**.

#### 4.4.3 Структура меню

Сборка → Выбрать цель...

Фильтр выбора цели фокусируется на вкладке «Параметры цели». При вводе имени будут отфильтрованы цели, которые не соответствуют введённой строке. Для перемещения по иерархическому списку целей также можно использовать клавиши со стрелками. После выбора нужной цели нажмите клавишу **Return** или **Enter** для её выполнения и запуска.

Сборка → Собрать выбранную цель

Выполняет сборку последней выбранной цели. Если ничего не выбрано, работает как Выбрать цель...

Сборка → Собрать и запустить выбранную цель

Выполняет сборку последней выбранной цели и запускает команду запуска после успешного завершения работы команды сборки.

Сборка → Скомпилировать текущий файл

Попытаться скомпилировать текущий файл путём поиска команды в возможном `compile_commands.json`.

Сборка → Завершить

Прекратить сборку цели.

Сборка → Передать фокус следующей вкладке слева

Передать фокус следующей вкладке модуля для сборки слева. Или открыть панель модуля для сборки, если она скрыта.

Сборка → Передать фокус следующей вкладке справа

Передать фокус следующей вкладке модуля для сборки справа. Или открыть панель модуля для сборки, если она скрыта.

Сборка → Загрузить цели из каталога сборки CMake

Открывает диалоговое окно со списком файлов, в котором пользователь может выбрать `CMakeCache.txt`. Когда файл выбран, модуль генерирует цели команды сборки `cmake`, которые можно выполнить в каталоге сборки для проекта на основе CMake.

#### 4.4.4 Благодарности

Автор модуля для сборки ПО Kate — Kåre Särs.

Благодарим участника «Google Code-In 2011» Salma Sultana за большой вклад в написание данного раздела.

### 4.5 Модуль «Закреть кроме/только...»

#### 4.5.1 Введение

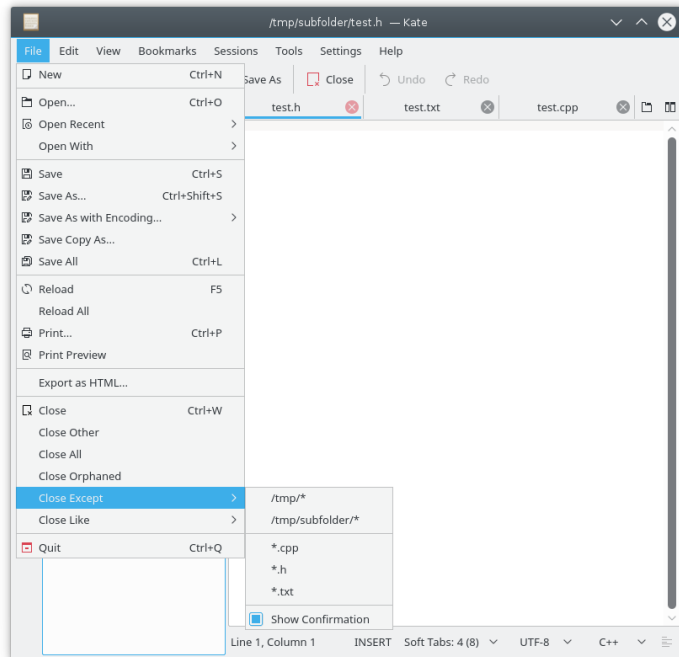
Этот модуль позволяет закрывать группу документов в зависимости от их расширения или пути.

#### 4.5.2 Работа с модулем «Закреть кроме/только...»

Допустим, в Kate открыты следующие документы:

```
/tmp/subfolder/test.h  
/tmp/test.cpp  
/tmp/test.txt
```

Тогда будут доступны следующие варианты закрытия документов, как показано на снимке ниже:



Чтобы включить или отключить диалоговое окно подтверждения, установите или снимите флажок напротив последнего элемента списка. Выбранная опция будет применена к обоим действиям.

#### 4.5.3 Структура меню

Файл → Закрывать все, кроме

Закрывает все открытые документы, кроме тех, путь или расширение которых соответствуют выбранной опции в меню.

Файл → Закрывать только

Закрывает все открытые документы, путь или расширение которых соответствуют выбранной опции в меню.

## 4.6 Модуль выбора цвета

### 4.6.1 Введение

Этот модуль добавляет встроенную панель предварительного просмотра и выбора цветов в тексте (например, #FFFFFF, white).

Чтобы подключить этот модуль, откройте диалог настройки параметров Kate (Настройка → Настроить Kate...). Затем выберите пункт Выбор цвета и закройте диалог.

### 4.6.2 Настройка

На странице параметров модуля выбора цвета в диалоге настройки Kate возможно задать указанные далее параметры поведения этого модуля.

Предварительный просмотр известных названий цветов

Следует ли показывать панель выбора цвета для известных названий цветов (например, skyblue). Список названий цветов доступен на [этой странице](#).

Поле просмотра цвета после его описания

Следует ли размещать встроенную панель предварительного просмотра после описания цвета в тексте.

Шаблон определения шестнадцатеричного кода цвета

Здесь возможно выбрать самый подходящий вариант для цветов, используемых в коде.

## 4.7 Цветные скобки

### 4.7.1 Введение

Модуль цветных скобок окрашивает пары скобок в разные цвета для облегчения восприятия текста. Тем не менее, окрашиваются не все скобки. Скобка не будет окрашена, если соответствующая ей парная открывающая или закрывающая скобка не отображается. Пара скобок, если она единственная в строке, также не будет окрашена.

### 4.7.2 Настройка

Этот модуль не предоставляет возможностей настройки.

## 4.8 Модуль CTags

### 4.8.1 Введение

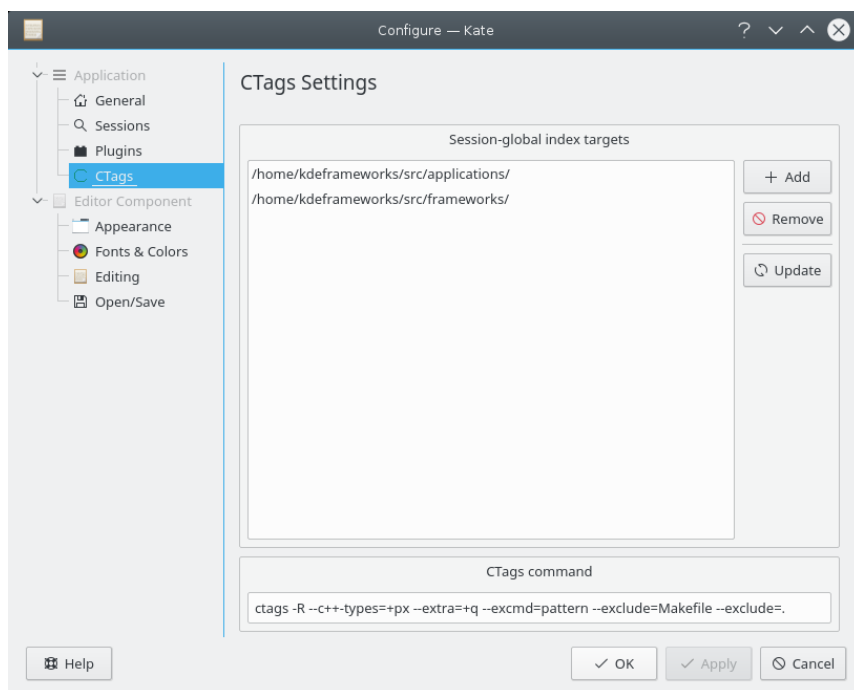
CTags создаёт файл индекса (или тегов) языковых объектов, найденных в исходных файлах, который позволяет быстро и легко находить эти элементы с помощью этого модуля в Kate. Тег указывает на языковой объект, для которого доступна (либо создана) запись индекса. Создание тегов поддерживается для следующих [языков программирования](#).

### 4.8.2 Настройка

Модуль «CTags» использует два разных файла баз данных для индексирования.

Добавить или удалить каталоги, содержащие в себе исходный код, а также обновить общую базу данных CTags возможно на странице настройки CTags в диалоге настройки Kate.

#### 4.8.2.1 Общее индексирование



#### Настройка глобальной базы данных CTags

В нижней части страницы параметров возможно указать команду CTags.

Для получения дополнительных сведений обо всех доступных параметрах ознакомьтесь с map-страницей CTags в «Центре справки KDE» (также возможно указать URL-адрес map:/ctags непосредственно в Konqueror).

Чтобы начать индексирование, нажмите кнопку Обновить. Когда индексирование завершится, закройте диалог.

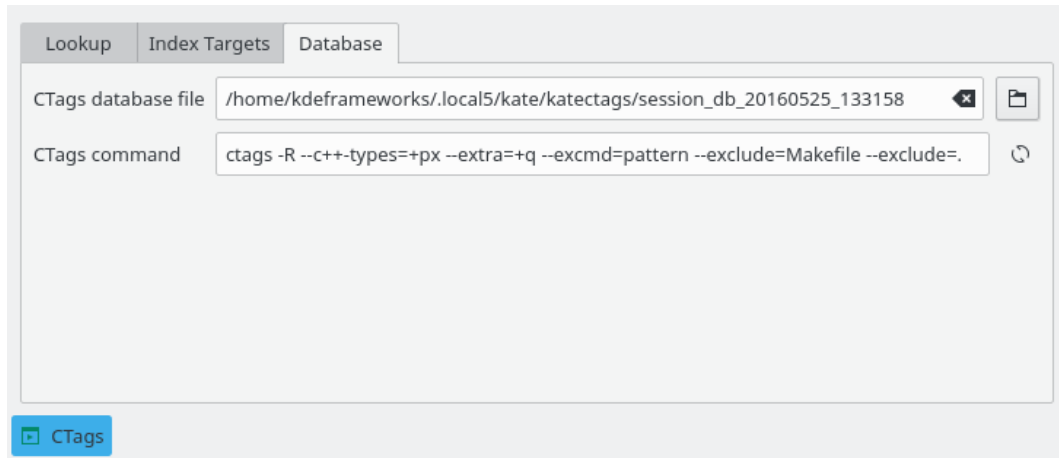
#### 4.8.2.2 Индексирование текущего сеанса

Чтобы настроить индексирование текущего сеанса, откройте панель CTags.

### Индексируемые файлы

Эта вкладка позволяет добавить или удалить каталоги, содержащие в себе исходный код, а также обновить общую базу данных CTags.

### База данных



### Настройка базы данных сеанса CTags

Выберите другой файл базы данных CTags, настройте команду CTags или верните команду, используемую по умолчанию.

## 4.8.3 Работа с модулем CTags

Чтобы перейти к строке, содержащей интересующий языковой объект (например, функцию, символ и так далее), наведите на него указатель мыши, а затем выберите одно из действий в меню CTags.

По умолчанию действия в меню CTags не имеют назначенных комбинаций клавиш. Для их настройки используйте [редактор комбинаций клавиш](#).

Либо воспользуйтесь вкладкой Поиск на панели CTags.

При вводе символов в поле поиска запустится поиск и будут показаны совпавшие имена таких языковых объектов, как функции, классы, символы и так далее, а также тип файла и его имя.

Выберите элемент в списке, чтобы перейти к соответствующей строке в исходном файле.

## 4.8.4 Структура меню

CTags → Перейти на шаг назад

Возвращает к предыдущему результату поиска в журнале поиска.

CTags → Поиск текущего текста

Открывает на панели CTags вкладку Поиск и отображает все языковые объекты, соответствующие выбранному тексту, в виде списка.

CTags → Перейти к объявлению

Если курсор находится в объекте-определении, будет открыт документ, содержащий соответствующее объявление, если необходимо, и активирован его просмотр (курсор при этом будет помещён в начало объявления).

CTags → Перейти к определению

Если курсор находится в объекте-объявлении, будет открыт документ, содержащий соответствующее определение, если необходимо, активирован его просмотр (курсор при этом будет помещён в начало определения).

## 4.9 Модуль предварительного просмотра документа

### 4.9.1 Введение

Этот модуль позволяет выполнить предварительный просмотр финального формата редактируемого документа в режиме реального времени на боковой панели. Таким образом, при редактировании, например, текста Markdown или SVG-изображения результат будет сразу же отображаться рядом с исходным текстом.

Для отображения предварительного просмотра используется модуль KParts, который в данный момент указан как предпочитаемый для типа MIME документа. Если такой модуль KParts не указан, предварительный просмотр недоступен.

Чтобы изменить предпочитаемый модуль, откройте раздел Привязки файлов в параметрах системы и измените Приоритет компонентов на вкладке Встраивание.

Тип MIME	Модуль KParts
Текст Markdown	KMarkdownWebViewPart или OkularPart
Изображение SVG	SVGPart
Файлы графического интерфейса Qt™	KUIViewerPart
Файлы графов DOT	KGraphViewerPart

Таблица 4.1: Некоторые доступные модули KParts

### 4.9.2 Структура меню

Вид → Боковые панели → Показать панель «Предварительный просмотр»

Включает или отключает отображение предварительного просмотра на боковой панели Kate.

### 4.9.3 Интерфейс

Кнопки в верхней части окна предварительного просмотра позволяют выполнять следующие действия:

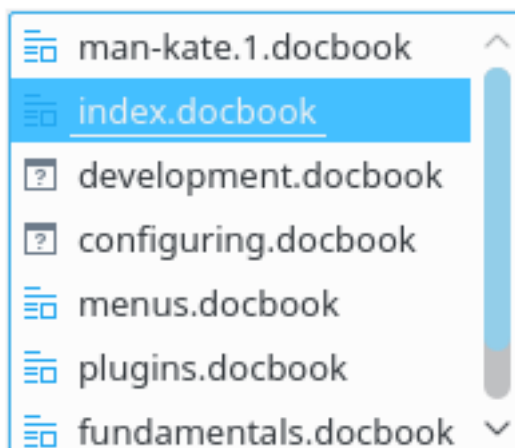
- Закрепить окно предварительного просмотра для выбранного документа. Выбор этой опции позволяет продолжить предварительный просмотр этого документа даже в том случае, когда пользователь перевёл фокус на другой документ внутри того же окна Kate.
- Включить или отключить автоматическое обновление предварительного просмотра содержимого текущего документа.
- Вручную обновить предварительный просмотр содержимого текущего документа.
- Раскрывающееся меню с действиями модуля KParts

## 4.10 Модуль переключателя документов

### 4.10.1 Структура меню

Вид → Последние документы (Ctrl+Tab), Вид → Последние документы (в обратном порядке) (Ctrl+Shift+Tab)

Открывает список последних открытых документов:



Чтобы двигаться по списку вперёд, удерживайте нажатой клавишу Ctrl и используйте клавишу Tab. Для движения в обратную сторону нажмите также клавишу Shift.

Чтобы двигаться по списку, удерживайте нажатой комбинацию клавиш Ctrl+Tab и используйте клавиши «Вверх», «Вниз», «Home» или «End». Нажатие клавиши с буквой позволяет выполнять циклический переход по всем пунктам в списке с соответствующей первой буквой. Если отпустить клавиши быстрого доступа, представление переключится на выбранный документ в списке.

## 4.11 Модуль обзора файловой системы

Модуль «Обзор файловой системы» — это средство просмотра папок, позволяющее открывать файлы из отображаемой папки в текущем окне.

### 4.11.1 Структура меню

Вид → Боковые панели → Показать панель «Просмотр файловой системы»

Включает или отключает отображение панели просмотра файловой системы Kate.

### 4.11.2 Интерфейс

Модуль просмотра файловой системы состоит из следующих элементов:

Панель инструментов

Панель инструментов содержит стандартные кнопки навигации:

- < Назад  
Выполняет переход в папку, предшествующую текущей в журнале смены папок. Кнопка недоступна, если в журнале переходов такая папка отсутствует.
- > Вперёд  
Выполняет переход в папку, расположенную следующей в журнале смены папок. Кнопка недоступна, если в журнале переходов такая папка отсутствует.



### Закладки

Открывает вложенное меню для операций редактирования и добавления закладок и создания папок для хранения закладок.



### Папка текущего документа

При нажатии этой кнопки в области просмотра папок будет показана папка, в которой содержится текущий активный документ, если это возможно. Эта кнопка недоступна, если активный документ является новым, несохранённым файлом или папка, в которой он находится, не может быть определена.



### Параметры

#### Список

Показывает только имена файлов.

#### Таблица

Показывает Имя, Дату изменения и Размер файлов.

#### Разворачиваемый список

Аналог «Списка», но папки возможно развернуть для просмотра содержимого.

#### Разворачиваемая таблица

Этот режим также позволяет разворачивать папки, но при этом показываются дополнительные столбцы, доступные в режиме «Таблица».

#### Показывать скрытые файлы

Показывает файлы, которые обычно скрыты операционной системой.

#### Автоматически синхронизировать с текущим документом

Когда эта опция включена, модуль будет автоматически переходить в папку открытого на данный момент документа каждый раз, когда будут производиться изменения.

### Расположение (строка пути)

Отображает путь к текущей открытой папке, аналогично Dolphin. Чтобы перейти в любую папку, просто щёлкните по её названию. Также возможно нажать кнопку со стрелкой рядом с названием папки, чтобы открыть список подкаталогов этой папки. Возможно также выбрать одну из точек входа, щёлкнув по крайнему левому значку строки пути (этот значок отображает текущую точку входа).

Щелчок справа от текста в строке пути вызовет поле ввода, где возможно ввести путь к папке. В строке адреса предусмотрены подсказки в виде списка ранее введённых адресов. Чтобы выбрать один из этих адресов, нажмите кнопку со стрелкой, расположенную справа от текста.

#### Подсказка

В строке адреса предусмотрено автоматическое дополнение названий папок. Способ дополнения возможно выбрать с помощью контекстного меню текстовой записи.

### Область просмотра папок

Стандартный просмотр папок KDE.

### Поле фильтра

Поле фильтра предоставляет возможность выбрать типы файлов для показа в области просмотра папок. Фильтр использует стандартные маски, которые разделяются пробелами. Возможно ввести, например, \*.crr \*.h \*.moc

Для просмотра всех файлов введите один символ звёздочки (\*).

Фильтр сохраняет последние 10 значений. Для использования одного из сохранённых значений нажмите кнопку со стрелкой, расположенную справа от текста фильтра, и выберите необходимое значение фильтра. Отключить фильтр возможно нажатием кнопки Очистить текст, расположенной слева от стрелки открытия раскрывающегося списка.

### 4.11.3 Настройка

Для настройки модуля перейдите на страницу Просмотр файловой системы в [диалоге настройки Kate](#).

#### Панель инструментов







Для настройки кнопок на панели инструментов модуля переместите нужные функции в список Выбранные действия и упорядочите их с помощью кнопок со стрелками, расположенных сбоку от списка.

## 4.12 Модуль списка документов

### 4.12.1 Введение

Список документов отображает список всех документов, открытых на текущий момент в Kate. У изменённых файлов появляется маленькое изображение дискеты (слева от заголовка).

В верхней части списка документов находится панель инструментов со следующими кнопками:

-  Создать новый документ
-  Открыть существующий документ
-  Предыдущий документ
-  Следующий документ
-  Сохранить текущий документ
-  Сохранить текущий элемент под новым именем

По умолчанию список документов представлен в режиме дерева, которое отображает структуру папок всех открытых в данный момент документов. Также доступен режим списка, отображающий все документы простым списком. Чтобы переключить режим, щёлкните правой кнопкой мыши по списку и выберите нужный вариант в меню Режим просмотра.

Если в режиме списка открыть два или более файлов с одинаковыми именами (расположенных в разных папках), к имени второго файла добавится «(2)» и так далее. Во всплывающей подсказке к этому файлу будет показано его полное имя, включая путь к нему, что позволяет легко выбрать нужный файл.

Чтобы отобразить документ в текущей активной створке, щёлкните по названию документа в списке.

В контекстном меню присутствуют некоторые стандартные действия из меню Файл.

Здесь также присутствуют действия файлового менеджера: переименовать или удалить файл. Чтобы скопировать в буфер обмена полный путь к файлу, нажмите Копировать расположение.

Чтобы упорядочить список, щёлкните по нему правой кнопкой мыши и выберите опцию Сортировать. Доступны следующие варианты:

По имени документа

Упорядочивает документы по именам в алфавитном порядке.

По пути к документу

Упорядочивает документы по пути к ним в алфавитном порядке.

По порядку открытия

Упорядочивает документы по порядку их открытия.

По умолчанию список документов визуализирует журнал, выделяя названия последних документов фоновым цветом. Если документ был отредактирован, добавляется дополнительный цвет. Самый последний документ сделан самым заметным, что позволяет быстро найти его. Эту функцию возможно отключить на странице [Документы](#) диалога настройки.

По умолчанию список документов расположен слева от области редактирования.

### 4.12.2 Структура меню

Вид → Предыдущий документ (Alt+Вверх)

Открывает документ, отображаемый над текущим открытым документом в списке «Документы».

Вид → Следующий документ (Alt+Вниз)

Открывает документ, отображаемый под текущим открытым документом в списке «Документы».

Вид → Показать активный документ

Показывает текущий открытый документ в списке «Документы».

### 4.12.3 Настройка

Подсветка фона

В этом разделе возможно включить или отключить затенение фона последних документов, а также выбрать, какие цвета использовать, если оно включено.

Сортировать по

Устанавливает принцип упорядочения документов. Этот параметр также возможно указать с помощью щелчка правой кнопкой мыши по списку документов.

Режим просмотра

Доступны две опции, влияющие на отображение панели «Документы». Опция *Дерево документов* отобразит иерархическую структуру документов, а опция *Список документов* отобразит документы простым списком.

Показывать полный путь

Если включён режим просмотра «Дерево документов», записи папок на панели документов будут содержать не только название папки, но и полный путь к ней в файловой системе. Эти данные не отображаются в режиме просмотра «Список документов».

Показывать панель инструментов

Если включён режим просмотра «Дерево документов» и этот параметр, над списком документов отображается панель инструментов (она содержит, например, кнопку *Сохранить*). Снимите флажок, если эту панель инструментов следует скрыть.

Показывать кнопку закрытия

Если этот параметр включён, Kate при наведении указателя на открытые документы будет отображать кнопку закрытия.

## 4.13 Модуль GDB

### 4.13.1 Введение

Модуль GDB Kate предоставляет простой интерфейс к любому отладчику, который поддерживает [протокол DAP \(Debugger Adapter Protocol\)](#). В частности, этот модуль обеспечивает работу с отладчиком GDB (GNU Project Debugger); подробное описание доступно [здесь](#).

#### Важно

Перед началом работы с модулем настоятельно рекомендуется ознакомиться с GDB. Более подробные сведения об использовании GDB доступны на [веб-сайте GDB](#).

Модуль GDB подключается в разделе [«Модули»](#) диалога настройки Kate.

#### Подсказка

При компиляции с помощью gcc/g++ рекомендуется использовать аргумент командной строки -ggdb.

Когда всё будет подготовлено, откройте исходный файл в Kate, выберите «профиль отладчика», введите путь к исполняемому файлу на вкладке Параметры панели Отладка и для запуска отладки выберите в меню Отладка → Начать отладку.

«Профиль отладчика» позволяет указать сервер DAP, который будет использоваться (например, GDB), и способ запуска этого сервера. Как правило, сервер запускает процесс в соответствии с приведённым выше описанием, но также возможно подключить сервер к уже запущенному процессу (в этом случае потребуется указать идентификатор процесса, а не исполняемый файл). Могут быть доступны и другие режимы, специфичные для языка и сервера DAP. Дополнительные сведения об этом и соответствующие подробности настройки приводятся далее.

### 4.13.2 Структура меню и панели инструментов

Все эти функции доступны в меню Kate, и большинство из них доступно также на панели «Отладка».

Вид → Боковые панели → Показать панель «Отладка»

Отображает панель, содержащую вывод GDB, используемую командную строку GDB и другие параметры.

Вид → Боковые панели → Показать панель «Лок. переменные и стек»

Показывает список всех загруженных в данный момент переменных и их значений и обратную трассировку GDB.

Отладка → Цели

Открывает вложенное меню, состоящее из списка целей (исполняемых программ).

Отладка → Начать отладку

Запускает GDB с заданной целью.

Отладка → Завершить отладку

Останавливает GDB.

Отладка → Начать отладку заново

Перезапускает GDB.

## Руководство пользователя Kate

Отладка → Поставить точку останова / Прервать

Устанавливает точку останова в текущей позиции курсора.

Отладка → Войти

Выполняет текущую инструкцию (вызов функции будет отлажен).

Отладка → Перейти к следующей строке

Выполняет текущую инструкцию (вызов функции не будет отлажен).

Отладка → Выйти

Продолжает выполнение, пока выполняемая программа не завершится.

Отладка → Перейти в позицию курсора

Перемещает программный счётчик (следующее выполнение).

Отладка → Выполнить до курсора

Выполняет программу, пока она не дойдёт до текущей позиции курсора.

Отладка → Продолжить

Игнорирует точки останова и выполняет программу, пока она не завершится (успешно или нет).

Отладка → Показать значение

Отображает значение переменной, на которую в данный момент указывает курсор.

Настройка → Видимые панели инструментов → Модуль GDB

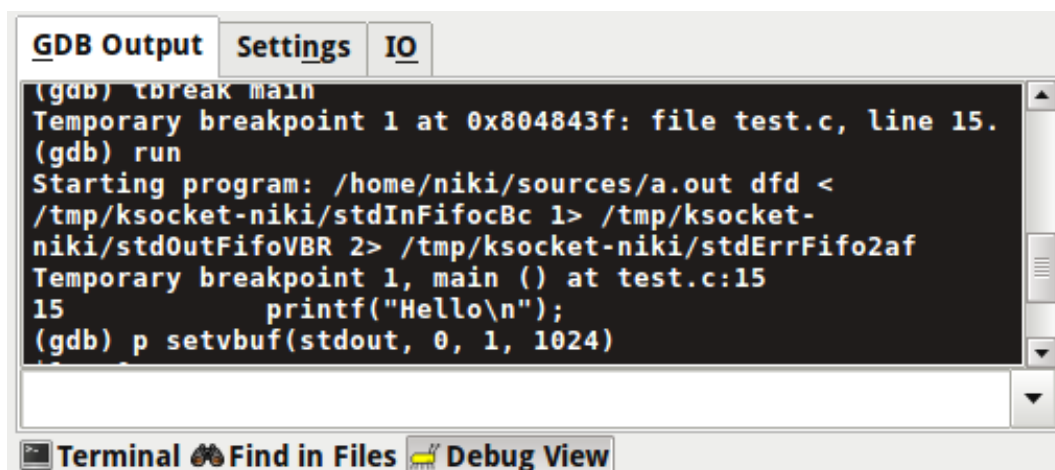
Отображает панель инструментов отладки.

### 4.13.3 Панель «Отладка»

Панель Отладка состоит из нескольких вкладок:

Вывод GDB

Содержит вывод из GDB и командную строку GDB.



Вкладка Вывод GDB, отображающая результат текущего сеанса отладки.

Параметры

Исполняемый файл

Путь к исполняемому файлу для отладки.

## Руководство пользователя Kate

### Рабочий каталог

Текущий рабочий каталог, относящийся к цели.

### Аргументы

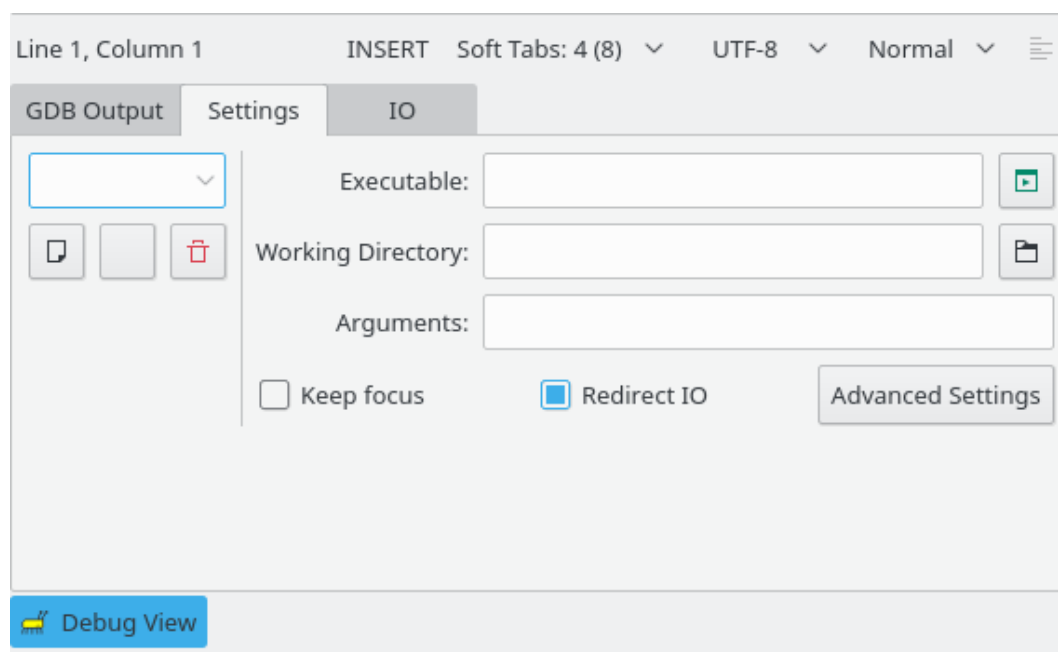
Аргументы для программы.

### Удерживать фокус

Удерживать фокус в командной строке GDB.

### Перенаправлять ввод и вывод

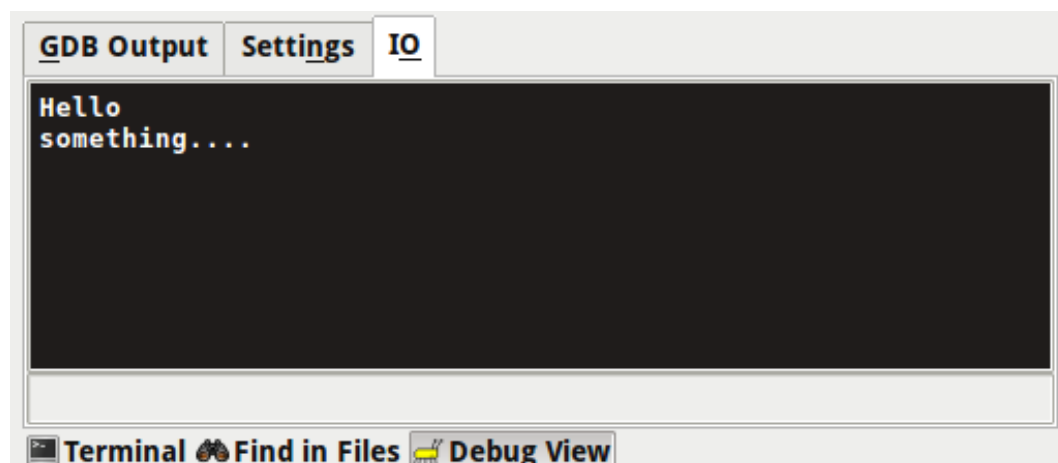
Открывает на панели Отладка новую вкладку Ввод и вывод, в которой доступны просмотр вывода выполняемой программы и возможность ввода в неё.



Вкладка Параметры, отображающая параметры текущего сеанса отладки.

### Ввод и вывод

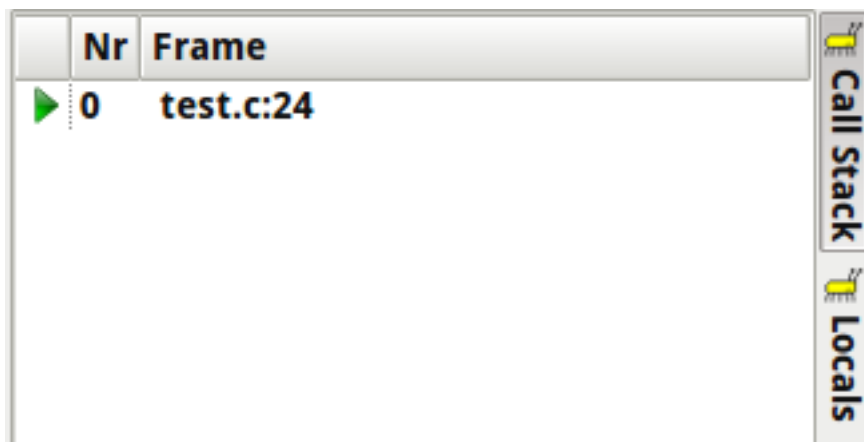
Содержит область, в которой отображается вывод из выполняемой программы, и командную строку, которая обеспечивает возможность ввода в неё.



Вкладка Ввод и вывод, отображающая вывод простой тестовой программы.

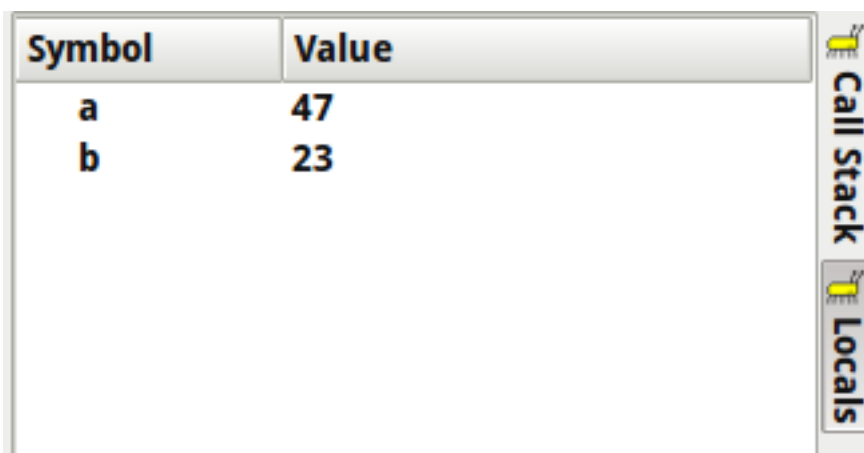
#### 4.13.4 Локальные переменные и стек

Боковая панель Стек содержит список отформатированного стека вызовов, возвращённого из GDB.



Панель Стек модуля GDB.

Боковая панель Локальные переменные состоит из списка всех загруженных в данный момент переменных из программы и соответствующих значений.



Панель Локальные переменные модуля GDB.

#### 4.13.5 Благодарности

Благодарим участника «Google Code-In 2011» Martin Gergov за большой вклад в написание данного раздела.

#### 4.13.6 Настройка

На странице настройки модуля доступны для выбора «профили отладчика». Там показана стандартная конфигурация (JSON), её можно «перезаписать» заданной пользователем в подобной форме. Пример:

```
{
  "dap": {
    "debugpy": {
      "url": "https://github.com/microsoft/debugpy",
```



ция (загруженная из файла) переопределяет (внутреннюю) конфигурацию по умолчанию и, в свою очередь, перезаписывается параметрами записи «dap» в конфигурации проекта .kateproject.

#### 4.13.6.1 Настройка среды выполнения

Подробные сведения доступны в разделе [Среда выполнения клиента LSP](#) ниже. Здесь же достаточно упомянуть о том, что может потребоваться запустить процесс для диагностики (и сервер DAP) в «специальной среде», которая может определяться просто переменными среды или же каким-либо контейнером (при наличии необходимых зависимостей и условий для корректного выполнения).

Аналогично примеру в упомянутом разделе, следующая конфигурация может быть задана в .kateproject.

```
{
  // это также может быть массив объектов
  "exec": {
    "hostname": "foobar"
    // команда также может быть массивом строк
    "prefix": "podman exec -i foobarcontainer",
    "mapRemoteRoot": true,
    "pathMappings": [
      // возможна любая из следующих форм
      // также существует более автоматизированная альтернатива, по ←
      // подробнее о ней в указанном разделе
      [ "/dir/on/host", "/mounted/in/container" ]
      { "localRoot": "/local/dir", "remoteRoot": "/remote/dir" }
    ]
  },
  "dap": {
    "debugpy": {
      "run": {
        // в этом разделе это применимо ко всем конфигурациям
        // это устанавливает соответствие или выполняет объединен ←
        // ие для объекта выше
        "exec": { "hostname": "foobar" },
        // если соединение с сервером установлено,
        // можно дополнительно указать в явном виде порт (для пуб ←
        // ликации/перенаправления)
        "port": 5678,
        // сервер также можно настроить на приём дополнительных у ←
        // злов, не только localhost
        "host": "0.0.0.0"
      }
    }
  }
}
```

Подробные сведения приводятся в упомянутом разделе, но, по сути, префикс prefix предваряет командную строку сервера DAP, указанную в другом месте. В результате сервер запускается в указанном контейнере, а потом, в свою очередь, также и запущенный процесс. pathMapping обеспечивает преобразование путей к файлам между представлением редактора и представлением сервера DAP (контейнера), например, при работе с установкой точек останова или обработке представленной обратной трассировки. Обратите внимание, что такое сопоставление опционально и может быть или не быть полезным. При работе с кодом C/C++, который компилируется в основной системе, символическая информация ссылается на исходные файлы в основной системе, которых вообще не существует в другой среде. Однако в других определённых сценариях (например, python) обстоятельство ссылка идёт на фактические файлы времени выполнения (в другой среде).

Следует помнить про указанное далее:

- Сервер DAP должен присутствовать в среде/контейнере и быть настроен для поддержки корректной работы отладчика (следовательно, если нужно, привилегированно, с необходимыми возможностями).
- Редактор и DAP должны иметь возможность обмениваться данными. Если используется контейнер, последний должен либо использовать соединение с сетью основной системы, либо предоставлять подходящий связанный или опубликованный порт вместе с соответствующим фрагментом конфигурации — как в примере выше (так как автоматически выбранный при порте 0 не будет работать).
- Указанный исполняемый файл или идентификатор процесса должен быть в перспективе «контейнера», как и любые аргументы (отлаживаемый исполняемый файл).

Кроме того, как и в случае с LSP, некоторые переменные среды уже заданы; `KATE_EXEC_PLUGIN` установлена в значение `dap`, `KATE_EXEC_SERVER` — в значение типа отладчика или языка (например, `python`), а `KATE_EXEC_PROFILE` — в значение записи конфигурации (например, `launch`).

## 4.14 Модуль проектов

### 4.14.1 Введение

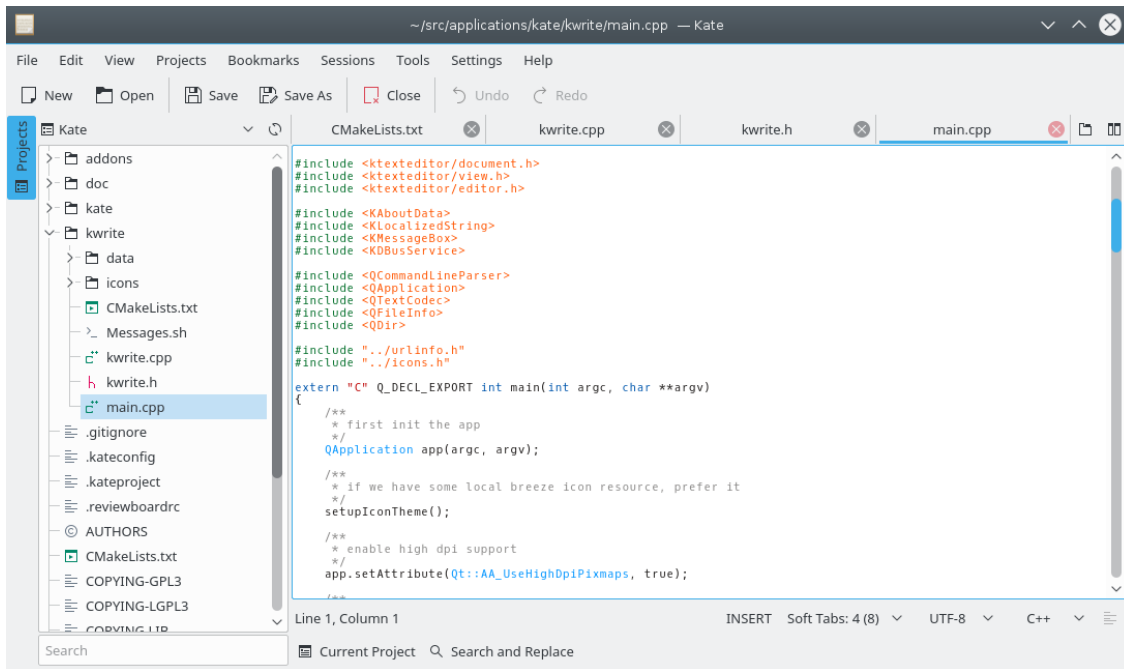
Принцип работы модуля проектов — предоставлять структурированный список файлов, относящихся к проекту, со следующими свойствами:

1. Обеспечение структурированного просмотра файлов
2. Упрощение и ускорение открытия и переключения проектов
3. Поддержка поиска и замены в проекте
4. Обеспечение функции простого автодополнения
5. Упрощение быстрого открытия файлов в проекте
6. Поддержка сборки проектов

### 4.14.2 Структурированный просмотр файлов

Подключив модуль проекта на странице параметров Kate, откройте файл в проекте, чтобы появилась боковая панель со списком всех проектов и их отдельных файлов, как показано на снимке ниже:

## Руководство пользователя Kate



Как это видно на снимке, текущий активный проект — это «Kate», и его содержимое отображено в качестве разворачиваемого списка. Чтобы открыть тот или иной файл в редакторе, просто щёлкните по нему левой кнопкой мыши. Также предусмотрено контекстное меню, с помощью которого возможно открывать файлы в других приложениях, например файл с расширением `.ui` — в Qt Designer.

Чтобы активировать фильтр, начните вводить название файла в поле поиска, которое расположено под списком.

### 4.14.3 Переключение между проектами

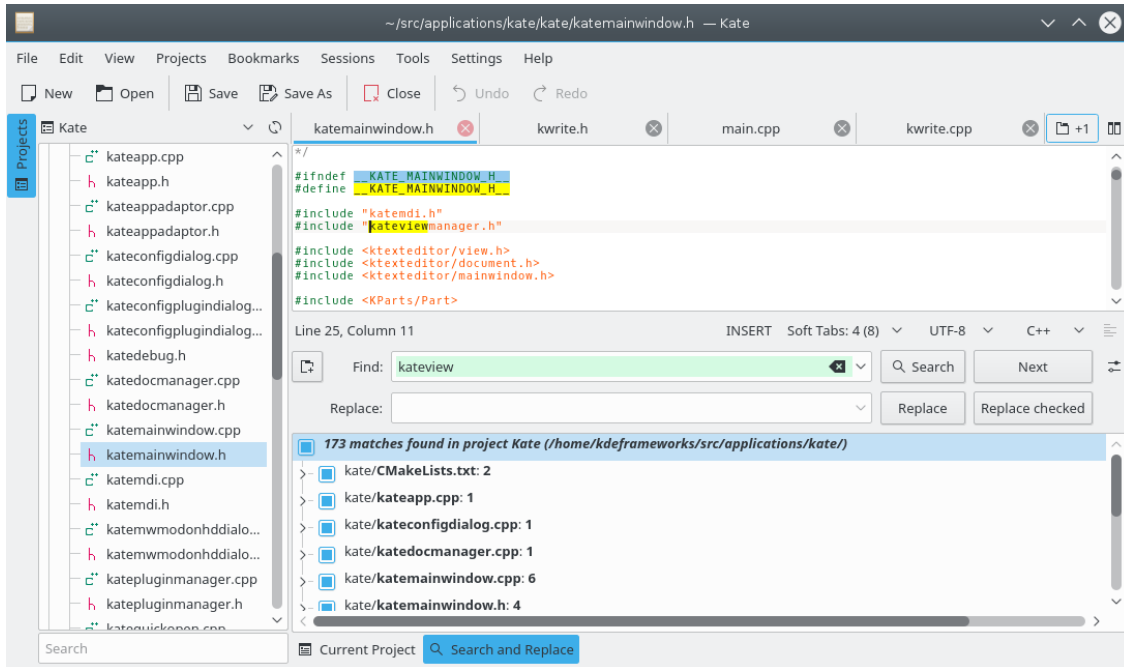
Основная идея — избавить пользователя от необходимости открывать проект вручную, такая функция даже не предусмотрена. При открытии файла модуль проектов быстро сканирует папки на предмет наличия файла `.kateproject`. Если такой файл найден, проект будет загружен автоматически.

Более того, если открыть в Kate другой документ, относящийся к другому проекту, модуль автоматически сделает этот проект текущим, поэтому активным всегда будет нужный проект. Конечно, также поддерживается переключение между проектами с помощью поля со списком.

### 4.14.4 Поиск и замена в проектах

В редактор Kate встроен модуль «Поиск и замена», отображающийся на боковой панели, которая расположена в нижней части окна. Загрузив проект, откройте боковую панель модуля поиска и замены и переключите режим на поиск и замену в текущем проекте:

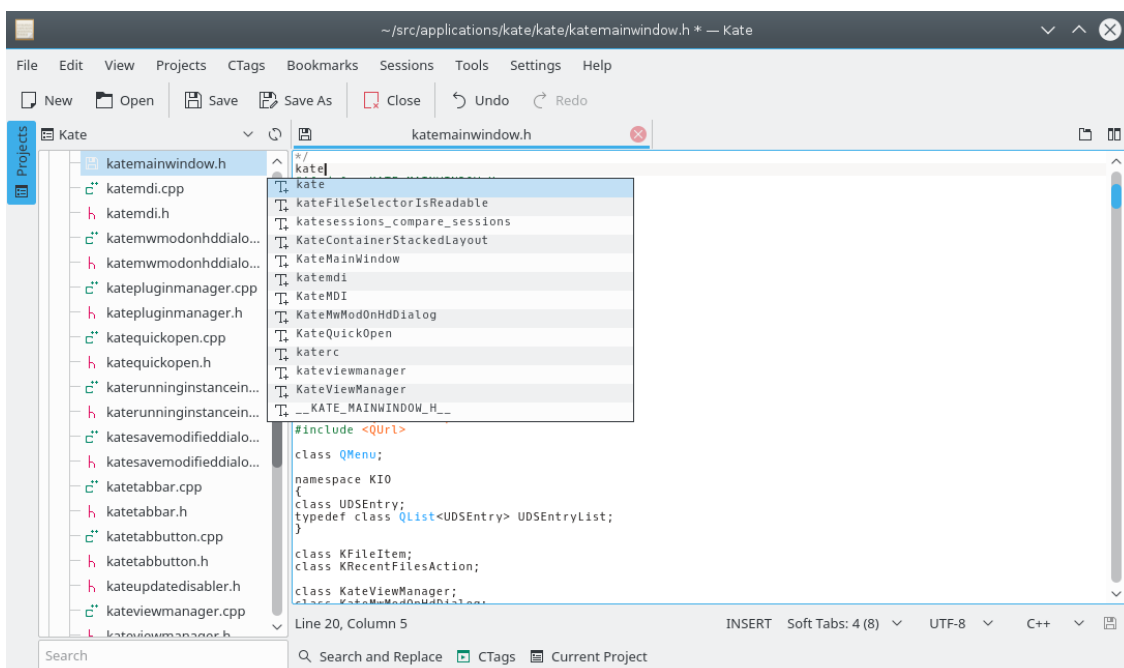
## Руководство пользователя Kate



### 4.14.5 Простое автодополнение

Модуль предоставляет простые средства автодополнения на основе CTags, учитывая данные относящихся к проекту файлов. При первоначальном открытии проекта CTags анализирует все файлы проекта в фоновом потоке и сохраняет информацию CTags в `/tmp`. Этот файл затем используется для всплывающего окна автодополнения в Kate.

Для сравнения: без такого автодополнения Kate поддерживает показ вариантов автодополнения только для тех слов, которые используются в текущем файле. Поэтому автодополнение в модуле проектов намного эффективнее.



Если отсутствует CTags, пассивное всплывающее окно предупреждает об этой проблеме.

Также следует отметить, что файл CTags в /tmp очищается при выходе из Kate, поэтому модуль не засоряет папки ненужными файлами.

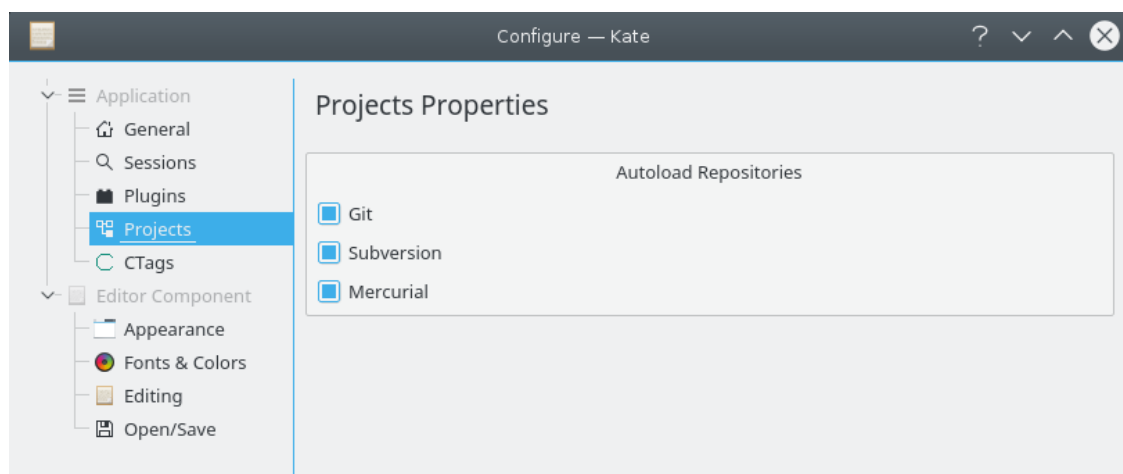
#### 4.14.6 Поддержка сборки проекта

Ещё одна особенность — это поддержка модуля для сборки ПО, что обеспечивает автоматическую корректную настройку.

#### 4.14.7 Создание проектов

##### 4.14.7.1 Автоматическая загрузка проектов

В модуле проектов предусмотрена функция автозагрузки. Список файлов доступен в системе управления версиями. Для этого в параметрах настройки должна быть включена автоматическая загрузка репозитория для соответствующей системы управления версиями (по умолчанию — включена):



##### 4.14.7.2 Создание проектов вручную

Достаточно просто создать файл .kateproject в корневой папке проекта. Например, для проекта «Kate» файл .kateproject будет выглядеть следующим образом:

```
{
  "name": "Kate",
  "files": [
    {
      "git": 1
    }
  ]
}
```

Содержимое файла написано с помощью синтаксиса JSON. Название проекта — «Kate», и файлы, содержащиеся в нём, должны быть прочитаны из Git.

Вместо git также допустимо использовать subversion (svn) или mercurial (hg). Если выполнять чтение файлов из системы управления версиями не требуется, можно просто вызвать Kate из командной строки:

```
kate /path/to/folder
```

или следующим образом указать программе рекурсивно загружать файлы из каталогов:

```
{
  "name": "Kate",
  "files": [
    {
      "directory": "kate",
      "filters": [
        "*.cpp",
        "*.h",
        "*.ui",
        "CMakeLists.txt",
        "Find*.cmake"
      ],
      "recursive": 1,
      "hidden": 1
    }
  ],
  "exclude_patterns": [
    "^build/.*"
  ]
}
```

Здесь вложенные папки и фильтры определяют, что является частью проекта. Также можно совместить управление версиями и управление файлами на основе фильтров. Получение скрытых файлов не будет выполняться, если опция

"hidden" установлена в значение «0».

"exclude\_patterns" — список шаблонов регулярных выражений, которые можно использовать для исключения папок и файлов из иерархии проекта. В рассматриваемом примере будет исключены все файлы и папки в каталоге build корневого каталога.

Если требуется добавить поддержку сборки, можно создать такой файл .kateproject:

```
{
  "name": "Kate",
  "files": [
    {
      "git": 1
    }
  ],
  "build": {
    "directory": "build",
    "build": "make all",
    "clean": "make clean",
    "install": "make install",
    "targets": [
      {
        "name": "all",
        "build_cmd": "ninja",
        "run_cmd": "./bin/kate"
      },
      {
        "name": "kate",
        "build_cmd": "ninja kate-bin"
      }
    ]
  }
}
```

Указанные выше цели затем появятся в [Модуле сборки](#) в разделе «Цели, созданные модулем». Если указан массив

“targets” , то

“build” ,

“clean” и

“install” игнорируются. Каждый элемент массива указывает цель.

“name” представляет собой имя цели,

“build\_cmd” используется для сборки цели, а

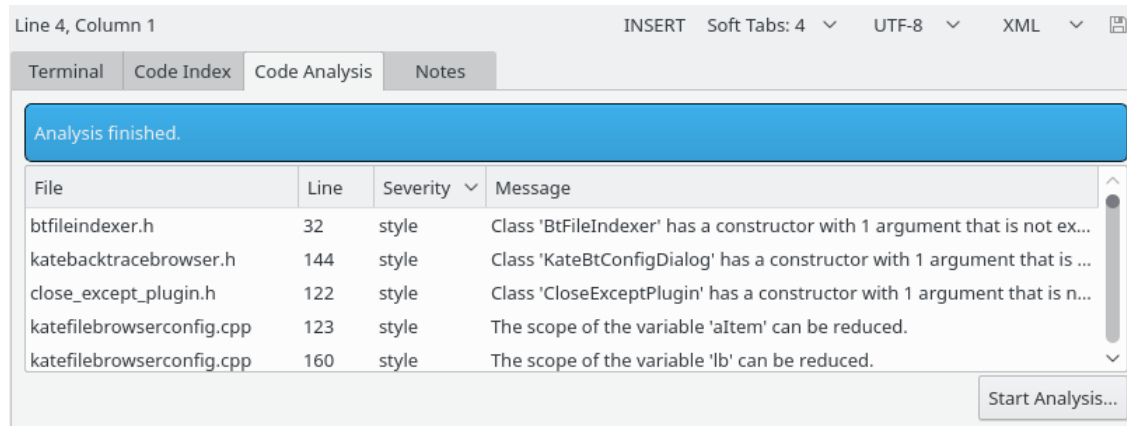
“run\_cmd” — для запуска цели. Важнее всего

“directory” — это место выполнения команд.

Если файл .kateproject отслеживается системой управления версиями, но возникла необходимость изменить конфигурацию для определённого рабочего места, необходимые изменения возможно сохранить в отдельном файле с именем .kateproject.local. Содержимое этого файла будет приоритетнее содержимого файла .kateproject.

#### 4.14.8 Текущий проект

С помощью [Проекты](#) → [Перейти...](#) (Alt+1) возможно открыть в нижней части редактора панель Текущий проект с четырьмя вкладками:



Панель терминала

[Эмулятор терминала](#), запускающийся в корневой папке проекта.

Индекс кода

При вводе символов в панель поиска будет запущен поиск и показаны имена функций, классов, символов и так далее, а также тип файла и номер строки.

Выберите элемент в списке, чтобы перейти к соответствующей строке в исходном файле.

Анализ кода

Нажмите кнопку [Начать анализ...](#) для запуска статического анализа кода C и C++ с помощью cppcheck и формирования отчёта с отображением имени файла, номера строки, степени серьёзности (стиль, предупреждение и так далее) и самой найденной ошибки.

Выберите элемент в списке, чтобы перейти к соответствующей строке в исходном файле.

Примечания

Текст, введённый в этой вкладке, будет сохранён в файле .kateproject.notes.

#### 4.14.9 Меню «Проекты»

Меню Проекты позволяет переключаться между текущими открытыми проектами. Отображается модулем проектов.

Проекты → Назад (Ctrl+Alt+Стрелка влево)

Перейти к предыдущему проекту.

Проекты → Вперёд (Ctrl+Alt+Стрелка вправо)

Перейти к следующему проекту.

Проекты → Перейти... (Alt+1)

Открывает область Текущий проект в нижней части окна редактора.

### 4.15 Модуль «Клиент LSP»

Модуль предоставляет множество языковых функций, таких как дополнение кода, навигация по коду или поиск ссылок на основе [протокола языкового сервера](#).

Как только модуль будет активирован на странице модулей, в диалоге настройки Kate появится новый раздел под названием «Клиент LSP».

#### 4.15.1 Структура меню

Далее, в описании пунктов меню, также приводятся соответствующие команды (если таковые имеются). В документации этих команд доступны дополнительная информация и интерпретация (они могут варьироваться в зависимости от используемого языка программирования). Фраза «текущий символ» относится к символу, соответствующему текущей позиции курсора, как это определено языком и реализацией сервера.

Клиент LSP → Перейти к определению

[textDocument/definition] Перейти к определению текущего символа.

Клиент LSP → Перейти к объявлению

[textDocument/declaration] Перейти к объявлению текущего символа.

Клиент LSP → Перейти к определению типа

[textDocument/typeDefinition] Перейти к определению типа текущего символа.

Клиент LSP → Найти ссылки

[textDocument/references] Найти ссылки на текущий символ.

Клиент LSP → Найти реализации

[textDocument/implementation] Найти реализации текущего символа.

Клиент LSP → Подсветить

[textDocument/documentHighlight] Подсветить ссылки на текущий символ в текущем документе.

Клиент LSP → Наведение

[textDocument/hover] Информация о текущем символе при наведении.

Клиент LSP → Форматировать

[textDocument/formatting] [textDocument/rangeFormatting] Форматировать текущий документ или текущий выделенный фрагмент.

Клиент LSP → Переименовать

[textDocument/rename] Переименовать текущий символ.

Клиент LSP → Быстрое исправление

[textDocument/codeAction, workspace/executeCommand] Рассчитывает и применяет быстрое исправление для диагностики в текущей позиции (или строке).

Клиент LSP → Показывать документацию в списке автодополнения

Показывать документацию по выделенному объекту в списке автодополнения.

Клиент LSP → Включить справку сигнатуры при автодополнении

Также показывать справку сигнатуры в списке автодополнения.

Клиент LSP → Включать объявление в ссылках

Запрашивать включение объявления символа при запросе ссылок.

Клиент LSP → Добавлять скобки при дополнении функций

Автоматически добавлять пару скобок после автоматического дополнения названия функции.

Клиент LSP → Показывать информацию при наведении

Показывать информацию при наведении (указателя мыши). Независимо от значения этого параметра, запрос этих данных всегда возможно инициировать вручную.

Клиент LSP → Форматировать при наборе

[document/onTypeFormatting] Форматировать части документа при наборе определённых символов пользователем. Например, при наборе символа переноса строки возможно создать отступ или выполнить любое другое действие, определяемое сервером LSP. Сценарии расстановки отступов в редакторе могут пытаться сделать то же самое (в зависимости от режима расстановки), поэтому не рекомендуется включать оба соответствующих параметра одновременно.

Клиент LSP → Добавочная синхронизация документа

Отправлять для обработки сервером отдельные части документа во время редактирования, а не весь документ (если на сервере предусмотрена поддержка такого режима обработки).

Клиент LSP → Подсвечивать место перехода

Предоставить временный визуальный сигнал после выполнения перехода (к определению, объявлению и так далее).

Клиент LSP → Показывать уведомления о диагностике

[textDocument/publishDiagnostics] Обработать и показывать уведомления о диагностике, отправляемые сервером.

Клиент LSP → Показывать подсветку диагностики

Добавить подсветку текста для диапазонов, указанных в диагностике.

Клиент LSP → Показывать отметки диагностики

Отмечать в документе строки, указанные в диагностике.

Клиент LSP → Перейти на вкладку диагностики

Перейти на вкладку диагностики в панели модуля.

Клиент LSP → Закрыть все не относящиеся к диагностике вкладки

Закрыть все вкладки, не относящиеся к диагностике (например, ссылки), на панели модуля.

Клиент LSP → Перезапустить сервер LSP

Перезапустить сервер LSP текущего документа.

Клиент LSP → Перезапустить все серверы LSP

Остановить все серверы LSP и затем перезапустить их.

## 4.15.2 Поддержка перехода к идентификаторам

Клиент LSP Client позволяет перейти к любому идентификатору в проекте или текущем файле. Чтобы перейти к какому-либо идентификатору в файле, воспользуйтесь панелью «Клиент LSP: структура идентификаторов», расположенной в правой части окна Kate. В этой панели приводится список всех идентификаторов, обнаруженных сервером в текущем документе.

### 4.15.2.1 Настройка структуры идентификаторов клиента LSP

По умолчанию все идентификаторы упорядочены по порядку в документе, но также возможно выполнить сортировку по алфавиту. Для этого следует щёлкнуть по панели правой кнопкой мыши и выбрать пункт «Сортировать по алфавиту».

По умолчанию панель отображает идентификаторы в виде дерева, но с помощью контекстного меню также возможно представить их в виде списка.

### 4.15.2.2 Поддержка глобального перехода к идентификаторам

Чтобы перейти к любому идентификатору в проекте, следует открыть диалог перехода с помощью комбинации клавиш `Ctrl+Alt+p`. При открытии диалог пуст, но по мере набора текста пользователем будут появляться соответствующие идентификаторы. Качество совпадений, а также возможности фильтрации зависят от используемого сервера. Например, `clangd` поддерживает неточную фильтрацию, а какой-либо другой сервер может не поддерживать эту возможность.

## 4.15.3 Другие возможности

Поддерживается команда переключения заголовка источника `clangd`. Чтобы переключить заголовок источника в проекте C или C++ следует либо выбрать пункт контекстного меню «Переключить заголовок источника», либо нажать клавишу `F12`.

Для быстрого перехода к идентификатору возможно навести указатель мыши на символ и затем использовать комбинацию `Ctrl +` левая кнопка мыши.

## 4.15.4 Настройка

На странице настройки модуля возможно установить стандартные значения некоторых перечисленных выше элементов меню. На этой странице также имеется поле, в котором возможно указать путь к файлу конфигурации сервера. Это файл JSON, который позволяет указать сервер LSP для запуска (и последующего обмена данными с помощью `stdin/stdout`). Для удобства включена конфигурация по умолчанию (она доступна на странице настройки модуля). Чтобы сделать дальнейшие пояснения доступнее, здесь приводится фрагмент этой конфигурации:

```
{
  "servers": {
    "bibtex": {
      "use": "latex",
      "highlightingModeRegex": "^BibTeX$"
    },
    "c": {
      "command": ["clangd", "-log=error", "--background-index"],
      "commandDebug": ["clangd", "-log=verbose", "--background-↔
index"],
      "url": "https://clang.llvm.org/extra/clangd/",
      "highlightingModeRegex": "^(C|ANSI C89|Objective-C)$"
    },
    "cpp": {
```

```

    "use": "c",
    "highlightingModeRegex": "^(C\\+\\+|ISO C\\+\\+|Objective-C ↵
        \\+\\+)$"
  },
  "d": {
    "command": ["dls", "--stdio"],
    "url": "https://github.com/d-language-server/dls",
    "highlightingModeRegex": "^D$"
  },
  "fortran": {
    "command": ["fortls"],
    "rootIndicationFileNames": [".fortls"],
    "url": "https://github.com/hansec/fortran-language-server",
    "highlightingModeRegex": "^Fortran.*$"
  },
  "javascript": {
    "command": ["typescript-language-server", "--stdio"],
    "rootIndicationFileNames": ["package.json", "package-lock. ↵
        json"],
    "url": "https://github.com/theia-ide/typescript-language- ↵
        server",
    "highlightingModeRegex": "^JavaScript.*$",
    "documentLanguageId": false
  },
  "latex": {
    "command": ["texlab"],
    "url": "https://texlab.netlify.com/",
    "highlightingModeRegex": "^LaTeX$"
  },
  "go": {
    "command": ["go-langserver"],
    "commandDebug": ["go-langserver", "-trace"],
    "url": "https://github.com/sourcegraph/go-langserver",
    "highlightingModeRegex": "^Go$"
  },
  "python": {
    "command": ["python3", "-m", "pyls", "--check-parent-process ↵
        "],
    "url": "https://github.com/palantir/python-language-server",
    "highlightingModeRegex": "^Python$"
  },
  "rust": {
    "command": ["rls"],
    "path": ["%{ENV:HOME}/.cargo/bin", "%{ENV:USERPROFILE}/.cargo ↵
        /bin"],
    "rootIndicationFileNames": ["Cargo.lock", "Cargo.toml"],
    "url": "https://github.com/rust-lang/rls",
    "highlightingModeRegex": "^Rust$"
  },
  "ocaml": {
    "command": ["ocamlmerlin-lsp"],
    "url": "https://github.com/ocaml/merlin",
    "highlightingModeRegex": "^Objective Caml.*$"
  }
}

```

Обратите внимание, что каждая запись «command» может быть массивом или строкой (в

этом случае она будет поделена на записи массива). Кроме того, также учитывается запись верхнего уровня «global» (рядом с записью «server»); подробнее об этом далее. Поиск указанного исполняемого файла выполняется обычным способом, то есть с помощью переменной PATH. Если исполняемый файл установлен в каком-либо нестандартном расположении, может потребоваться развернуть соответствующий путь. В расположении, которое хранится в обычной переменной PATH, можно воспользоваться (символической) ссылкой или сценарием-обёрткой. Как показано выше, также можно указать «path» (расположение, в котором будет выполнен поиск после стандартных мест).

Во всех записях «command», «root» и «path» будет выполняться расширение переменных.

Инструкцией «highlightingModeRegex» можно воспользоваться для привязки режима подсветки, который используется Kate, к идентификатору языка сервера. Если регулярное выражение не указано, используется сам идентификатор языка. Если запись «documentLanguageId» установлена в значение «false», идентификатор языка не предоставляется серверу при открытии документа. Это может подойти для серверов, которые определяют тип документа точнее, чем способ на основе режима Kate.

Кроме того, каждый объект записи сервера может также иметь запись «initializationOptions», данные которой передаются серверу как часть метода «initialize». Если такая запись присутствует, данные записи «settings» будут переданы серверу с помощью уведомления «workspace/didChangeConfiguration». Как «completionTriggerCharacters», так и «signatureTriggerCharacters» можно указать как объект JSON со строковыми участниками «exclude» и/или «include». Они будут использоваться, соответственно, для исключения или добавления символов в соответствующий набор переключателей, который предоставляется сервером.

Применяются различные этапы переопределения и слияния.

- Пользовательская конфигурация (загруженная из файла) переопределяет (внутреннюю) конфигурацию по умолчанию
- Запись «lspclient» в конфигурации проекта .kateproject переопределяет предыдущие записи
- Полученная в результате обработки запись «global» используется как дополнение (не переопределение) любой из записей сервера

Для каждой комбинации (root, servertype) используется один экземпляр сервера. Если значение «root» указано как абсолютный путь, оно используется без обработки. В противном случае оно считается указанным относительно «projectBase» (как определено в [модуле проектов](#)), если это применимо, или иным образом связанным с каталогом документа. Если значение не указано и «rootIndicationFileNames» представляет собой массив имён файлов, то выбирается родительский каталог текущего документа, содержащий такой файл. Альтернативный вариант (если не указано значение «root» и «rootIndicationFilePatterns» представляет собой массив шаблонов имён файлов) — выбирается родительский каталог текущего документа, соответствующий шаблону имени файла. Последний резервный вариант для «root» — домашний каталог пользователя. Полученное в результате такой обработки значение «root» определяет, требуется ли для документа отдельный экземпляр сервера. Если такой экземпляр нужен, значение «root» будет передано как rootUri/rootPath.

Рекомендуется не указывать значение root, так как оно не так важно для сервера (хотя ситуации могут быть разными). Если экземпляров сервера будет немного, они будут работать эффективнее и обеспечивать «более широкое» поле зрения, чем в случае использования многих отдельных экземпляров сервера.

Как упоминалось ранее, в некоторых записях выполняется расширение переменных. Использование этой возможности в сочетании со «сценарием-обёрткой» позволяет гибко подстроиться к самым разным обстоятельствам. Например, рассмотрим сценарий разработки Python, состоящий из нескольких проектов (например, репозитория git), каждый из которых имеет свою виртуальную среду. При использовании стандартной конфигурации у сервера языка Python не будет информации о виртуальной среде. Но это можно исправить следующим образом. Сначала следует ввести следующий текст в поле «Параметры сервера пользователя» модуля клиента LSP:

```
{
  "servers":
  {
    "python":
    {
      "command": ["pylsp_in_env", "%{Project:NativePath} ←
                  }"],
      "root": "."
    }
  }
}
```

Корневая запись выше относится к каталогу проекта и обеспечивает запуск отдельного языкового сервера для каждого проекта (что необходимо в этом случае, так как каждый проект имеет собственную уникальную виртуальную среду).

`pylsp_in_env` — это небольшой «сценарий-обёртка», который следует поместить в `PATH` со следующим (требует редактирования) содержимым:

```
#!/bin/bash
cd $1
# запустить сервер (python-lsp-server) в виртуальной среде
# (т.е. с заданными переменными виртуальной среды)
# поэтому источником будет виртуальная среда
source XYZ
# сервер и аргументы могут быть разными
exec myserver
```

Это только один из примеров более общего шаблона, с которым может быть немного удобнее работать (описание доступно в разделе [Среда выполнения](#) ниже).

#### 4.15.4.1 Конфигурация сервера LSP

Каждый сервер LSP имеет свой собственный способ настройки; для конфигурации могут использоваться средства, специфичные для языка или инструмента, например, `tox.ini` (в том числе для `python`), `.clang-format` для форматирования в стиле C++. Такая конфигурация затем может также использоваться другими (отличными от LSP) инструментами (такими как `tox` или `clang-format`). Кроме того, некоторые серверы LSP загружают конфигурацию из пользовательских файлов (например, `.ccls`). Более того, настраиваемая конфигурация сервера также может быть передана с помощью LSP (протокол): смотрите вышеупомянутые записи «`initializationOptions`» и «`settings`» в конфигурации сервера.

Поскольку применяются различные уровни переопределения и слияния, в приведённом далее примере определённые пользователем параметры конфигурации клиента корректируют некоторые из параметров конфигурации `python-language-server`.

```
{
  "servers": {
    "python": {
      "settings": {
        "pyls": {
          "plugins": {
            "pylint": {
              "enable": true
            }
          }
        }
      }
    }
  }
}
```

```
}
}
```

К сожалению, конфигурация сервера LSP зачастую не очень хорошо документирована, поэтому только изучение исходного кода позволяет найти способы настройки и набор доступных параметров конфигурации. В частности, сервер из приведённого выше примера поддерживает намного больше параметров в «settings». Ознакомьтесь с [документацией другого клиента LSP](#), содержащей примеры серверов других языков и соответствующие параметры настройки, которые возможно легко и просто преобразовать в конфигурацию JSON, которая описана в этом руководстве.

#### 4.15.4.2 Форматирование при сохранении сервера LSP

Можно включить «форматирование при сохранении» в диалоге настройки параметров LSP.

#### 4.15.4.3 Подавление диагностики сервера LSP

Иногда диагностические данные могут оказаться не слишком полезны. Диагностика может быть лишней, особенно если диагностических сообщений слишком много (часто однотипных). В некоторых случаях поток сообщений можно скорректировать по языку программирования (на сервере). Например, [механизм настройки clangd](#) позволяет корректировать некоторые аспекты диагностики. Тем не менее, способ корректировки не всегда очевиден, а также желаемая корректировка может быть невозможна из-за ограничений сервера или внутренней ошибки.

Поэтому в модуле предусмотрена поддержка подавления диагностических сообщений (она похожа, например, на подавление valgrind). Самую тонкую настройку возможно выполнить с помощью ключа «suppressions» в (объединённой) конфигурации JSON.

```
{
  "servers": {
    "c": {
      "suppressions": {
        "rulename": ["filename", "foo"],
        "clang_pointer": ["", "clang-tidy", "clear_pointer"],
      }
    }
  }
}
```

Каждое (корректное) правило имеет произвольное имя и определяется массивом длины 2 или 3, в котором указано регулярное выражение для установления соответствия (полного) имени файла, регулярное выражение для установления соответствия диагностического текста и необязательное регулярное выражение для установления соответствия (диапазона исходного кода) текста, к которому применяется диагностика.

Помимо вышеупомянутой тонкой настройки, в контекстном меню на вкладке диагностики также предусмотрено добавление и удаление подавлений, которые точно соответствуют определённому диагностическому тексту (либо глобально (любой файл), либо локально (конкретный рассматриваемый файл)). Эти подавления хранятся и загружаются из конфигурации сеанса.

#### 4.15.4.4 Устранение неполадок сервера LSP

Одно дело описать настройку (нестандартного) сервера LSP для какого-либо языка программирования, другое же — сделать так, чтобы сервер работал без каких-либо проблем. К счастью, обычно сервер работает именно так. Впрочем, иногда могут возникать сложности из-за случайных допущенных неточностей в параметрах настройки или более серьёзных проблем с самим сервером. Второй вариант может проявлять себя в виде сообщения о неудачных

попытках запустить сервер, которое появляется на вкладке вывода данных Kate. Следует учитывать, что такие сообщения представляют собой лишь сообщения верхнего уровня или хода обработки, а не подробные диагностические сообщения, а также могут не содержать данных о том, что фактически является другим процессом (сервер LSP).

Обычный способ диагностики заключается в том, чтобы добавить к команде запуска (языкового сервера) один или несколько флагов; это включает (дополнительное) журналирование (в какой-либо файл или стандартный поток ошибок), если оно не выполняется по умолчанию. Если затем запустить Kate из командной строки, возможно, получится узнать о проблеме больше.

Информацию также можно получить, изучив обмен данными между клиентом LSP Kate и сервером LSP. Опять же, на последнем обычно есть средства трассировки. Клиент LSP также предоставляет дополнительную трассировку отладки (в stderr) при вызове Kate с должным образом экспортированной переменной среды LSPCLIENT\_DEBUG=1.

#### 4.15.4.5 Настройка среды выполнения

Приведённый выше пример виртуальной среды python — лишь один из вариантов «среды выполнения», работающей отличным и отдельным от обычной среды основной системы образом. Этого можно достичь с помощью различных параметров переменных (например, virtualenv) или настройки (s)chroot (переключения на другой каталог в качестве нового корня), контейнера (например, podman, docker) или ssh-сеанса к другой основной системе. В каждом из этих случаев «другая среда» определяется «префиксом выполнения». Это означает, что в этой другой среде программу можно вызвать/запустить с помощью «префикса» (программа и аргументы), добавленного к предполагаемому вызову. Например, podman exec -i containername или ssh user@host.

В частности, как было сделано в приведённом ранее примере virtualenv, можно запустить сервер LSP в такой отдельной среде (например, контейнер со всеми необходимыми проекту зависимостями). «Ручной» подход, описание которого приведено выше, имеет следующий недостаток: выполняется замена стандартной командной строки LSP, её потом необходимо указать и продублировать ещё раз в сценарии-обёртке. Кроме того, в некоторых других вышеупомянутых примерах «путь к пространству имён» основной системы (как он виден редактору) может отличаться от соответствующего для среды. Чтобы решить эти вопросы более системно (а не просто с помощью «индивидуального» подхода), можно дополнить конфигурацию.

Например, в конфигурации .kateproject можно указать следующее. Очевидно, что «фиктивные» комментарии включать не следует.

```
{
  // это также может быть массив объектов
  "exec": {
    "hostname": "foobar"
    // команда также может быть массивом строк
    "prefix": "podman exec -i foobarcontainer",
    "mapRemoteRoot": true,
    "pathMappings": [
      // возможна любая из следующих форм
      // также существует более автоматизированная альтернатива, по ←
      // подробнее о ней далее/ниже
      [ "/dir/on/host", "/mounted/in/container" ]
      { "localRoot": "/local/dir", "remoteRoot": "/remote/dir" }
    ]
  },
  "lspclient": {
    "servers": {
      "python": {
        // это устанавливает соответствие или выполняет объединен ←
        // ие для объекта выше
        "exec": { "hostname": "foobar" },

```



## Руководство пользователя Kate

- Во время выполнения также задаются некоторые переменные среды, с помощью чего можно (тонко) настроить поведение средства запуска с префиксом (но ещё раз обратите внимание на первый пункт). В частности, `KATE_EXEC_PLUGIN` устанавливается в значение `lspclient`, а `KATE_EXEC_SERVER` — в значение идентификатора сервера (например, `python`).
- В частности, `KATE_EXEC_INSPECT` также имеет значение 1. Это сообщает «средству запуска с префиксом», что получатель (модуль Kate) принимает некоторые данные вне диапазона/протокола. Таким образом, средство запуска может использовать некоторые возможности для определения привязок путей (например, `podman inspect`) и передать эту информацию в подходящем формате. Затем это будет удалено из потока, который в остальном соответствует протоколу LSP, и использовано для расширения заданной привязки путей. Это может быть альтернативой явной привязке ресурсов (снова дублирование определения контейнера). Конкретно, вместо этого затем можно было бы использовать вышеприведённый фрагмент;

```
{
  // ...
  "exec": {
    "hostname": "foobar"
    // нежелательный повтор имени, но вспомогательный сценарий поня ←
    // тен и прост
    "prefix": "exec_inspect.sh foobarcontainer podman exec -i ←
      foobarcontainer",
    "mapRemoteRoot": true,
    "pathMappings": []
  }
  // ...
}
```

И последнее по порядку, но не по значимости: что, если подход с «резервным КИО» не считается подходящим? На практике зачастую можно «подключить» сетевой корневой каталог к локальной файловой системе и затем указать последнюю в `pathMappings`. Для начала, широко известная файловая система `sshfs` (`fuse`) может подключить (действительно) сетевую систему к локальной. Если используется контейнер (`podman/docker`), большинство конфигураций (драйвера файловой системы) поддерживают следующий трюк;

```
$ rootdir=/proc/$(podman inspect --format '{{.State.Pid}}' containername) / ←
  root
# некоторые символьные ссылки могут приводить к проблемам, ниже приводитс ←
# я альтернатива
$ sudo mount --bind $rootdir /somewhere/containername
```

Если это не работает, можно воспользоваться `sshfs` или его вариантом для подключения корневого каталога сетевого ресурса или контейнера;

```
# смотрите соответствующие man-страницы; для обычного (незашифрованного) ←
# протокола обработки файлов достаточно sftp-server
$ socat 'exec:podman exec -i containername /usr/lib/openssh/sftp-server'
  'exec:sshfs -o transform_symlinks -o passive \:/ /somewhere/ ←
  containername'
# ...
$ fusermount -u /somewhere/containername
```

Обратите внимание, что такое подключение сетевого корневого каталога к основной файловой системе, скорее всего, потребует указывать вручную и в явном виде в разделе `pathMappings` (за исключением случаев, когда используется интеллектуальное средство запуска с префиксом, которое организывает всё это автоматически).

## 4.16 Модуль поиска и замены

### 4.16.1 Введение

Модуль «Поиск и замена» позволяет провести поиск по тексту или по [регулярным выражениям](#) сразу в нескольких файлах. Поиск возможно провести во всех открытых файлах, всех файлах в одном каталоге и (опционально) его подкаталогах или в активном файле. Возможно даже установить фильтр по имени файла, например, чтобы произвести поиск только в файлах с определённым расширением.

### 4.16.2 Интерфейс

#### 4.16.2.1 Поисковый запрос

В верхней части панели поиска файлов всегда отображаются следующие опции:



Возможно открыть сразу несколько вкладок поиска. Чтобы открыть новую вкладку, позволяющую произвести другой поиск, просто нажмите кнопку открытия новой вкладки, расположенную в верхнем левом углу панели поиска.



Кнопка в верхнем правом углу панели поиска в файлах позволяет переключаться между отображением дополнительных функций для режима поиска в папке и отображением результатов поиска в нижней части панели.

#### Искать

В это поле следует вводить текст для поиска. Поддерживается простой текст или регулярные выражения (если соответствующая функция включена).

#### Заменить на (текстовое поле)

Заменяющий текст, который будет добавлен в файл(ы) вместо текста в текстовом поле Искать.

#### Выполнить поиск

Закончив настройку, нажмите кнопку Выполнить поиск. Чтобы выполнить поиск в новой вкладке, нажмите и удерживайте кнопку Ctrl перед нажатием кнопки Выполнить поиск. Для выполнения поиска также возможно нажать клавишу Enter в текстовом поле Искать (соответственно, при нажатии комбинации клавиш Ctrl+Enter поиск будет выполнен в новой вкладке).

#### Заменить

Закончив настройку, нажмите кнопку Заменить, чтобы произвести замену текста, введённого в текстовом поле Искать, на текст, указанный в текстовом поле Заменить на. Для выполнения замены также возможно нажать клавишу Enter в текстовом поле Заменить на.


#### Далее

Перейти к следующему элементу в результатах поиска, при необходимости переключившись на другой файл.

#### Заменить отмеченные

То же самое, что и кнопка Заменить, однако сработает только на отмеченных файлах.

#### 4.16.2.2 Параметры поиска в папке

Эти параметры отображаются под вышеупомянутыми параметрами запроса. Если вместо этого отображаются результаты поиска, нажмите кнопку , чтобы отобразить параметры поиска в папке.

##### Поиск

Предусмотрено три варианта: В открытых файлах — поиск по всем файлам, открытым на данный момент в Kate, В папке — поиск внутри папки и (опционально) во вложенных папках, В текущем файле — в текущем активном файле.

Если подключён модуль проектов, возможно также произвести поиск в текущем проекте или во всех открытых проектах.

##### С учётом регистра

Ограничить результаты поиска только теми, которые содержат ту же комбинацию прописных и строчных букв, что и поисковый запрос.

##### Регулярные выражения

Позволяет использовать в поиске [регулярные выражения](#) вместо простого текста.

##### Развернуть результаты

Показать все результаты, найденные в каждом файле, а не просто список файлов, которые содержат поисковый запрос.

##### Папка

Возможно ввести путь к папке, в которой требуется выполнить поиск. Например, чтобы выполнить поиск в исходном коде Kate, введите `~/development/kde/kate/`. Эта функция доступна только при использовании режима В папке.

##### Выбрать файл или папку

Нажмите эту кнопку, чтобы указать расположение папки с помощью диспетчера файлов. Кнопка активна только в режиме поиска В папке.



Нажмите эту кнопку, чтобы указать в поле Папка родительскую папку текущей выбранной папки. Кнопка активна только в режиме поиска В папке.



Эта кнопка позволяет указать в поле Папка папку текущего открытого документа. Кнопка активна только в режиме поиска В папке.

##### Фильтр

Позволяет выполнять поиск только в тех файлах, которые соответствуют определённому шаблону. Например, чтобы провести поиск только по файлам, написанным на C++, введите `*.cpp`. Чтобы выполнить поиск в файлах, начинающихся с «kate», введите `kate*`. Поддерживается ввод нескольких фильтров, которые между собой следует разделять запятой (,). Эта функция недоступна при использовании режима В открытых файлах.

##### Исключить

Функция, противоположная Фильтру: исключает из поиска те файлы, которые соответствуют определённому шаблону. Поддерживается ввод нескольких шаблонов, которые между собой следует разделять запятой (,). Эта функция недоступна при использовании режима В открытых файлах.

##### Рекурсивно

Если этот флажок установлен, Kate проведёт поиск также и во вложенных папках выбранного каталога. Эта функция доступна только в режиме В папке.

#### Искать в скрытых файлах

Если этот флажок установлен, Kate проведёт поиск также и в скрытых файлах и папках, которые обычно не отображаются операционной системой. Эта функция доступна только в режиме В папке.

#### Следовать по символическим ссылкам

Модуль поиска в файлах обычно не обрабатывает **символические ссылки**. Когда эта функция активирована, модуль будет выполнять переход по ним и осуществлять поиск в файлах или папках, на которые они ссылаются. Эта функция доступна только в режиме В папке.


#### Внимание

Символические ссылки могут ссылаться на родительскую папку текущей папки, в которой производится поиск, либо на другие папки, содержащие символические ссылки на свои родительские папки. Если эта опция включена и в папке, в которой производится поиск, содержится такая ссылка, Kate будет повторно переходить по ссылке и выполнять поиск, то есть он никогда не будет завершён.

#### Искать в двоичных файлах

Если этот флажок установлен, Kate будет также производить поиск по файлам, которые не являются текстовыми файлами.

#### 4.16.2.3 Результаты поиска

Результаты поиска отображаются под параметрами запроса. Если вместо этого отображаются параметры поиска в папке, нажмите кнопку . Результаты поиска также начнут отображаться автоматически после того, как поиск будет произведён.

Результаты поиска отображают список файлов, содержащих искомый текст, а также количество найденных в каждом из них совпадений.

Чтобы просмотреть список совпадений в файле, просто щёлкните по стрелке разворачивания, расположенной слева от названия этого файла (если была выбрана опция Развернуть результаты, список уже будет развёрнут). Для каждого найденного совпадения будет показан номер строки и её содержимое (с поисковым запросом, выделенным полужирным шрифтом).

Чтобы открыть файл, в котором был найден искомый текст, просто дважды щёлкните по нему левой кнопкой мыши. Файл будет открыт в Kate, если это требуется. Также возможно переместить курсор к расположению конкретного совпадения: сделайте в списке двойной щелчок по этому совпадению, а не по имени файла.

#### 4.16.3 Структура меню

Правка → Поиск в файлах (Ctrl+Alt+F)

Открыть панель поиска и замены.

Правка → Перейти к следующему совпадению

Перейти к следующему совпадению из результатов поиска модулем поиска и замены.

Правка → Перейти к предыдущему совпадению

Перейти к предыдущему совпадению из результатов поиска модулем поиска и замены.

Вид → Боковые панели → Показать панель «Поиск и замена»

Включить или отключить отображение панели Поиск и замена.

## 4.17 Модуль «Сценарии и фрагменты» Kate

### 4.17.1 Введение

Модуль «Сценарии и фрагменты» Kate позволяет пользователю добавлять собственные элементы автоматизации и управлять ими. Они могут быть разными: от статических фрагментов обычного текста для многократного использования до очень сложных шаблонов с настраиваемыми автоматизированными полями с поддержкой сценариев, а также полностью настраиваемых сценариев JavaScript для изменения содержимого документа.

В большинстве случаев с этими типами содержимого можно работать одним и тем же образом. Поэтому, чтобы воспринимать информацию было легче, далее в тексте не говорится отдельно о статических фрагментах, содержащих сценарии фрагментах и сценариях. Вместо этого просто используется термин «фрагмент».

Модуль также поддерживает назначение комбинаций клавиш и вставку фрагментов с помощью автодополнения кода. Более того, можно загружать подборки фрагментов (так называемые комплекты) из Интернета.

### 4.17.2 Структура меню

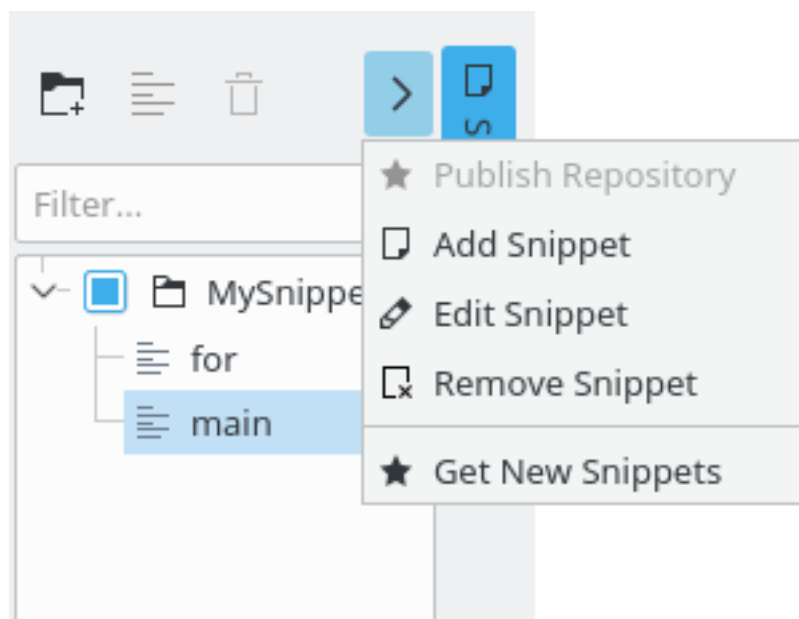
Вид → Боковые панели → Показать панель «Фрагменты»

Показывает панель фрагментов, содержащую все фрагменты комплекта, которые соответствуют типу текущего открытого файла.

Сервис → Создать фрагмент

Создать новый фрагмент, представляющий собой текст для многократного использования, который возможно вставить в любую часть любого документа, или небольшой сценарий.

### 4.17.3 Панель фрагментов



Панель фрагментов Kate.

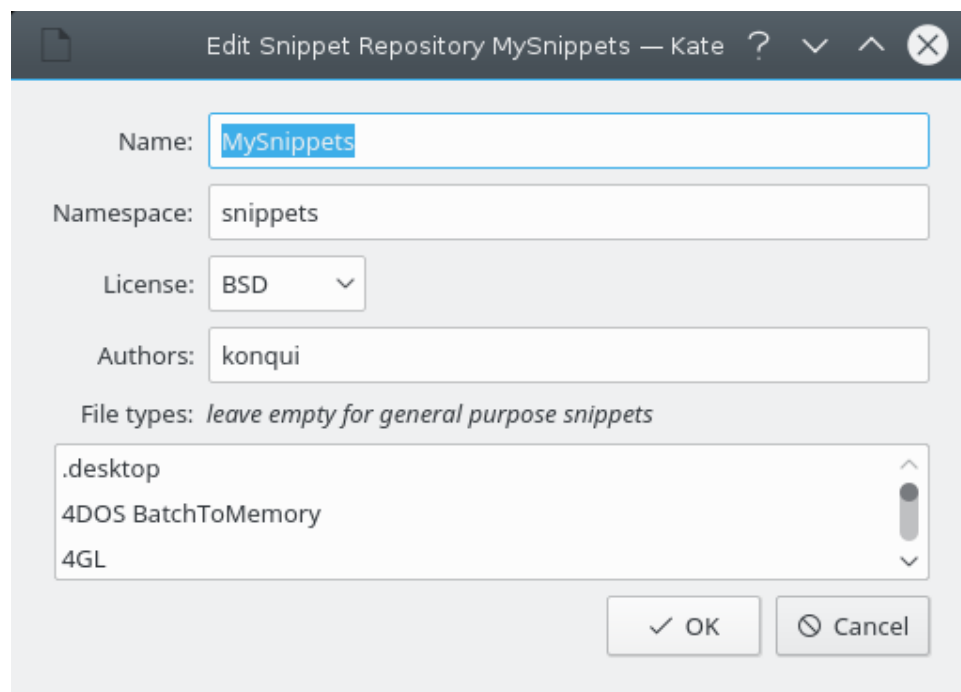
На панели отображается список комплектов фрагментов, а также опции создания собственных комплектов, получения их из Интернета или загрузки из локального файла. Для каждого комплекта предусмотрен флажок, который возможно использовать для его активации или деактивации. Также имеются кнопки для редактирования и удаления существующих комплектов.

#### 4.17.3.1 Загрузка файлов комплектов фрагментов

Комплекты фрагментов возможно загрузить из Интернета. Достаточно нажать кнопку Получить новые фрагменты, и откроется окно со списком комплектов фрагментов, доступных для загрузки. После загрузки комплект следует активировать.

#### 4.17.3.2 Создание и редактирование комплектов

Чтобы создать новый комплект фрагментов, нажмите кнопку Создать комплект. Появится диалоговое окно, в котором следует указать название файла фрагментов, лицензию и автора. Указав необходимые данные, нажмите кнопку ОК.



Интерфейс редактора комплектов.

В редакторе комплектов фрагментов доступны следующие параметры:

##### Название

Появляется в списке фрагментов на панели, а также по нему выполняется поиск при использовании функции автодополнения кода.

##### Пространство имён

Префикс при использовании функции автодополнения кода.

##### Лицензия

Позволяет выбрать лицензию для комплекта фрагментов.

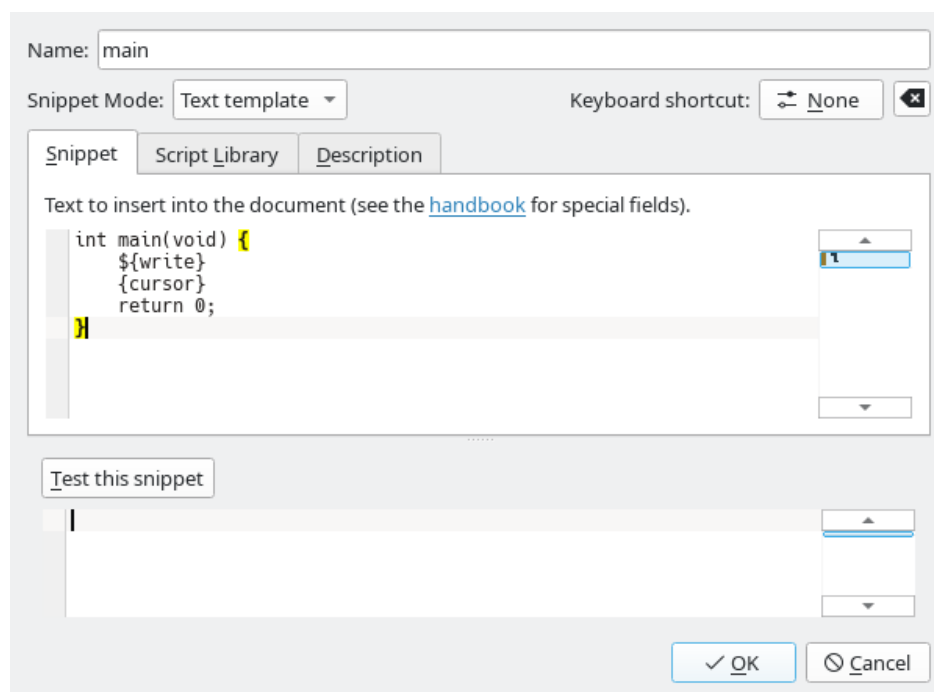
##### Авторы

Позволяет ввести имена авторов файла фрагментов.

## Типы файлов

Позволяет выбрать типы файлов, к которым должен применяться комплект фрагментов. По умолчанию установлено значение «», поэтому комплект будет применяться ко всем файлам. Установленное значение возможно изменить, указав, например, C++, или выбрав готовый вариант из списка. Возможно установить более одного типа, если удерживать нажатой клавишу Shift при их добавлении.

### 4.17.3.3 Создание и редактирование фрагментов



Интерфейс редактора фрагментов.

#### Название

Название в списке фрагментов и в списке автодополнения (обязательно).

#### Тип

Фрагменты могут быть текстовыми шаблонами или сценариями. Оба типа позволяют использовать функции JavaScript (подробные сведения приводятся далее) и, соответственно, получить схожие результаты. Вместе с тем, текстовые шаблоны лучше подходят для случаев, когда основной задачей является вставка текста, а сценарии зачастую значительно упрощают работу, когда текст нужно изменить.

#### Замечание

Чтобы эта функциональная возможность была доступна, программа Kate должна быть собрана с KTextEditor версии 6.15.0 или выше. При использовании более ранних версий будут доступны только фрагменты типа «текстовый шаблон».

#### Комбинация клавиш

Нажатие этой комбинации позволяет вставить фрагмент в текущий документ (или выполнить его там).

#### Фрагмент (тип Текстовый шаблон)

Текст, который будет вставлен в документ.

Фрагмент может содержать поля, которые возможно редактировать. Циклический переход между ними выполняется с помощью клавиши Tab. В тексте шаблона возможно использовать следующие выражения для создания полей:

`{имя_поля}` создаёт простое редактируемое поле. При редактировании все последующие вхождения того же имени\_поля будут копиями содержимого первого поля.

##### Подсказка

Имена полей могут содержать любые символы, кроме закрывающей фигурной скобки `}`.

Чтобы вставить `{text}` без изменения в виде фрагмента, экранируйте символ `$` обратной косой чертой: `\{text}`. В этом случае текст не превратится в поле. Чтобы вставить перед полем символы обратной косой черты без изменения, экранируйте каждый из них отдельной обратной косой чертой: `\\{field}`, `\\\\{field}` и так далее.

`{func(другое_поле1, другое_поле2, ...)}` создаёт поле, которое будет содержать актуальный результат выполнения функции JavaScript при каждом редактировании. Более подробные сведения доступны на вкладке [Библиотека сценариев](#).

`{имя_поля=значение по умолчанию}` устанавливает значение по умолчанию для поля. Значение по умолчанию — это выражение JavaScript. Используйте кавычки (`{имя_поля="текст"}`) для указания фиксированной строки в качестве значения по умолчанию.

##### Подсказка

При использовании значений по умолчанию (`{имя_поля=значение по умолчанию}`) следует помнить о том, что значение по умолчанию вычисляется сразу после вставки фрагмента в документ и не обновляется при изменении полей в дальнейшем.

В значениях по умолчанию можно ссылаться на другие поля, если таковые определены до вычисляемого значения по умолчанию. Но это позволит получить доступ только к стандартным значениям этих полей.

Если не задано пользовательское значение по умолчанию, значением по умолчанию будет имя поля.

`{cursor}` отмечает конечную позицию курсора после заполнения всех полей. Вставка текста в этой позиции приведёт к завершению редактирования. Кроме того, после нажатия клавиши Esc для завершения редактирования курсор перейдёт в эту позицию.

#### Фрагмент (тип Сценарий)

Код JavaScript для обработки этого фрагмента

Если в этом коде есть инструкция `return`, возвращённая строка будет вставлена в текущей позиции курсора. Также можно воспользоваться [программным интерфейсом сценариев Kate](#), чтобы внести изменения в документ напрямую.

Кроме того, в коде можно использовать функции, определённые на вкладке [Библиотека сценариев](#); они являются общими для всех фрагментов в комплекте.

#### Библиотека сценариев

Вспомогательные функции JavaScript для использования во фрагментах. Они являются общими для всех фрагментов в комплекте.

Возможно использовать [программный интерфейс для работы со сценариями Kate](#), чтобы получить обозначенный фрагмент текста, весь текст, имя файла и другие данные с помощью соответствующих методов объектов `document` и `view`. Дополнительные сведения доступны в документации программного интерфейса для работы со сценариями.

Когда функция вызывается в поле фрагмента типа Текстовый шаблон, возвращённое ей значение вставляется в текст.

**Замечание**

Использование функций во фрагментах типа Текстовый шаблон:

- Функции могут получать доступ к актуальному содержимому всех полей с помощью объекта `fields`: используйте `fields.my_field` или `fields["my_field"]`. Когда функция используется в качестве значения по умолчанию для поля, доступны только ранее определённые в тексте поля.
- Поля можно передавать функциям как аргументы. Можно использовать непосредственно имя поля, если оно представляет собой корректный идентификатор JavaScript и нет переменных с таким же именем. Если это не так, следует использовать объект `fields`: `${func(field)}`, `${func(fields.document)}`, `${func(fields["my_field"])}`
- Поля функций (например, `${func()}`) каждый раз вычисляются заново при изменении содержимого поля. Однако это не относится к вызовам функций в значениях по умолчанию; они вычисляются только один раз, при вставке фрагмента в документ.
- В случае более сложных сценариев важно понимать, что сначала в документ вставляется необработанный фрагмент, а потом — вычисляемые функции. Например, если функция извлекает текст в строке, куда вставляется фрагмент, этот текст также будет содержать `${functionCall()}`. Там, где это приводит к сложностям, рекомендуется использовать фрагмент типа Сценарий.
- Чтобы обернуть текущий выбранный текст в теги: `<strong>${view.selectedText()}</strong>`
- Помните, что в поле вставляется только возвращаемое значение функции.

**Description**

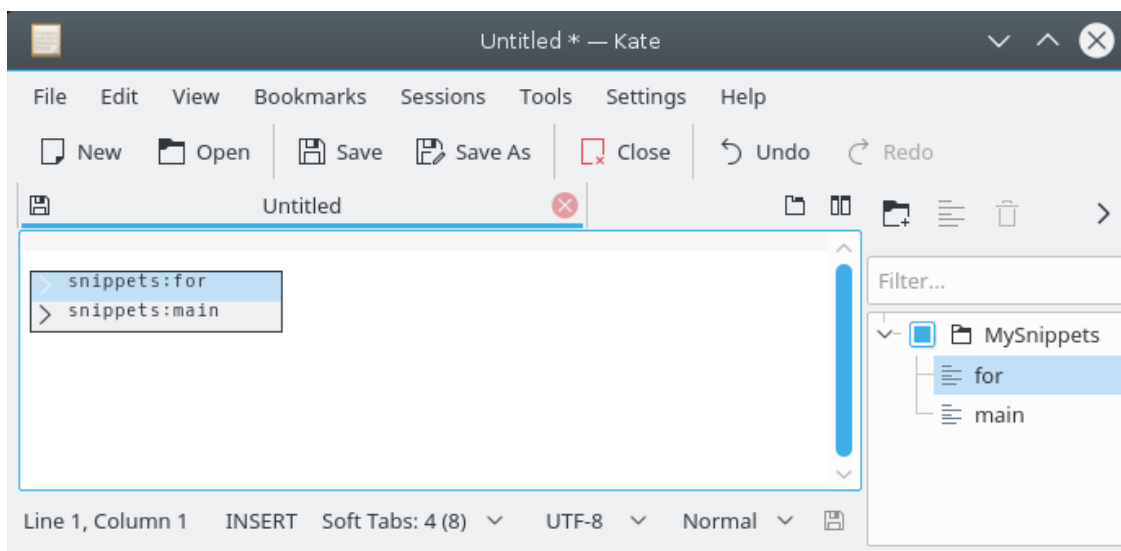
Необязательное описание функционала фрагмента. Оно будет показано во всплывающих подсказках. Описание может содержать базовое форматирование HTML.

**Пример использования**

В следующем примере вызывается сценарий, оборачивающий выбранный текст (если ничего не выбрано — текст по умолчанию) в теги (тип фрагмента Сценарий):

```
let range = view.hasSelection() ? view.selection() : new Range(view. ←
  cursorPosition(), view.cursorPosition());
let innertext = range.isEmpty() ? "Bold" : document.text(range);
document.removeText(range);
document.insertText(range.start, "<strong>" + innertext + "</strong ←
  >");
```

#### 4.17.4 Использование фрагментов



Процесс выбора из списка фрагментов.

Получить доступ к фрагментам возможно несколькими способами:

- Щёлкнуть по фрагменту на боковой панели.
- Воспользоваться комбинацией клавиш (если она была назначена).
- Использовать во время редактирования комбинацию клавиш `Ctrl+Пробел` для просмотра всех фрагментов в удобном окне и выбора нужного. Эта комбинация клавиш обеспечивает функциональность, аналогичную дополнению кода.

Если во фрагменте содержатся переменные (кроме `${cursor}`), курсор будет автоматически перемещён к первому вхождению переменной, после чего программа перейдёт в состояние ожидания ввода текста. Переход к следующей переменной выполняется с помощью клавиши `Tab`.

#### 4.17.5 Благодарности

Автор модуля «Фрагменты» Kate — Joseph Wenninger.

Благодарим участника «Google Code-In 2011» Martin Gergov за большой вклад в написание данного раздела.

### 4.18 Модуль клавиатурных макросов

#### 4.18.1 Введение

Запись и воспроизведение клавиатурных макросов (последовательностей действий на клавиатуре).

#### 4.18.2 Основные функции

##### 4.18.2.1 Чтобы начать запись клавиатурного макроса:

Сервис → Клавиатурные макросы → Записать макрос... (`Ctrl+Shift+K`).

Модуль будет записывать нажатия клавиш, пока пользователь не остановит запись.

#### 4.18.2.2 Чтобы завершить запись:

Сервис → Клавиатурные макросы → Завершить запись макроса (Ctrl+Shift+K).

Модуль прекратит записывать нажатия клавиш и сохранит их последовательность в качестве текущего макроса.

#### 4.18.2.3 Чтобы отменить запись:

Сервис → Клавиатурные макросы → Отменить запись макроса (Ctrl+Alt+Shift+K).

Модуль прекратит записывать нажатия клавиш, но текущий макрос не изменится.

#### 4.18.2.4 Чтобы воспроизвести текущий макрос:

Сервис → Клавиатурные макросы → Воспроизвести макрос (Ctrl+Alt+K).

Модуль воспроизведёт текущий макрос.

Переданная без аргументов команда `kmplay` также позволит воспроизвести текущий макрос.

### 4.18.3 Именованные макросы

Клавиатурные макросы можно сохранять, присваивая им имена.

Именованные макросы сохраняются при смене сеансов Kate (они хранятся в файле `keyboardmacros.json` в каталоге пользовательских данных Kate (обычно это `~/local/share/kate/`)).

#### 4.18.3.1 Чтобы сохранить текущий макрос:

Сервис → Клавиатурные макросы → Сохранить текущий макрос (Alt+Shift+K).

Модуль выведет запрос на ввод названия и сохранит под ним макрос.

Команда `kmsave name` сохранит текущий макрос под именем `name`.

#### 4.18.3.2 Чтобы загрузить сохранённый макрос в качестве текущего:

Сервис → Клавиатурные макросы → Загрузить именованный макрос. . . .

Это вложенное меню модуля содержит пункты, представляющие собой список сохранённых макросов. При активации какого-либо из пунктов будет загружен соответствующий макрос (в качестве текущего).

Команда `kmload name` загрузит макрос, сохранённый под именем `name`, в качестве текущего.

#### 4.18.3.3 Чтобы воспроизвести сохранённый макрос без его загрузки:

Сервис → Клавиатурные макросы → Воспроизвести именованный макрос. . . .

Это вложенное меню модуля содержит пункты, представляющие собой список сохранённых макросов. При активации какого-либо из пунктов будет воспроизведён (без загрузки) соответствующий макрос.

Обратите внимание, что каждый сохранённый макрос представляет собой действие, входящее в текущий набор действий, поэтому ему можно назначить настраиваемую комбинацию клавиш: Настройка → Комбинации клавиш. . . .

Команда `kmplay name` воспроизведёт макрос, сохранённый под именем `name`, не загружая его.

#### 4.18.3.4 Чтобы стереть (то есть удалить) сохранённый макрос:

Сервис → Клавиатурные макросы → Стереть именованный макрос. . . .

Это вложенное меню модуля содержит пункты, представляющие собой список сохранённых макросов. При активации какого-либо из пунктов будет стёрт (то есть удалён) соответствующий макрос.

Команда `kmwipe name` сотрёт макрос, сохранённый под именем `name`.

#### 4.18.3.5 Советы по командам:

Обратите внимание, что после префикса `km` во всех этих командах идут разные буквы, что позволяет с лёгкостью вызывать их с помощью системы дополнения по нажатию клавиши `Tab`!

### 4.18.4 Ограничения

В текущей версии программы клавиатурные макросы не воспроизводятся корректно, когда используются определённые типы виджетов графического интерфейса (например, `QMenu`, `QuickOpenLineEdit` или `TabSwitcherTreeView`). Сложно назвать точную причину, но есть основания полагать, что эти виджеты работают с событиями клавиатуры нестандартным образом.

## 4.19 Модуль SQL

### 4.19.1 Введение

SQL (structured query language, «язык структурированных запросов») — это специальный язык для обновления, удаления и запроса информации из баз данных.

Модуль SQL позволяет:

- Создать базу данных
- Подключиться к существующим базам данных
- Вставить и удалить данные из базы данных
- Выполнить запрос
- Отобразить результаты в таблице

### 4.19.2 Подключение к базе данных

Выберите пункт `Добавить подключение...` в меню SQL или на панели инструментов, а затем укажите драйвер базы данных Qt™, который следует использовать (среди доступных вариантов: `QSQLITE`, `QMYSQL3`, `QMYSQL`, `QODBC3`, `QODBC`, `QPSQL7` и `QPSQL`). Если нужный драйвер отсутствует в списке, его потребуется установить. Затем нажмите кнопку `Далее`.

Если выбранная база данных использует файл для сохранения данных, просто укажите путь к этому файлу и нажмите кнопку `Далее`. Если для работы с ней требуется подключение к серверу, введите имя хоста сервера, имя пользователя, пароль и другую информацию, которая запрашивается драйвером, а затем нажмите кнопку `Далее`.

Затем укажите название подключения и нажмите кнопку `Завершить`.

### 4.19.3 Выполнение запросов

#### 4.19.3.1 INSERT/DELETE/UPDATE

С помощью модуля SQL возможно вставлять, удалять и обновлять данные с помощью так же, как из командной строки или из программы. Просто введите запрос и нажмите кнопку Выполнить запрос на панели инструментов или выберите пункт меню SQL → Выполнить запрос (Ctrl+E).

---

#### Example 4.1 Примеры запросов

---

##### INSERT

```
INSERT INTO имя_таблицы ("запись1", "запись2", "запись3", "запись4", ←  
"запись5")  
VALUES ("значение1", "значение2", "значение3", "значение4", "значение ←  
5" )
```

##### DELETE

```
DELETE FROM имя_таблицы WHERE name = "текст"
```

##### UPDATE

```
UPDATE имя_таблицы SET "запись1" = "текст", "запись2" = "текст", "зап ←  
ись3" = "текст", "запись4" = "текст", "запись5" = "текст"
```

---

#### 4.19.3.2 SELECT

После выполнения запроса SELECT результат будет представлен либо в виде таблицы на панели Вывод данных SQL в нижней части Kate или в виде текста на панели Текстовый вывод SQL. Если имеется какая-либо ошибка, она будет показана в текстовом выводе.

---

#### Example 4.2 Пример запроса SELECT

---

```
SELECT * FROM имя_таблицы
```

---

На панели инструментов Вывод данных SQL доступны несколько кнопок:

##### Подогнать размеры столбцов

Изменяет размеры столбцов так, чтобы они соответствовали содержимому.

##### Подогнать размеры строк

Изменяет размеры строк так, чтобы они соответствовали содержимому.

##### Копировать

Выбирает всё содержимое таблицы и копирует его в буфер обмена.

##### Экспорт

Экспортирует всё содержимое таблицы в файл, буфер обмена или текущий документ в формате значений, разделённых запятыми.

#### Очистить

Удаляет всё из таблицы.

Цвета таблицы возможно изменить в разделе SQL параметров настройки (Настройка → Настроить Kate...).

### 4.19.4 Просмотр

Возможно просматривать базу данных с помощью боковой панели Схема базы данных, расположенной в левой части окна редактора. Отображаемая информация зависит от того, какой драйвер базы данных используется.

Чтобы обновить этот список, щёлкните правой кнопкой мыши в любом месте боковой панели и выберите опцию Обновить. Чтобы создать запрос к записи в списке, щёлкните правой кнопкой мыши по записи, выберите в контекстном меню пункт Создать запрос и затем выберите тип запроса (SELECT, UPDATE, INSERT или DELETE).

### 4.19.5 Структура меню

SQL → Добавить подключение...

Добавляет новое подключение с помощью драйвера базы данных.

SQL → Удалить подключение

Удаляет выбранное подключение.

SQL → Изменить подключение...

Позволяет изменить параметры текущего подключения.

Подключения

Все созданные подключения к базам данных отображены в списке между элементами меню Изменить подключение... и Выполнить запрос. Чтобы отправить запрос к соответствующей базе данных или внести изменения в эту базу данных, просто выберите один из этих пунктов.

SQL → Выполнить запрос (Ctrl+E)

Выполняет запрос пользователя.

### 4.19.6 Благодарности

Автор модуля SQL — Marco Mentasti.

Благодарим участника «Google Code-In 2011» Ömer Faruk ORUÇ за большой вклад в написание данного раздела.

## 4.20 Модуль просмотра идентификаторов

### 4.20.1 Работа с модулем «Закрывать кроме/только...»

Этот модуль позволяет разработчикам просматривать символы (функции, макросы и структуры) из исходного кода.

Чтобы просмотреть код, щёлкните по соответствующему элементу обработанной информации.

В настоящее время поддерживаются следующие языки:

C/C++, Java™, Perl, PHP, Python, Ruby, XSLT, Tcl/Tk, Fortran

Список возможностей:

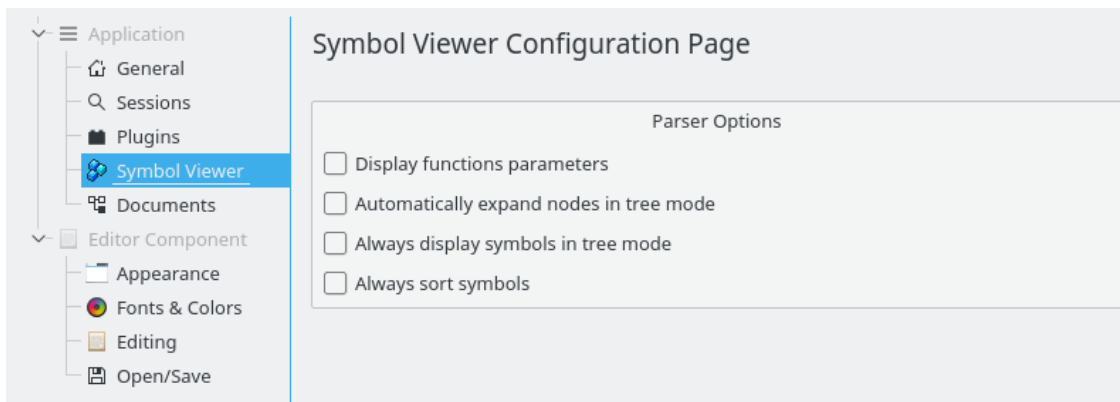
- Режим списка или дерева
- Включение или отключение сортировки
- Показ или скрытие параметров функций
- Разворачивание или сворачивание иерархического списка
- Автообновление при изменении документа
- Обработка кода, основанная на способах подсветки синтаксиса из KDE Frameworks

#### 4.20.2 Структура меню

Вид → Боковые панели → Показать панель «Список идентификаторов» (Ctrl+\\)

Включает или отключает отображение панели просмотра идентификаторов Kate.

#### 4.20.3 Настройка



Выбор параметров обработки по умолчанию

### 4.21 Модуль встроенного терминала

Встроенный эмулятор терминала — это копия терминала Konsole KDE, созданная для удобства пользователя. Чтобы открыть терминал, выберите пункт меню Вид → Боковые панели → Показать панель терминала (при отображении панель получит фокус ввода). Если подключена функция [Синхронизировать терминал с текущим документом автоматически \(если возможно\)](#), каталог в терминале будет при возможности изменён на каталог текущего документа после показа или внесения изменений в текущий документ.

По умолчанию окно терминала расположено в нижней части редактора под областью редактирования.

Konsole возможно настроить с помощью контекстного меню (доступно по щелчку правой кнопкой мыши). Более подробные сведения доступны в [руководстве пользователя Konsole](#).

Встроенный эмулятор терминала предоставляется модулем «Терминал».

### 4.21.1 Структура меню

Вид → Боковые панели → Показать панель терминала

Включает или отключает отображение встроенного эмулятора терминала.

Терминал будет создан при первичной активации модуля.

При отображении эмулятор терминала получает фокус, поэтому возможно сразу же начать вводить команды. Если активирована функция [Синхронизировать терминал с текущим документом автоматически \(если возможно\)](#) на странице Терминал в [главном диалоге настройки параметров](#), в сеансе командной оболочки в качестве текущего будет использоваться каталог активного документа, если это локальный файл.

Сервис → Перенаправить в терминал

Перенаправляет выделенный текст во встроенный эмулятор терминала. После текста не будет добавлен символ новой строки.

Сервис → Синхронизировать терминал с текущим документом

Во встроенном терминале текущим станет каталог активного документа.

Также возможно настроить Kate на постоянную синхронизацию терминала с текущим документом. Дополнительные сведения: [Раздел 4.21.2](#).

Сервис → Запустить выполнение текущего документа

Запускает выполнение текущего документа в Konsole. Можно запускать выполнение только локальных документов. Программа Kate выведет предупреждение перед запуском выполнения документа, так как это действие может быть серьезной угрозой безопасности.

Сервис → Активировать/Деактивировать терминал

Переключает фокус с текущего документа на терминал и наоборот.

### 4.21.2 Настройка

Модуль терминала возможно настроить на странице Терминал в [диалоге настройки редактора](#).

Доступны следующие опции:

Синхронизировать терминал с текущим документом автоматически (если возможно)

Встроенный терминал изменит текущий каталог на каталог активного документа после запуска и после того, как новый документ получит фокус. Если эта функция не включена, все действия по навигации в терминале потребуются выполнять вручную.

Установить переменную среды EDITOR равной «kate -b»

Устанавливает переменную среды EDITOR в такое значение, чтобы программы, работающие во встроенном терминале и автоматически открывающие файлы в редакторе, открывали их в Kate вместо обычного текстового редактора командной оболочки. Продолжить использование терминала будет возможно только после закрытия редактируемого файла в Kate (в этом случае вызывающая программа получит сообщение о завершении редактирования файла).

Скрыть Konsole при нажатии клавиши «Esc»

Эта опция позволяет закрыть встроенный терминал нажатием клавиши Esc. Могут возникнуть конфликты с приложениями терминала, в которых используется клавиша Esc (например, с vim). Добавьте такие приложения в расположенное ниже поле для ввода текста. Пункты списка следует разделять запятой.

## 4.22 Модуль текстового фильтра

### 4.22.1 Работа с модулем текстового фильтра

Этот модуль используется для обработки выделенного текста с помощью команд терминала. Выделенный текст используется в качестве входных данных команды. В зависимости от выбранного пользователем варианта, выходные данные будут либо использованы для замены выделенного текста, либо скопированы в буфер обмена.

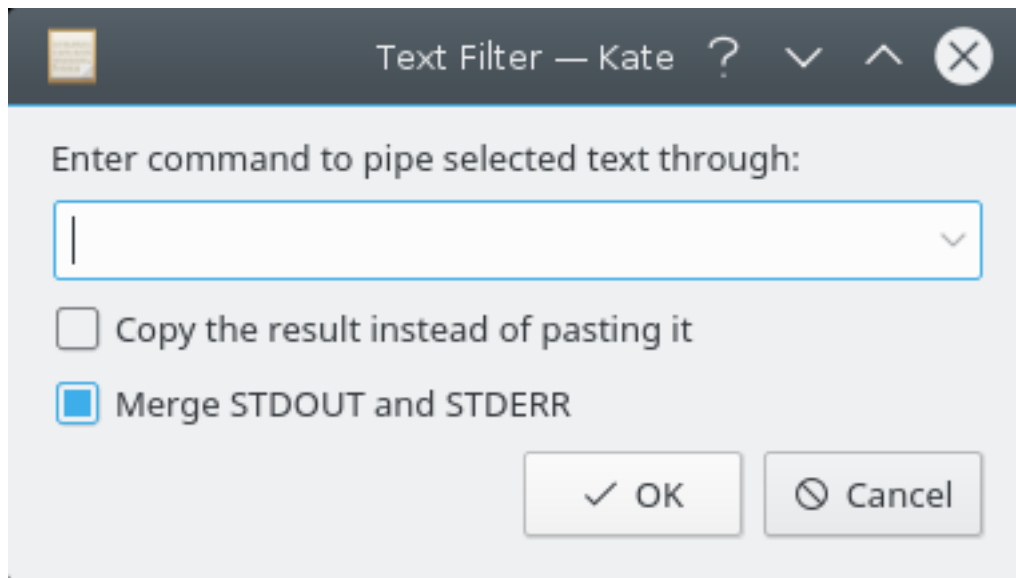
Примеры:

- `less /etc/fstab` — вставить содержимое этого файла или скопировать его в буфер обмена
- `wc` — посчитать строки, слова и символы выбранного текста и вставить эти данные в документ или скопировать их в буфер обмена
- `sort` — отсортировать строки выбранного текста и вставить результат в документ или скопировать его в буфер обмена

### 4.22.2 Структура меню

Сервис → Применить заданный командой фильтр... (Ctrl+\)

Открывает диалог текстового фильтра:



Введите в поле команду оболочки или выберите предыдущую команду из журнала.

Копировать результат (не вставлять)

Копировать результат в буфер обмена, не изменяя документ.

Объединить STDOUT и STDERR

Если флажок установлен, вывод STDOUT и STDERR будет объединён и сообщений об ошибках не будет. Если флажок снят, STDERR выводится как отдельное пассивное сообщение.

## 4.23 Модуль проверки XML

Данный модуль помогает проверить документ XML на допустимость и правильность формата.

Когда модуль проверяет текущий файл, в нижней части главного окна Kate выводится список ошибок и предупреждений. Чтобы перейти к конкретной ошибке в файле, выберите соответствующий элемент в списке. Если в файле есть тег «ДОСТУРЕ», то для проверки допустимости синтаксиса будет использоваться указанный этим тегом DTD. Поиск файла DTD будет проводиться относительно расположения файла текущего документа, то есть, если тег документа указан как «DTD/xhtml1-transitional.dtd», а файл документа расположен по адресу /home/peter/test.xml, программа ожидает, что файл DTD находится по адресу /home/peter/DTD/xhtml1-transitional.dtd. Приложением поддерживаются и внешние DTD, указанные с помощью http-адреса.

Если в файле отсутствует параметр `doctype`, то проверяется правильность формата.

Более подробные сведения об XML доступны на [официальных страницах XML W3C](#).

Для работы модуля требуется внешняя команда `xmllint`, входящая в состав библиотеки `libxml2`. Без неё он не сможет правильно работать.

Чтобы подключить этот модуль, откройте диалог настройки параметров Kate (Настройка → Настроить Kate...), выберите пункт Проверка XML в разделе Приложение / Модули и закройте диалог.

### 4.23.1 Структура меню

XML → Проверить корректность XML

Запуск проверки, как объяснено выше.

### 4.23.2 Благодарности

Авторские права на модуль «Проверка XML» Kate: (c) Daniel Naber [daniel.naber@t-online.de](mailto:daniel.naber@t-online.de), 2002.

Авторские права на документацию: (c) Daniel Naber, 2002.

## 4.24 Модуль автодополнения для XML

Этот модуль позволяет получать подсказки о том, что разрешено в определённой позиции в файле XML, в соответствии с DTD файла. Будут перечислены возможные элементы, атрибуты, значения атрибутов или объекты, в зависимости от положения курсора (например, все объекты перечислены, если символ слева от курсора — «&»). Также возможно закрыть ближайший открытый тег слева.

Для работы модуля необходим файл DTD в формате XML, созданный программой на Perl `dtddata`. DTD в таком формате называется «мета DTD». Некоторые из мета DTD устанавливаются вместе с программой. Они доступны в каталоге `katexmltools/ qtpaths --paths GenericDataLocation`, который также является папкой по умолчанию при выборе опции `Назначить мета DTD....` Чтобы создавать свои файлы мета DTD, загрузите программу `dtddata` с веб-сайта <http://dtddata.sourceforge.net>.

#### 4.24.1 Использование

Запустите Kate и откройте диалог настройки: Настройка → Настроить Kate.... Затем выберите Автодополнение для XML на странице Приложение → Модули и закройте окно. После этого выберите XML → Назначить мета DTD.... Если документ не содержит значения «DOCTYPE» или тип документа неизвестен, потребуется указать расположение файла мета DTD в файловой системе. В ином случае будет автоматически загружен файл мета DTD, совпадающий с DOCTYPE текущего документа.

Теперь модуль возможно использовать при вводе текста:

< (клавиша «меньше чем»)

При нажатии этой клавиши будет показан список возможных элементов (если только курсор уже не находится внутри тега). Следует отметить, что сейчас эту возможность нельзя использовать для вставки элемента верхнего уровня (например, «<html>»).

</(клавиша «меньше чем» + косая черта)

При вводе этих символов будет предложено закрыть текущий элемент (ближайший открытый тег слева от курсора). Чтобы подтвердить, нажмите клавишу Enter. В отличие от пункта меню Закрыть элемент, это сработает только при назначенном DTD.

" (кавычки)

При нажатии клавиши «кавычки» будет показан список возможных значений атрибутов (если они есть), если курсор находится внутри тега.

(пробел)

При нажатии клавиши «пробел» будет показан список возможных атрибутов для текущего элемента, если курсор находится внутри тега.

& (клавиша «&»)

При нажатии этой клавиши будет показан список именованных объектов.

#### 4.24.2 Возможности и ограничения

Чтобы проверить все функциональные возможности и ограничения, загрузите файл `katexmltools/testcases.xml` в каталог `qtpaths --paths GenericDataLocation` редактора Kate и следуйте инструкциям из этого файла.

#### 4.24.3 Структура меню

XML → Вставить элемент... (Ctrl+Enter)

Открыть диалоговое окно, позволяющее вставить элемент XML. Символы <, > и закрывающий тег будут вставлены автоматически. Если перед этим был выбран текст, он будет автоматически заключён в открывающий и закрывающий теги. Если файл мета DTD был назначен с помощью пункта меню Назначить мета DTD..., в этом диалоговом окне также будет возможно завершать все элементы, которые вставляются в текущей позиции курсора.

XML → Закрыть элемент (Ctrl+<)

Проверить текст на наличие незакрытых тегов и закрыть их соответствующими закрывающими тегами. Поиск начинается с позиции курсора и осуществляется в направлении справа налево. Если не будет найден открытый тег, ничего не произойдёт.

XML → Назначить мета DTD...

Указать, какой файл мета DTD следует использовать для текущего документа. Обратите внимание, что параметры не будут сохранены и их потребуется задать повторно при следующем запуске Kate.

#### 4.24.4 Благодарности

Авторские права на модуль «Автодополнение для XML» Kate: (c) Daniel Naber [daniel.naber@t-online.de](mailto:daniel.naber@t-online.de), 2001, 2002

KDE SC версия 4 (c) Tomáš Trnka, 2010

Авторские права на документацию: (c) Daniel Naber, 2001, 2002

### 4.25 Модуль просмотра компиляции

Этот модуль позволяет использовать навигатор компилятора внутри программы Kate. Навигатор компилятора обычно ограничивается только одним файлом, и его сложно использовать для просмотра кода файла в проекте. Этот модуль упрощает данный процесс путём автоматического определения команд компиляции для файла.

#### 4.25.1 Использование

Первое, что следует сделать — открыть вкладку навигатора компилятора. Воспользуйтесь комбинацией клавиш `Ctrl+Alt+I` для открытия панели команд, найдите пункт «Открыть текущий файл в навигаторе компилятора» и нажмите клавишу ввода. Чтобы упростить доступ к этому действию, можно добавить его на панель инструментов или назначить для него комбинацию клавиш.

Далее потребуется настроить локальный экземпляр навигатора компилятора или воспользоваться общедоступным экземпляром <https://godbolt.org/>. См. <https://github.com/compiler-explorer/compiler-explorer>. Как только экземпляр появится, воспользуйтесь полем с раскрывающимся списком Параметры на открытой ранее вкладке навигатора компилятора и выберите пункт Изменить URL-адрес. . . , а затем введите URL-адрес сервера навигатора компилятора.

Далее потребуется `compile_commands.json`, если нужно просмотреть codegen для файла в проекте. Если такой необходимости нет, этот шаг можно пропустить. Нужный результат легко получить с помощью современной системы сборки, например CMake или Meson. Если используется CMake, достаточно передать

```
-DCMAKE_EXPORT_COMPILE_COMMANDS=ON
```

 при вызове `cmake`.

`compile_commands.json` следует разместить в корневом каталоге проекта. Закройте и снова откройте вкладку навигатора компилятора в Kate. После повторного открытия вкладки на ней должны появиться команды компиляции, необходимые для сборки файла.

Если навигатор компилятора используется просто для просмотра codegen для чего-либо временного, можно просто записать команды компиляции для файла и нажать кнопку Скомпилировать. Обратите внимание, что автоматическая перекомпиляция при редактировании не поддерживается. Для перекомпиляции файла необходимо каждый раз нажимать «Скомпилировать».

Как только файл будет скомпилирован, справа появится codegen. Можно сделать щелчок правой кнопкой мыши и выбрать в контекстном меню пункт Показать связанный код, чтобы просмотреть codegen для конкретной строки. Аналогичным образом работает щелчок правой кнопкой мыши справа: выберите в контекстном меню пункт Прокрутить к источнику для просмотра исходного кода указанной строки codegen.

### 4.26 Модуль форматирования

Этот модуль позволяет с лёгкостью форматировать код. Он обеспечивает сохранение журнала отмены действий над документом и позиции курсора пользователя во время форматирования кода, чтобы форматирование кода не мешало работе. Это особенно важно для автоматического форматирования при сохранении.

#### 4.26.1 Использование

Модуль позволяет пользователю форматировать код двумя способами:

- Вручную (с помощью действия «Форматировать документ»)
- Автоматически при сохранении

**Подсказка**

Можно найти это действие на панели команд и назначить для него комбинацию клавиш, чтобы упростить доступ

#### 4.26.2 Поддерживаемые языки и средства форматирования

Текущий список поддерживаемых языков и средств форматирования:

- C/C++/ObjectiveC/ObjectiveC++/Protobuf/GLSL/Java - clang-format
- Javascript/Typescript/JSX/TSX - prettier
- JSON
  - prettier
  - clang-format
  - jq
- Dart - dartfmt
- Rust - rustfmt
- Go - gofmt
- XML - xmllint
- Zig - zigfmt
- CMake - cmake-format
- D - dfmt
- Fish Shell - fish\_indent
- Bash - shfmt
- Nix - nixfmt
- QML - qmlformat
- HTML - prettier
- Swift - swiftformat
- Erlang - erlfmt
- Godot Script - gdformat
- Python
  - autopep8
  - ruff
- YAML
  - yamlfmt
  - prettier

### 4.26.3 Настройка

Модуль можно настроить двумя способами:

- На глобальном уровне (с помощью окна настройки программы)
- Или на уровне проекта (с помощью файла `.kateproject`)

Модуль сначала выполнит попытку прочитать конфигурацию из файла `.kateproject`, а уже потом глобальную. Пример:

```
{
  "formatOnSave": true,
  "formatterForJson": "jq",
  "cmake-format": {
    "formatOnSave": false
  },
  "autopep8": {
    "formatOnSave": false
  }
}
```

Код выше:

- включает «форматирование при сохранении» на глобальном уровне
- указывает «jq» в качестве средства форматирования для JSON
- отключает «форматирование при сохранении» для `cmake-format` и `autopep8`

Чтобы настроить форматирование для проекта, сначала создайте файл `.kateproject`, а затем добавьте в него объект

“formatting”. В объекте

“formatting” можно указать параметры (как показано в предыдущем примере). Пример:

#### Замечание

Чтобы использовать эту возможность, необходимо включить [Модуль проектов](#)

```
{
  "name": "My Cool Project",
  "files": [
    {
      "git": 1
    }
  ],
  "formatting": {
    "formatterForJson": "clang-format",
    "autopep8": {
      "formatOnSave": false
    }
  }
}
```

### 4.26.4 Временно отключить форматирование при сохранении

При работе часто возникает потребность на время отключить форматирование. Это можно сделать с помощью меню: Сервис → Форматировать при сохранении. Форматирование также можно включить или отключить на Панели команд.

## Глава 5

# Дополнительные возможности редактирования

Более подробные сведения о дополнительных возможностях редактирования доступны в разделе «Дополнительные возможности редактирования» руководства пользователя [KatePart](#).

## Глава 6

# Расширение функциональных возможностей Kate

T.C. Hollingsworth

Перевод на русский язык: Андрей Балагута

Перевод на русский язык: Алексей Опарин

Обновление перевода: Мария Шикунова

Редактирование перевода: Олеся Герасименко

### 6.1 Введение

Как и любой другой улучшенный текстовый редактор, Kate предоставляет разнообразные возможности расширения своей функциональности. Возможно [создавать простые сценарии](#), которые реализуют дополнительные возможности, с помощью JavaScript, а также использовать модули Kate, написанные на языке C++. Расширив возможности Kate, [присоединяйтесь к нам](#) и делитесь созданными инструментами с пользователями по всему миру!

### 6.2 Работа с подсветкой синтаксиса

Сведения о добавлении или изменении определений подсветки синтаксиса доступны в [разделе «Подсветка синтаксиса»](#) руководства пользователя KatePart.

### 6.3 Написание сценариев JavaScript

Сведения о написании сценариев с использованием JavaScript доступны в [разделе «Написание сценариев JavaScript»](#) руководства пользователя KatePart.

### 6.4 Модули Kate (C++)

[Модули Kate](#), написанные на том же языке, что и сам редактор Kate (C++), позволяют значительно расширить его функциональные возможности.

Начните работу с посещения веб-сайта Kate для ознакомления с [практическим руководством по написанию модуля Kate](#).

## Глава 7

# Режим ввода Vi

Более подробные сведения об использовании режима ввода Vi в Kate доступны в разделе «Режим ввода Vi» руководства пользователя [KatePart](#).

## Глава 8

# Команды меню

### 8.1 Меню «Файл»

#### Файл → Создать (Ctrl+N)

Создаёт новый документ. Он будет добавлен в список Документы, который находится слева на боковой панели, и назван Безымянный.

#### Файл → Открыть... (Ctrl+O)

Вызывает стандартное диалоговое окно KDE Открыть файл. Выберите нужный файл с помощью средства просмотра файлов и нажмите кнопку Открыть для его открытия.

#### Файл → Последние файлы

Это быстрый способ открыть какой-либо из последних сохранённых документов. При выборе этого пункта меню появится список последних сохранённых файлов. Щёлкните по одному из них, чтобы открыть его в это приложение (файл будет открыт, если его расположение не изменилось).

#### Файл → Открыть в

Во вложенном меню доступен список приложений, связанных с типом MIME текущего документа. Выберите один из пунктов списка, чтобы открыть файл в соответствующем приложении.

При выборе пункта Другое приложение... будет открыт диалог со списком доступных программ, что позволяет выбрать другое приложение и открыть в нём текущий файл (после этого файл также продолжит оставаться открытым в Kate).

#### Файл → Сохранить (Ctrl+S)

Сохраняет текущий документ. Если он уже был сохранён, то старый файл будет перезаписан без предупреждения. Если документ сохраняется впервые, будет открыт диалог сохранения (подробнее об этом далее).

#### Файл → Сохранить как... (Ctrl+Shift+S)

Сохраняет документ в другой файл. Для этого используется диалоговое окно со списком файлов, описание которого приведено ранее в разделе [Открыть](#).

#### Файл → Сохранить в другой кодировке...

Сохраняет документ в другой файл с другой кодировкой.

#### Файл → Сохранить копию как...

Сохраняет документ в другой файл; возможно продолжить редактировать оригинальный документ.

Файл → Сохранить все (Ctrl+L)

Сохраняет все открытые файлы.

Файл → Обновить (F5)

Обновляет текущий файл. Эта команда удобна, если файл, открытый в Kate, был изменён другой программой.

Файл → Обновить все

Обновляет все открытые файлы.

Файл → Печать... (Ctrl+P)

Открывает диалог распечатки текущего документа, позволяющий пользователю указать параметры печати.

Файл → Экспорт в HTML...

Сохраняет текущий документ в качестве файла HTML, отформатированного согласно текущим параметрам подсветки синтаксиса и цветовой схемы.

Файл → Закрывать (Ctrl+W)

Закрывает текущий файл. Если файл был изменён со времени последнего сохранения, будет предложено сохранить его перед закрытием.

Файл → Закрывать остальные

Закрывает остальные открытые документы.

Файл → Закрывать все

Закрывает все файлы, которые были открыты в Kate.

Файл → Закрывать недоступные

Закрывает все открытые документы, которые уже невозможно открыть заново из-за того, что соответствующие файлы стали недоступны.

Файл → Выход (Ctrl+Q)

Закрывает Kate и все открытые файлы. Те файлы, в которых с момента последнего сохранения производились изменения, будет предложено сохранить.

## 8.2 Меню «Правка»

Правка → Отменить действие (Ctrl+Z)

Отменяет последнюю команду редактирования (ввод, копирование, вырезание и так далее).

### Замечание

Если включён режим группировки отмены, эта команда отменяет несколько однотипных команд вроде ввода символов.

Правка → Повторить (Ctrl+Shift+Z)

Выполняет последнее отменённое действие.

Правка → Вырезать (Ctrl+X)

Удаляет текущий выделенный фрагмент и помещает его в буфер обмена. Буфер обмена работает незаметно и позволяет передавать данные между приложениями.

Правка → Копировать (Ctrl+C)

Копирует текущий выбранный текст в буфер обмена, чтобы его было возможно вставить в другое место. Буфер обмена работает незаметно и позволяет передавать данные между приложениями.

Правка → Вставить (Ctrl+V)

Вставляет первый элемент из буфера обмена в позиции курсора. Буфер обмена работает незаметно и позволяет передавать данные между приложениями.

Замечание

Если включён режим замены выделенного фрагмента, вставленный текст заменит выделенный фрагмент (если он есть).

Правка → Вставить выделенное (Ctrl+Shift+Ins)

Будет вставлено содержимое ранее выбранного **выделения мышью**. Выделите какой-либо текст указателем мыши, чтобы вставить его в текущий открытый файл с помощью этого пункта меню.

Правка → Вставить с заменой буфера обмена выделением

Выделенный текст будет заменён содержимым **буфера обмена**.

Правка → История буфера обмена

Во вложенном меню будет отображаться начало фрагментов текста, недавно скопированных в буфер обмена. Чтобы вставить один из фрагментов в текущий открытый файл, выберите соответствующий пункт в этом вложенном меню.

Правка → Копировать в формате HTML

Сохраняет текущий документ в качестве файла HTML, отформатированного согласно текущим параметрам подсветки синтаксиса и цветовой схемы.

Правка → Выделить все (Ctrl+A)

Выделяет весь текст в документе. Полезно, если требуется скопировать в другое приложение весь файл.

Правка → Снять выделение (Ctrl+Shift+A)

Снимает выделение с текста (если есть).

Правка → Блочное выделение (Ctrl+Shift+B)

Переключает режим выделения. Когда установлен режим Блок (он обозначается тестом «[Блок]» в строке состояния), возможно выполнять вертикальное выделение, например, выделить столбцы с пятого по десятый в строках 9-15.

Правка → Режимы ввода

Переключает между обычным режимом ввода и режимом Vi. В режиме ввода Vi поддерживаются наиболее часто используемые команды и движения из обычного и визуального режима vim и отображается дополнительная строка состояния режима Vi. Строка состояния показывает команды во время их ввода, вывод команд и текущий режим. Поведение этого режима возможно настроить на вкладке «Режим ввода Vi» страницы «Редактирование» диалога настройки этого приложения.

Правка → Режим замены (Ins)

Переключает между режимами вставки и замены. Если включён режим ВСТАВКА, символы будут вставляться в позиции курсора. Если включён режим ЗАМЕНА, набираемые символы будут заменять текущие, если курсор расположен перед каким-либо символом. В строке состояния показан текущий режим: ВСТАВКА или ЗАМЕНА.

### Правка → Найти... (Ctrl+F)


Открывает панель поиска в нижней части окна редактора. Слева от поля ввода текста расположена кнопка закрытия панели.

Поиск выполняется по мере ввода текста. Обнаруженные совпадения выделяются в тексте, при этом цвет фона поля ввода становится светло-зелёным. Если для поискового запроса не обнаружено совпадений, цвет фона поля ввода станет светло-красным.

Чтобы перейти к следующему или предыдущему совпадению в документе, используйте

кнопки  или .

Обнаруженные в документе совпадения останутся выделенными даже после закрытия панели поиска. Чтобы снять это выделение, нажмите клавишу Esc.

Возможно указать, следует ли учитывать регистр при поиске. Если кнопка  нажата, будет выполняться поиск только тех записей, которые соответствуют указанному регистру (верхнему или нижнему) всех символов шаблона поиска.

Чтобы перейти к расширенной панели поиска и замены, нажмите кнопку , которая расположена с правой стороны панели пошагового поиска.

### Правка → Варианты поиска → Продолжить поиск (F3)

Повторяет последний поисковый запрос (при наличии), не вызывая панель пошагового поиска; поиск начинается с текущей позиции курсора.

### Правка → Варианты поиска → Найти предыдущее (Shift+F3)

Повторяет последний поисковый запрос (при наличии), не вызывая панель пошагового поиска; поиск начинается с текущей позиции курсора и выполняется в обратном направлении.

### Правка → Варианты поиска → Следующее вхождение выделенного текста (Ctrl+H)

Находит следующее вхождение выделенного текста.

### Правка → Варианты поиска → Предыдущее вхождение выделенного текста (Ctrl+Shift+H)

Находит предыдущее вхождение выделенного текста.

### Правка → Заменить... (Ctrl+R)

Открывает панель расширенного поиска и замены. Слева от поля ввода текста расположена кнопка закрытия панели.

Режимом поиска возможно управлять с помощью следующих параметров, которые доступны в раскрывающемся списке: Обычный текст, Только полные слова, Управляющие последовательности и Регулярное выражение.


Если выбрана опция Управляющие последовательности или Регулярное выражение, в конце контекстного меню поля ввода текста будет доступен пункт Добавить.... С его помощью к шаблону поиска или замены возможно добавить управляющие последовательности или регулярные выражения из предустановленных списков.


Чтобы перейти к следующему или предыдущему совпадению в документе, используйте


кнопки  или .

Введите текст, на который следует выполнить замену, в поле Заменить на и нажмите кнопку Заменить (будет заменено только текущее выделенное вхождение) Заменить все (будут заменены все вхождения в документе).

Чтобы изменить поведение функции поиска и замены, возможно выбрать другие па-

раметры (они доступны в нижней части панели). Если кнопка  нажата, будет выполняться поиск только тех записей, которые соответствуют указанному регистру (верхнему или нижнему) всех символов шаблона поиска. Чтобы поиск и замена выполнялись только в пределах текущего выбранного фрагмента, должна быть нажата

кнопка . При нажатии кнопки Найти все в документе будут выделены все вхождения и показано их количество.

Чтобы перейти к панели пошагового поиска, нажмите кнопку , которая расположена с правой стороны панели расширенного поиска и замены.

Правка → Перейти → Перейти к парной скобке (Ctrl+6)

Перемещает курсор к парной открывающей или закрывающей скобке.

Правка → Перейти → Выделить до открывающей/закрывающей скобки (Ctrl+Shift+6)

Выделяет текст между парными открывающей и закрывающей скобками.

Правка → Перейти → Перейти к предыдущей изменённой строке

Строки, в которые были внесены изменения с момента открытия файла, называются изменёнными строками. Это действие позволяет перейти к предыдущей изменённой строке.

Правка → Перейти → Перейти к следующей изменённой строке

Строки, в которые были внесены изменения с момента открытия файла, называются изменёнными строками. Это действие позволяет перейти к следующей изменённой строке.

Правка → Перейти → Перейти на строку... (Ctrl+G)

Открывает панель строки перехода в нижней части окна и переводит курсор на указанный пользователем номер строки в документе. Номер строки возможно ввести вручную или выбрать с помощью стрелок в правой части поля ввода. Маленькая стрелка вверх увеличивает число, маленькая стрелка вниз — уменьшает. Чтобы закрыть панель, нажмите на кнопку, расположенную в её левой части.

## 8.3 Меню «Вид»

Вид → Новое окно

Открывает ещё один экземпляр Kate. Новый экземпляр будет точно таким же, как предыдущий.

Вид → Следующая вкладка (Alt+Вправо)

Активирует следующую вкладку на панели вкладок.

Вид → Предыдущая вкладка (Alt+Влево)


Активирует предыдущую вкладку на панели вкладок.

Вид → Открыть последние закрытые документы заново (Ctrl+Shift+T)

Открывает последние закрытые документы заново.

Вид → Быстрый переход (Ctrl+Alt+O)

Отображает поле ввода и список открытых файлов в области редактора. Ввод текста в поле для ввода инициирует поиск по именам и расположению документов. Для навигации по списку используйте клавиши «Вверх» и «Вниз». Чтобы переключиться на выбранный из списка документ, нажмите клавишу Enter или дважды щёлкните по документу левой кнопкой мыши. Это позволяет быстрее переключаться между документами, если их открыто очень много.

Это действие также доступно с помощью кнопки  в верхней правой части окна редактора.

Вид → Разделить окно → Предыдущая створка (Shift+F8)

Передаёт фокус предыдущей створке, если окно разделено.

Вид → Разделить окно → Следующая створка (F8)

Передаёт фокус следующей створке, если окно разделено.

Вид → Разделить окно → Открыть створку слева

Передаёт фокус левой створке (с помощью положения курсора, если требуется).

Вид → Разделить окно → Открыть створку справа

Передаёт фокус правой створке (с помощью положения курсора, если требуется).

Вид → Разделить окно → Открыть створку сверху

Передаёт фокус верхней створке (с помощью положения курсора, если требуется).

Вид → Разделить окно → Открыть створку снизу

Передаёт фокус нижней створке (с помощью положения курсора, если требуется).

Вид → Разделить окно → Разделить по вертикали (Ctrl+Shift+L)

Разделяет область (которой может быть главная область редактирования) на две части одинакового размера, при этом новая часть будет располагаться слева от текущей. Новая часть становится активной, и в ней отображается тот же документ.

Смотрите также раздел [Работа с многодокументным интерфейсом Kate](#).

Вид → Разделить окно → Разделить по горизонтали (Ctrl+Shift+T)

Разделяет область (которой может быть главная область редактирования) на две части одинакового размера, при этом новая часть будет располагаться под текущей. Новая часть становится активной, и в ней отображается тот же документ.

Смотрите также раздел [Работа с многодокументным интерфейсом Kate](#)

Вид → Разделить окно → Разделить по вертикали и переместить документ

Разделить текущую створку окна на две вертикальных и расположить активный документ в правой части.

Смотрите также раздел [Работа с многодокументным интерфейсом Kate](#).

Вид → Разделить окно → Разделить по горизонтали и переместить документ

Разделить текущую створку окна на две горизонтальных и расположить активный документ в нижней части.

Смотрите также раздел [Работа с многодокументным интерфейсом Kate](#)

Вид → Разделить окно → Сменить ориентацию створок

Переключает между горизонтальным и вертикальным разделением.

Вид → Разделить окно → Закрыть текущую створку (Ctrl+Shift+R)

Закрывает текущую часть. Эта опция недоступна, если область редактирования не разделена на части.

При закрытии части документ, который отображается в ней, не закрывается: он всё ещё будет доступен в меню «Вид» и в списке файлов.

Смотрите также раздел [Работа с многодокументным интерфейсом Kate](#)

Вид → Разделить окно → Закрыть неактивные створки

Закрывает все части, кроме активной (с мигающим курсором). Эта опция недоступна, если область редактирования не разделена на части.

При закрытии части документ, который отображается в ней, не закрывается: он всё ещё будет доступен в меню «Вид» и в списке файлов.

Вид → Разделить окно → Скрыть неактивные створки

Скрывает все створки, кроме активной.

Вид → Разделить окно → Переместить разделитель влево

Если включён режим деления окна, эта функция переместит границу между двумя вертикально разделёнными документами левее.

Вид → Разделить окно → Переместить разделитель вправо

Если включён режим деления окна, эта функция переместит границу между двумя вертикально разделёнными документами правее.


Вид → Разделить окно → Переместить разделитель вверх

Если включён режим разделения окна, эта функция переместит границу между двумя горизонтально разделёнными документами выше.

Вид → Разделить окно → Переместить разделитель вниз

Если включён режим разделения окна, эта функция переместит границу между двумя горизонтально разделёнными документами ниже.

#### Замечание

Некоторые опции из меню Вид/Разделить окно также доступны с помощью кнопки  в верхнем правом углу окна редактора.

Вид → Боковые панели

Вид → Боковые панели → Показать боковые панели (Ctrl+Alt+Shift+F)

Включает или отключает отображение боковых панелей. Команда не действует на средства управления содержимым боковых панелей, они остаются видимыми, и, конечно, продолжают действовать заданные для них комбинации клавиши (если имеются).

Вид → Боковые панели → Показать панель модуля

Список всех подключённых модулей. Чтобы отобразить или скрыть боковую панель того или иного модуля, установите или снимите флажок напротив соответствующего элемента.

Вид → Командная строка (F7)

Перемещает фокус ввода из области редактирования на [командную строку](#) и обратно.

Вид → Увеличить размер шрифта (Ctrl++)

Увеличивает размер шрифта текста документа.

Вид → Уменьшить размер шрифта (Ctrl+-)

Уменьшает размер шрифта текста документа.

Вид → Схема

Это меню содержит доступные схемы. Здесь возможно изменить схему для текущего документа. Чтобы изменить схему по умолчанию, откройте [страницу «Шрифты и цвета»](#) диалога настройки.

Вид → Перенос строк → Динамический перенос строк (F10)

Включает или отключает динамический перенос строк текущего документа. При динамическом переносе текст в окне возможно просматривать без горизонтального прокручивания, то есть строка, непрерывная на самом деле, для удобства делится на несколько.

Вид → Перенос строк → Маркеры динамического переноса строк

Позволяет выбрать, когда и как должны показываться маркеры динамического переноса строк. Список доступен только в том случае, если включён параметр Динамический перенос строк.

Вид → Перенос строк → Маркеры статического переноса строк

Включает или отключает отображение вертикальной линии, указывающей позицию переноса строк, заданную в [диалоге настройки](#). Эта функция доступна только для моноширинных шрифтов.

Вид → Границы → Полоса закладок (F6)

Это переключаемый элемент. Если флажок установлен, в левой части активной области редактирования будет показана полоса закладок, которая обозначает позиции отмеченных в редакторе строк.

Вид → Границы → Номера строк (F11)

Позволяет показать или скрыть номера строк слева от текста.

Вид → Границы → Отметки на полосе прокрутки

Включает или отключает отображение закладок (и других отметок) на полосе вертикальной прокрутки. Если флажок установлен, на полосе прокрутки отображаются тонкие линии, окрашенные в цвет отметки. Щёлкните по такой линии средней кнопкой мыши, и в области просмотра будет показано место отметки.

Вид → Границы → Мини-карта в полосе прокрутки

Заменяет полосу прокрутки визуализацией текущего документа. Более подробные сведения доступны в разделе «[Мини-карта в полосе прокрутки](#)» руководства пользователя [KatePart](#).

Вид → Сворачивание блоков кода

Эти параметры позволяют настроить [сворачивание блоков кода](#):

Полоса сворачивания блоков (F9)

Позволяет включать и отключать отображение полосы сворачивания блоков в левой части области просмотра.

Свернуть текущий блок кода

Сворачивает ближайший к курсору блок.

Развернуть текущий блок кода

Разворачивает ближайший к курсору блок.

Свернуть блоки 1-го уровня (Ctrl+Shift+-)

Сворачивает все блоки верхнего уровня в документе. Чтобы развернуть блоки верхнего уровня, нажмите треугольник, который указывает вправо.

Развернуть блоки 1-го уровня (Ctrl+Shift++)

Разворачивает все блоки верхнего уровня в документе.

Показать непечатаемые пробелы

Показывает или скрывает рамку вокруг непечатаемых пробельных символов.

## 8.4 Меню «Закладки»

Под пунктами меню, описание которых приводится далее, доступны пункты самих закладок. Текст каждого такого пункта — это номер и несколько символов строки, на которую указывает эта закладка. При выборе пункта меню закладки курсор перейдёт на начало отмеченной строки. Редактор автоматически выполнит прокрутку текста, если эта строка не видна.

Закладки → Поставить закладку (Ctrl+B)

Устанавливает или удаляет закладку для текущей строки активного документа (если закладки не было, она устанавливается, если была — удаляется).

Закладки → Убрать все закладки

Убирает все закладки из документа и из списка, который располагается в конце этого меню.

Закладки → Предыдущая (Alt+Page Up)

Перемещает курсор к ближайшей предыдущей закладке. Пункт меню будет содержать номер строки и начало текста на ней. Этот пункт доступен, только если выше курсора есть строка с закладкой.

Закладки → Следующая (Alt+Page Down)

Перемещает курсор к ближайшей следующей закладке. Пункт меню будет содержать номер строки и начало текста на ней. Этот пункт доступен, только если ниже курсора есть строка с закладкой.

## 8.5 Меню «Сеанс»

С помощью пунктов этого меню возможно использовать сеансы и управлять ими. Более подробные сведения доступны в разделе [Работа с сеансами](#).

Сеанс → Создать

Создаёт новый сеанс. Все текущие файлы будут закрыты.

Сеанс → Открыть сеанс

Открывает один из существующих сеансов, для выбора предоставляется диалоговое окно.

Сеанс → Быстрое открытие

Это вложенное меню позволяет быстро открыть существующий сеанс.

Сеанс → Сохранить сеанс

Сохраняет текущий сеанс. Если сеанс безымянный, будет предложено назвать его.

Сеанс → Сохранить сеанс как...

Сохраняет текущий сеанс под именем, которое будет предложено ввести.

Сеанс → Управление сеансами...

Отображает диалог управления сеансами, с помощью которого возможно их переименовывать и удалять.

## 8.6 Меню «Сервис»

Сервис → Заблокировать изменение текста

Устанавливает режим «только для чтения» для текущего документа, тем самым предохраняя его от любых возможных изменений.

Сервис → Тип документа

Позволяет выбрать тип файла для текущего документа. Переопределяет глобальный [тип файла](#), указанный с помощью пункта меню [Настройка → Настроить Kate...](#) на вкладке «Типы файлов», но только для текущего документа.

Сервис → Подсветка

Позволяет выбрать нестандартную схему подсветки для текущего документа. Переопределяет глобальный режим подсветки, указанный с помощью пункта меню [Настройка → Настроить редактор...](#), но только для текущего документа.

Сервис → Расстановка отступов

Позволяет выбрать [режим отступов](#) для текущего документа. Переопределяет глобальный режим отступов, указанный с помощью пункта меню [Настройка → Настроить редактор...](#), но только для текущего документа.

Сервис → Кодировка

Позволяет выбрать для текущего документа кодировку, отличную от указанной с помощью пункта меню [Настройка → Настроить редактор...](#) на вкладке [Открытие и сохранение](#).

Сервис → Конец строки

Позволяет выбрать режим конца строки для текущего документа. Переопределяет глобальный режим конца строки, заданный с помощью пункта меню *Настройка* → *Настроить редактор...*, но только для текущего документа.

Сервис → Добавлять отметку о порядке байтов (BOM)

Позволяет явным образом добавить маркер последовательности байтов в документах, закодированных в формате Юникод. Маркер последовательности байтов — это символ из стандарта Юникод, используемый для обозначения порядка байтов в текстовом файле или потоке. Дополнительные сведения: [Маркер последовательности байтов](#).

Сервис → Сценарии

Это вложенное меню содержит список сценариев. Возможно также [написать собственные сценарии](#). Это позволяет расширить функционал Kate инструментами, определёнными пользователем.

В документации [KatePart](#) есть полный список сценариев.

Сервис → Дополнить код (Ctrl+Пробел)

Вручную вызывает функцию дополнения кода (обычно это выполняется с помощью соответствующей комбинации клавиш).

Сервис → Автодополнение слов

Опции *Использовать нижестоящее слово* (Ctrl+9) и *Использовать вышестоящее слово* (Ctrl+8) автоматически дополняют введённый текст посредством поиска похожих слов выше или ниже позиции курсора. Автодополнение как в командной оболочке выводит на экран всплывающее окно автодополнения с совпадающими записями.

Сервис → Орфография → Автоматическая проверка орфографии (Ctrl+Shift+O)

При включённой опции *Автоматическая проверка орфографии* неправильно написанное слово будет подчёркнуто сразу же.

Сервис → Орфография → Проверка орфографии...

Запускает программу проверки орфографии, которая позволяет пользователю найти и исправить орфографические ошибки.

Более подробные сведения о программе проверки орфографии KDE доступны в разделе [Проверка орфографии в Основах KDE](#).

Сервис → Орфография → Проверка орфографии от курсора...

Запускает проверку правописания, но начинается она от позиции курсора, а не от начала документа.

Сервис → Орфография → Проверка орфографии в выделенном фрагменте...

Проверяет орфографию в выделенном фрагменте текста.

Сервис → Орфография → Выбрать словарь...

Отображает в нижней части окна редактора поле с раскрывающимся списком всех доступных словарей для проверки орфографии. Это поле позволяет легко переключаться между орфографическими словарями, например, для автоматической проверки орфографии в тексте на разных языках.

## 8.7 Меню «Настройка» и «Справка»

В приложении редактора Kate присутствуют стандартные меню KDE: *Настройка* и *Справка*. Более подробно они описаны в соответствующих разделах «Основ KDE» ([Меню «Настройка»](#) и [Меню «Справка»](#)). В редакторе эти меню содержат следующие дополнительные пункты:

Настройка → Цветовая схема

Возможно использовать отличную от системной цветовую схему (описание системных цветовых схем доступно в разделе [Цвета](#) справки к приложению «Параметры системы»).

Настройка → Показать заголовки вкладок

Вкладки возможно перемещать с помощью левой кнопки мыши, а также взаимодействовать с ними с помощью контекстного меню (для закрытия, копирования пути в буфер обмена или открытия расположения документа с помощью диспетчера файлов). Кнопка [Быстрый переход](#) позволяет легко переключаться между документами. Кнопка



вызывает контекстное меню из панели инструментов ВидРазделить окно

Настройка → Показывать путь в заголовке окна

Если этот параметр включён, отображается полный путь к активному документу, в противном случае — только имя файла. Если редактируются несколько файлов с одним и тем же именем, их будет возможно различать.

## Глава 9

# Настройка Kate

Anders Lund

Перевод на русский язык: Андрей Балагута

Перевод на русский: Алексей Опарин

Обновление перевода: Мария Шикунова

Обновление и редактирование перевода: Олеся Герасименко

### 9.1 Обзор

Kate предоставляет несколько средств настройки. Наиболее важные из них:

#### Диалог настройки

Главный инструмент настройки приложения Kate, редактора и дополнительных модулей.

#### Меню Настройка

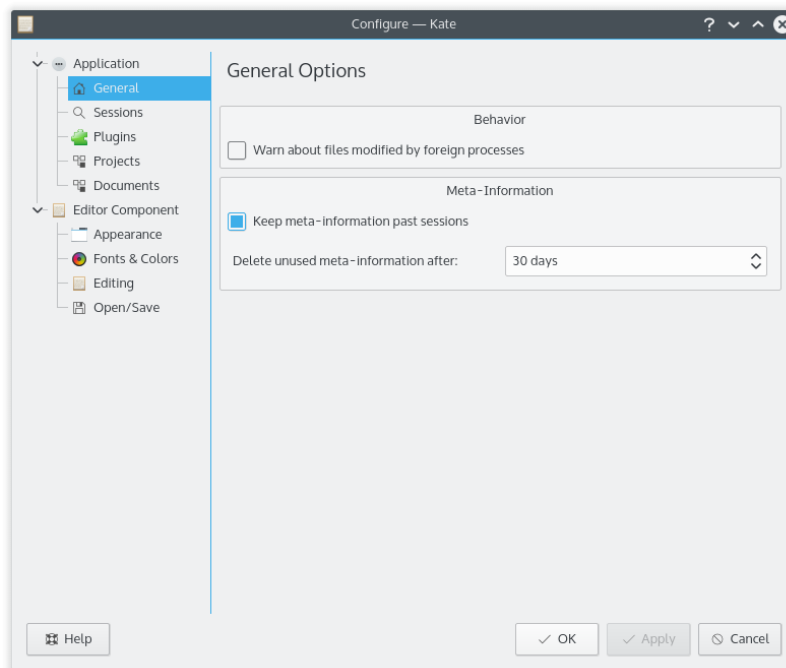
Позволяет настраивать часто используемые параметры и открывать диалоги настройки.

#### Меню Вид

Позволяет разделить текущее окно, включить/отключить отображение панелей значков и номеров строк для редактируемого документа.

Поведение встроенного терминала определяется его настройкой в параметрах системы и может быть изменено с помощью контекстного меню.

## 9.2 Основной диалог настройки



В диалоге настройки слева отображаются разделы, организованные в древовидную структуру, а справа — страница настройки, соответствующая выбранному разделу.

Параметры настройки делятся на две основные группы:

- [Настройка приложения](#)
- [Настройка компонента редактора](#)

## 9.3 Настройка приложения Kate

Эта группа содержит страницы общей настройки приложения Kate

### 9.3.1 Общие

Здесь возможно настроить несколько основных параметров Kate

#### Поведение

Открыть каждый документ в отдельном окне

При включении этого параметра каждый документ будет открываться в отдельном окне. Если параметр не включён, то все документы будут открываться в отдельных вкладках внутри текущего окна.

Переход к представлению вывода при получении сообщения следующего типа

Позволяет указать, когда Kate следует показывать панель вывода.

Доступные варианты: Никогда, Ошибка (при ошибке), Предупреждение (при предупреждении или ошибке), Информация (при информационном сообщении, предупреждении или ошибке) и Журнал (при записи журнала, информационном сообщении, предупреждении или ошибке).

## Руководство пользователя Kate

Использовать отдельный диалог для обработки файлов, изменённых внешней программой

Если флажок установлен, Kate (в модальном диалоговом окне) уведомит об изменении файлов другой программой, как только главное окно получит фокус ввода. Возможно производить обновления, сохранять или отменять изменения сразу для группы файлов.

Если флажок снят, Kate будет выводить отдельный запрос для каждого изменённого файла, как только представление этого файла получит фокус ввода.

### Быстрый переход

Режим сопоставления

Настройка списка для инструмента «**Быстрый переход**». Файлы могут быть сопоставлены по имени или пути.

Содержимое списка

Настройка списка для инструмента «**Быстрый переход**». Доступны следующие варианты: Файлы текущего проекта и Файлы всех проектов.

### Вкладки

Ограничение количества вкладок

Настройка максимального числа вкладок. Если ограничивать количество не требуется, выберите вариант Без ограничений.

Автоматически скрывать вкладки

Скрывать вкладки, если открыт только один документ.

Показывать кнопку закрытия

Если этот параметр отмечен флажком, на каждой вкладке будет отображаться кнопка закрытия.

Расширить вкладки

Если флажок установлен, вкладки будут развёрнуты на максимально возможную величину.

Создать новый документ двойным щелчком

Если флажок установлен, при двойном щелчке открывается новый документ.

Закрытие документа средней кнопкой мыши

Если флажок установлен, нажатие средней кнопки мыши закрывает документ.

Разрешить прокрутку вкладок

Разрешить прокрутку вкладок колесом мыши.

Сокращать текст вкладок

Текст вкладок может быть сокращён, если его длина слишком велика.

### Вкладки

Действие при нажатии кнопки «назад»

Позволяет указать, какой пункт меню выбирается при нажатии кнопки мыши «назад»: Предыдущая вкладка или Перейти на предыдущую запись в журнале.

Действие при нажатии кнопки «вперёд»

Позволяет указать, какой пункт меню выбирается при нажатии кнопки мыши «вперёд»: Следующая вкладка или Перейти на следующую запись в журнале.

## 9.3.2 Сеансы

Раздел содержит параметры, связанные с [использованием сеансов](#).

Поведение при запуске приложения

Определяет поведение Kate при запуске. Этот параметр возможно переопределить, указав другое значение с помощью [командной строки](#).

## Руководство пользователя Kate

### Начать новый сеанс

Если установить этот флажок, Kate при каждом запуске будет начинать новый безымянный сеанс.

### Загрузить последний сеанс

При запуске Kate будет открывать прошлый сеанс. Удобно, если не требуется изменять параметры настройки.

### Выбрать сеанс

Kate будет отображать диалог выбора одного из сохранённых сеансов (или загружать сеанс по умолчанию, если сохранённые сеансы отсутствуют). Это поведение по умолчанию. Удобно для пользователей, которые часто работают с разными сеансами.

### Поведение при запуске и завершении работы

Выберите поведение Kate при завершении работы приложения. Возможно указать, что Kate следует Автоматически сохранять и восстанавливать.

### Новые несохранённые файлы

Если флажок установлен, Kate будет автоматически сохранять все новые несохранённые файлы.

### Несохранённые изменённые файлы

При выборе этого пункта Kate будет автоматически сохранять все файлы с несохранёнными изменениями при завершении работы приложения.

### Закрывать Kate после закрытия всех открытых файлов

Если флажок установлен, Kate прекратит работу после того, как будет закрыт последний редактируемый файл. Если флажок снят, откроется пустая страница, что позволит сразу создать новый файл.

### Состав сеанса

#### Восстанавливать положение и размеры окна

Если флажок установлен, Kate будет восстанавливать конфигурацию окна в каждом новом сеансе.

#### Сохранять метаинформацию сеанса

Если флажок установлен, Kate сохранит метаданные (закладки, конфигурацию сеанса) даже при закрытии документа пользователем. Эти данные будут использованы, если документ вновь будет открыт в прежнем виде.

#### Удалять неиспользуемые данные через

Устанавливает максимальное количество дней, в течение которого будут храниться метаданные ранее открытых файлов. Долговременное хранение приводит к избыточному увеличению размера базы данных.

Любые изменения в данных сеанса (открытые файлы, параметры окна) всегда будут сохраняться.

### 9.3.3 Модули

На этой странице представлены установленные модули Kate. Для каждого отображается название и краткое описание. Для подключения модуля требуется установить соответствующий флажок.

Если модуль настраиваемый, к параметрам его настройки будет возможно перейти с помощью соответствующего дочернего элемента на этой странице.

Более подробные сведения о модулях доступны в разделе Глава 4.

### 9.3.4 Настройка компонента редактирования

Более подробные сведения об этой части диалога конфигурации доступны в разделе «Настройка компонента редактирования» руководства пользователя KatePart.

### 9.3.5 Настройка с помощью переменных документа

Более подробные сведения об этой части диалога конфигурации доступны в разделе «[Настройка с помощью переменных документа](#)» руководства пользователя KatePart.

## Глава 10

# Авторские права и лицензия

Kate. Авторские права на программу: (с) Команда разработчиков Kate, 2000, 2001, 2002—2005  
Команда Kate:

Christoph Cullmann [cullmann@kde.org](mailto:cullmann@kde.org)

Координатор проекта и ключевой разработчик

Anders Lund [anders@alweb.dk](mailto:anders@alweb.dk)

Ключевой разработчик, подсветка синтаксиса для языка Perl, документация

Joseph Wenninger [kde@jowenn.at](mailto:kde@jowenn.at)

Ключевой разработчик, подсветка синтаксиса

Michael Bartl [michael.bartl1@chello.at](mailto:michael.bartl1@chello.at)

Ключевой разработчик

Philip [phlip\\_cpp@my-deja.com](mailto:phlip_cpp@my-deja.com)

Компилятор проекта

Waldo Bastian [bastian@kde.org](mailto:bastian@kde.org)

Отличная система буферизации

Matt Newell [newellm@proaxis.com](mailto:newellm@proaxis.com)

Тестирование...

Michael McCallum [gholam@xtra.co.nz](mailto:gholam@xtra.co.nz)

Ключевой разработчик

Jochen Wilhemly [digisnap@cs.tu-berlin.de](mailto:digisnap@cs.tu-berlin.de)

Автор KWrite

Michael Koch [koch@kde.org](mailto:koch@kde.org)

Перенос KWrite на KParts

Christian Gebauer [gebauer@bigfoot.com](mailto:gebauer@bigfoot.com)

Точно не определено

Simon Hausmann [hausmann@kde.org](mailto:hausmann@kde.org)

Точно не определено

Glen Parker [glenebob@nmlink.com](mailto:glenebob@nmlink.com)

Журнал действий KWrite, интеграция KSpell

## Руководство пользователя Kate

Scott Manson [sdmanson@alltel.net](mailto:sdmanson@alltel.net)  
Подсветка синтаксиса XML в KWrite

John Firebaugh [jfirebaugh@kde.org](mailto:jfirebaugh@kde.org)  
Исправления и прочее

Dominik Haumann [dhdev@gmx.de](mailto:dhdev@gmx.de)  
Разработчик, мастер подсветки синтаксиса

Много других людей внесли свой вклад:

Matteo Merli [merlim@libero.it](mailto:merlim@libero.it)  
Подсветка для спес-файлов RPM, diff и прочее

Rocky Scaletta [rocky@purdue.edu](mailto:rocky@purdue.edu)  
Подсветка для VHDL

Юрий Лебедев  
Подсветка для SQL

Chris Ross  
Подсветка для Ferite

Nick Roux  
Подсветка для ILERPG

John Firebaugh  
Подсветка для Java™ и прочее

Carsten Niehaus  
Подсветка для L<sup>A</sup>T<sub>E</sub>X

Per Wigren  
Подсветка файлов сборки, а также для Python

Jan Fritz  
Подсветка для Python

Daniel Naber  
Исправление ошибок, модуль XML

Авторские права на документацию: (c) Seth Rothberg [sethmr@bellatlantic.org](mailto:sethmr@bellatlantic.org), 2000—2001

Авторские права на документацию: (c) Anders Lund [anders@alweb.dk](mailto:anders@alweb.dk), 2002, 2003, 2005

Перевод на русский язык: Андрей Балагута [uj2@mail.ru](mailto:uj2@mail.ru)

Редакция и обновление перевода: Николай Шафоростов [shafff@ukr.net](mailto:shafff@ukr.net)

Обновление перевода: Алексей Опарин [opaleksej@yandex.ru](mailto:opaleksej@yandex.ru)

Обновление перевода: Мария Шикунова [translation-team@basealt.ru](mailto:translation-team@basealt.ru)

Редактирование перевода: Олеся Герасименко [translation-team@basealt.ru](mailto:translation-team@basealt.ru)

Этот документ распространяется на условиях [GNU Free Documentation License](#).

Программа распространяется на условиях лицензии [GNU General Public License](#).

## Приложение А

# Регулярные выражения

Более подробные сведения об основах работы с регулярными выражениями доступны в [приложении «Регулярные выражения»](#) руководства пользователя KatePart.

## Приложение В

# Предметный указатель

Эмулятор терминала, 75  
Главное окно, 14  
Область редактирования, 14  
Список документов, 38  
конфигурация  
настройка  
параметры, 97