

Підручник з Kate

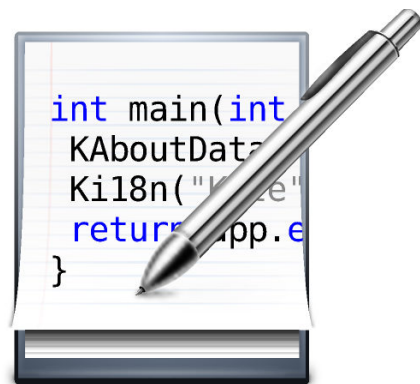
Anders Lund

Seth Rothberg

Dominik Haumann

T.C. Hollingsworth

Переклад українською: Юрій Чорноіван



Підручник з Kate

Зміст

1	Вступ	10
2	Основи	11
2.1	Як запустити Kate	11
2.1.1	За допомогою меню	11
2.1.2	За допомогою командного рядка	11
2.1.2.1	Параметри командного рядка	12
2.1.3	Перетягування і скидання	13
2.2	Робота у Kate	13
2.2.1	Швидкий запуск	13
2.2.2	Клавіатурні скорочення	13
2.3	Робота з багатодокументним інтерфейсом (MDI) Kate	14
2.3.1	Огляд	14
2.3.1.1	Головне вікно	14
2.3.2	Область редактора	15
2.4	Користування сеансами	15
2.5	Швидке відкриття	16
2.5.1	Користування «Швидким відкриттям»	16
2.5.2	Налаштовування «Швидкого відкриття»	16
2.6	Як отримати довідку	17
2.6.1	За допомогою Kate	17
2.6.2	З вашими текстовими файлами	17
2.6.3	Статті щодо Kate	17
3	Робота у редакторі Kate	18
4	Робота з додатками	19
4.1	Додатки програми Kate	19
4.2	Зовнішні інструменти	20
4.2.1	Налаштовування зовнішніх інструментів	21
4.2.2	Розгортання змінних	22
4.2.3	Список типових інструментів	24
4.3	Додаток перегляду зворотного трасування	27

4.3.1	Користування додатком перегляду зворотного трасування	27
4.3.2	Налаштування	27
4.4	Додаток для збирання	28
4.4.1	Вступ	28
4.4.2	Користування додатком збирання	28
4.4.2.1	Вкладка «Параметри цілі»	29
4.4.2.2	Вкладка «Вивід»	30
4.4.3	Структура меню	30
4.4.4	Подяки	31
4.5	Додаток закриття за критерієм	31
4.5.1	Вступ	31
4.5.2	Користування додатком закриття за критерієм	31
4.5.3	Структура меню	31
4.6	Додаток піпетки	32
4.6.1	Вступ	32
4.6.2	Налаштування	32
4.7	Кольорові дужки	32
4.7.1	Вступ	32
4.7.2	Налаштування	32
4.8	Додаток CTags	33
4.8.1	Вступ	33
4.8.2	Налаштування	33
4.8.2.1	Загальний показчик	33
4.8.2.2	Показчик сеансу	34
4.8.3	Користування додатком CTags	34
4.8.4	Структура меню	34
4.9	Додаток попереднього перегляду документів	35
4.9.1	Вступ	35
4.9.2	Структура меню	35
4.9.3	Інтерфейс	35
4.10	Додаток перемикання документів	35
4.10.1	Структура меню	35
4.11	Навігатор файловою системою	36
4.11.1	Структура меню	36
4.11.2	Інтерфейс	36
4.11.3	Налаштування	38
4.12	Список документів	38
4.12.1	Вступ	38
4.12.2	Структура меню	39
4.12.3	Налаштування	39
4.13	Додаток GDB	40
4.13.1	Вступ	40

4.13.2	Структура меню і панелі інструментів	40
4.13.3	Панель налагоджування	41
4.13.4	Стек викликів і локальні змінні	42
4.13.5	Подяки	43
4.13.6	Налаштування	43
4.13.6.1	Налаштовування середовища виконання	45
4.14	Додаток проєктів	46
4.14.1	Вступ	46
4.14.2	Структурований перегляд файлів	46
4.14.3	Перемикання проєктів	47
4.14.4	Пошук і заміна у проєктах	47
4.14.5	Просте автоматичне доповнення	48
4.14.6	Підтримка збирання проєкту	49
4.14.7	Створення проєктів	49
4.14.7.1	Автоматичне завантаження проєктів	49
4.14.7.2	Створення проєктів вручну	50
4.14.8	Поточний проєкт	52
4.14.9	Меню «Проєкти»	52
4.15	Додаток клієнта LSP	53
4.15.1	Структура меню	53
4.15.2	Підтримка переходу до символів	54
4.15.2.1	Налаштовування схеми символів клієнта LSP	55
4.15.2.2	Підтримка переходу до загальних символів	55
4.15.3	Інші можливості	55
4.15.4	Налаштування	55
4.15.4.1	Налаштування сервера LSP	58
4.15.4.2	Форматування при збереженні для сервера LSP	58
4.15.4.3	Придушення діагностики на сервері LSP	59
4.15.4.4	Діагностика сервера LSP	59
4.15.4.5	Налаштовування середовища виконання	59
4.16	Пошук з заміною	62
4.16.1	Вступ	62
4.16.2	Інтерфейс	62
4.16.2.1	Запит з пошуку	62
4.16.2.2	Параметри пошуку у теці	63
4.16.2.3	Результати пошуку	65
4.16.3	Структура меню	65
4.17	Скрипти і фрагменти тексту Kate	65
4.17.1	Вступ	65
4.17.2	Структура меню	66
4.17.3	Панель фрагментів	66
4.17.3.1	Завантаження файлів сховища фрагментів	66

4.17.3.2	Створення і редагування сховищ фрагментів	66
4.17.3.3	Створення і редагування фрагментів	68
4.17.4	Користування фрагментами	71
4.17.5	Подяки	71
4.18	Додаток клавіатурних макросів	71
4.18.1	Вступ	71
4.18.2	Основні прийоми користування	72
4.18.2.1	Розпочати записування клавіатурного макроса:	72
4.18.2.2	Щоб завершити записування, скористайтеся пунктом меню: . .	72
4.18.2.3	Щоб скасувати записування, скористайтеся пунктом меню: . .	72
4.18.2.4	Щоб відтворити поточний макрос, скористайтеся пунктом меню:	72
4.18.3	Іменовані макроси	72
4.18.3.1	Щоб зберегти поточний макрос, скористайтеся пунктом меню:	72
4.18.3.2	Щоб завантажити збережений макрос як поточний, скористай- теся пунктом меню:	72
4.18.3.3	Щоб відтворити збережений макрос без його завантаження, скористайтеся пунктом меню:	73
4.18.3.4	Щоб витерти (тобто вилучити) збережений макрос, скористай- теся пунктом меню:	73
4.18.3.5	Підказки щодо команд:	73
4.18.4	Обмеження	73
4.19	Додаток SQL	73
4.19.1	Вступ	73
4.19.2	Встановлення з'єднання з базою даних	74
4.19.3	Виконання запитів	74
4.19.3.1	INSERT/DELETE/UPDATE	74
4.19.3.2	SELECT	74
4.19.4	Навігація	75
4.19.5	Структура меню	75
4.19.6	Подяки	76
4.20	Додаток перегляду символів	76
4.20.1	Користування додатком закриття за критерієм	76
4.20.2	Структура меню	76
4.20.3	Налаштування	76
4.21	Додаток перегляду інструмента термінала	77
4.21.1	Структура меню	77
4.21.2	Налаштування	77
4.22	Додаток для фільтрування тексту	78
4.22.1	Користування додатком фільтрування тексту	78
4.22.2	Структура меню	78
4.23	Перевірка XML	79
4.23.1	Структура меню	79
4.23.2	Подяки	79

4.24	Доповнення XML	79
4.24.1	Як користуватися	80
4.24.2	Можливості і обмеження	80
4.24.3	Структура меню	80
4.24.4	Подяки	81
4.25	Додаток навігації результатами збирання	81
4.25.1	Користування	81
4.26	Додаток форматування	82
4.26.1	Користування	82
4.26.2	Підтримувані мови і засоби форматування	82
4.26.3	Налаштовування	83
4.26.4	Тимчасове вимикання форматування при збереженні	84
5	Додаткові інструменти редагування	85
6	Розширення можливостей Kate	86
6.1	Вступ	86
6.2	Як працювати з підсвічуванням синтаксису	86
6.3	Створення скриптів мовою JavaScript	86
6.4	Додатки до програми Kate (C++)	86
7	Режим введення VI	87
8	Пункти меню	88
8.1	Меню «Файл»	88
8.2	Меню «Зміни»	89
8.3	Меню «Перегляд»	92
8.4	Меню «Закладки»	96
8.5	Меню «Сеанси»	96
8.6	Меню «Інструменти»	97
8.7	Меню «Параметри» і «Довідка»	98
9	Як налаштувати Kate	100
9.1	Огляд	100
9.2	Основне діалогове вікно налаштування	101
9.3	Налаштування програми Kate	101
9.3.1	Загальне	101
9.3.2	Сеанс	103
9.3.3	Додатки	104
9.3.4	Налаштування компонента редактора	104
9.3.5	Налаштування змінних документа	104
10	Подяки і ліцензія	105
A	Формальні вирази	107
B	Предметний покажчик	108

Перелік таблиць

4.1 Деякі доступні додатки KParts	35
---	----

Анотація

Kate — це текстовий редактор для програмістів, створений командою розробників KDE.

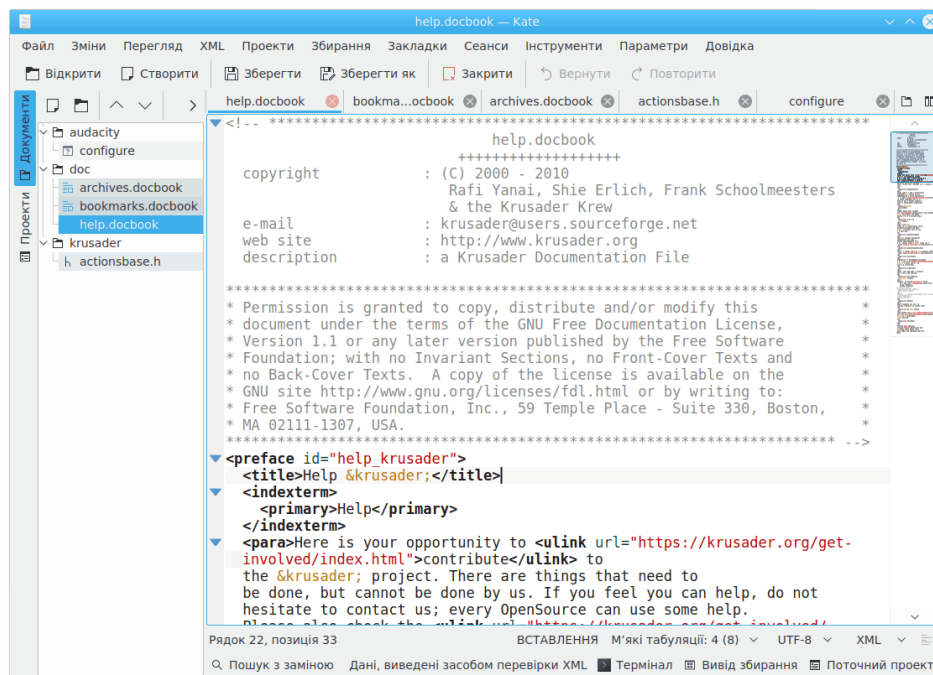
У цьому підручнику описано Kate версії 22.08

Розділ 1

Вступ

Ласкаво просимо до Kate, текстового редактора для програмістів, створеного розробниками KDE. Серед багатьох можливостей Kate придатне для налаштування підсвічування синтаксису для мов програмування від C і C++ до HTML і скриптів bash, можливість створення і підтримки проєктів, багатодокументний інтерфейс (MDI) та вбудований емулятор терміналу.

Але Kate є не лише редактором для програмістів. Закладена у програму можливість відкриття одразу декількох файлів робить її ідеальною для редагування багатьох файлів налаштування UNIX®. Наприклад, цей документ було написано у Kate.



Редагування цього підручника...

Розділ 2

ОСНОВИ

Переклад українською: Юрій Чорноіван

Якщо ви вже колись користувалися текстовим редактором, у вас не буде проблем з користуванням Kate. У наступних двох розділах, [Запуск Kate](#) і [Робота у Kate](#), ви ознайомитеся з усіма відомостями, потрібними вам для того, щоб якомога швидше розпочати роботу.

2.1 Як запуснути Kate

Ви можете запуснути Kate за допомогою засобу запуску програм або командного рядка.

2.1.1 За допомогою меню

Відкрийте меню програм KDE натисканням великої кнопки засобу запуску програм. Після натискання буде відкрито меню. Пересуньте вказівник у верхню частину меню до пункту **Програми** → **Інструменти** → **Текстовий редактор з додатковими можливостями Kate**.

2.1.2 За допомогою командного рядка

Ви можете запуснути Kate виконанням відповідної команди у командному рядку. Якщо ви вкажете як параметр команди назву файла, як у наведеному нижче прикладі, Kate створить або відкриє вказаний вами файл.

```
%kate мій_файл.txt
```

Якщо на вашому комп'ютері задіяно з'єднання з мережею, ви можете скористатися перевагами мережевої прозорості у KDE для того, щоб відкривати файли з мережі Інтернет.

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

Щоб змінити каталог для тимчасових файлів, типовою адресою якого є /tmp, встановіть відповідне значення змінної середовища TMPDIR до запуску Kate. Приклад:

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

2.1.2.1 Параметри командного рядка

Kate сприймає такі параметри командного рядка:

kate --help

Ця команда покаже список параметрів, доступ до яких можна отримати з командного рядка.

kate -s --start *назва*

Запускає Kate з відкритим сеансом *назва*. Якщо цього сеансу ще не існує, його буде створено. Якщо вказаний сеанс вже відкрито у іншому екземплярі Kate, вказані у параметрах файли буде завантажено у вже відкритий сеанс.

kate -p --pid PID

Використовувати лише екземпляр з вказаним PID (ідентифікатором процесу).

kate -e --encoding *кодування URL*

Використовувати вказане кодування для документа.

kate -l --line *рядок URL*

Переводить курсор у вказаний рядок після відкриття документа.

kate -c --column *стовпчик URL*

Переводить курсор у вказаний стовпчик після відкриття документа.

kate -i --stdin

Читає вміст документа з STDIN. Цей параметр подібний до звичайного параметра -, який використовується у багатьох програмах, які працюють за допомогою командного рядка, цей параметр надає вам змогу передати результати роботи певної команди до Kate.

kate --startanon

Запустити Kate у новому анонімному сеансі з параметром **-n**

kate -n --new

Примусово запустити новий екземпляр Kate (буде проігноровано, якщо було використано параметр **start**, а вказаний сеанс Kate уже було відкрито). Буде використано, якщо ви не вкажете ніяких параметрів і ніяких адрес URL.

kate -b --block

За використання вже запущеного екземпляра Kate, заблокувати дії у ньому до завершення його роботи, якщо надійшов наказ відкрити певні адреси URL.

Програмою Kate, запущеною з цим параметром, можна скористатися як редактором для введення повідомлень щодо внесків до систем керування версіями, зокрема Git або Subversion. Ці системи блокують редактор до завершення введення вами повідомлення, оскільки згодом відкривають тимчасовий файл, який буде порожнім, якщо буде виконано негайне повернення даних з Kate до програми, якою було викликано редактор.

Цей параметр потрібен для КІО (модулів введення-виведення даних KDE), якщо ви відкриваєте файл на віддаленому комп'ютері (який було тимчасово отримано до вашої системи), який згодом має бути вивантажено на його основне місце зберігання під час виконання дії зі збереження.

kate --tempfile

Якщо використано цей параметр, вказані файли вважатимуться тимчасовими і вилучатимуться (якщо ці файли зберігаються на вашому комп'ютері і ваш користувач має потрібні дозволи на зміну файлів) під час закриття програми, якщо їх не буде змінено з часу відкриття.

kate --desktopfile *назва_файла*

Базова назва файла для стільничного запису цієї програми.

Цей параметр, зокрема, корисний для програм-обгортки та програм, які мають декілька стільничних файлів (desktop). За його допомогою кожен стільничний файл може мати власний рядок команди, який вказується у записі Exec.

`kate --author`

Показує список авторів Kate у вікні термінала.

`kate -v --version`

Показує список відомостей щодо версії Kate.

`kate --license`

Показує інформацію щодо ліцензії на програму.

2.1.3 Перетягування і скидання

Kate використовує протокол перетягування зі скиданням KDE. Позначки файлів можна перетягнути і скинути до вікна Kate зі стільниці, інструменту для керування файлами — Dolphin або певного віддаленого сайту FTP, відкритого у одному з вікон Dolphin.

2.2 Робота у Kate

У розділі [Початок роботи](#) ви вже познайомилися з тим, як вмикати або вимикати чотири прості параметри, які нададуть вам змогу швидко налаштувати деякі з потужніших можливостей Kate. У розділі [Клавіатурні скорочення](#) наведено декілька типових клавіатурних скорочень для тих користувачів, які не можуть або не бажають використовувати мишу.

2.2.1 Швидкий запуск

У цьому розділі буде описано декілька з пунктів у меню **Перегляд**, за допомогою цих пунктів ви зможете швидко налаштувати Kate на роботу у бажаному для вас режимі.

Після першого запуску Kate ви побачите два вікна з білим тлом. Над цими вікнами буде розташовано панель інструментів з піктограмами з типовими мітками. Над всім цим буде розташовано смужку меню.

Вікно, розташоване ліворуч, — це бічна панель. У ній поєднано вікна **Документів** і **Навігатора файлової системи**. Перемикання між цими двома вікнами можна здійснити за допомогою натискання заголовків вкладок, розташованих ліворуч від вікна.

Якщо ви запустили Kate з зазначенням назви файла, який слід було відкрити, у вікні, розташованому праворуч, буде показано вміст файла, який ви редагуєте, а у вікні **Документи** на бічній панелі буде показано назву файла. Вікном **Навігатора файлової системи** можна скористатися для відкриття файлів.

Ви можете вмикати і вимикати показ усіх бічних панелей за допомогою меню **Перегляд** → **Вікна інструментів** або натискання комбінації клавіш **Ctrl+Alt+Shift+F**. За допомогою цього меню ви зможете поверхово оцінити потужність і гнучкість Kate. У цьому розділі меню ви побачите три пункти:

У списку **Вікна інструментів** буде наведено перелік усіх встановлених додатків. Щоб увімкнути чи вимкнути показ відповідного вікна інструментів позначте чи зніміть позначку із пункту у списку за допомогою клацання лівою кнопкою миші.

2.2.2 Клавіатурні скорочення

Значну кількість клавіатурних команд Kate (клавіатурних скорочень), можна налаштувати за допомогою меню [Параметри](#). Типово, у Kate використовуються наведені нижче клавіатурні скорочення.

F1	Довідка
Shift+F1	Що це?
Ctrl+N	Створює документ
Ctrl+L	Зберегти все
Ctrl+O	Відкрити документ
Ctrl+Alt+O	Швидке відкриття
Ctrl+Shift+F	Повноекранний режим
Ctrl+Shift+,	Налаштувати Kate
Ctrl+W / Ctrl+Esc	Закрити
Ctrl+Q	Вихід — закриває активну копію редактора
Ctrl+Alt+Shift+F	Показати бічні панелі
Ctrl+Shift+T	Розділити горизонтально
Ctrl+Shift+L	Розділити вертикально
F8	Наступний розділ перегляду
Shift+F8 / Ctrl+Esc	Попередній розділ перегляду
Ctrl+Shift+R	Закрити поточну панель перегляду
Alt+→	Наступна вкладка
Alt+←	Попередня вкладка
Ctrl+Shift+T	Повторно відкрити останні закриті документи

Крім того, ви можете скористатися клавіатурними скороченням компонента [KatePart](#) і усіх увімкнених [додатків до Kate](#).

2.3 Робота з багатодокументним інтерфейсом (MDI) Kate

2.3.1 Огляд

Вікно, перегляд, документ, фрейм, редактор... Що означають всі ці терміни у контексті Kate, як отримати найбільше користі від кожного з них? Цю главу присвячено відповідям на ці та деякі інші питання.

2.3.1.1 Головне вікно

Головним вікном Kate є стандартне вікно програми KDE з додаванням бічних панелей, на яких розташовано переліки інструментів. У вікні ви знайдете всі типові пункти меню разом з деякими додатковими пунктами, а також панель інструментів, призначену для доступу до команд, якими користуються найчастіше.

Найголовнішою частиною вікна є область редагування. Типово, буде показано єдиний компонент текстового редактора, за допомогою якого ви зможете працювати над вашими документами.

Можливості швартування панелей у вікні використовуються для вікон інструментів усіх додатків, які увімкнено за допомогою діалогового вікна параметрів програми.

Панелі інструментів можна розташовувати на будь-якій з бічних панелей. Щоб пересунути інструмент, наведіть вказівник миші на кнопку інструменту на бічній панелі, клацніть правою кнопкою миші і оберіть потрібну панель з контекстного меню.

Панель інструментів можна зробити *стійкою* за допомогою контекстного меню, яке викликається клацанням правою кнопкою миші на відповідній кнопці бічної панелі. На бічній панелі може бути показано декілька інструментів, отже, якщо один з них є стійким, інші інструменти також може бути показано одночасно з ним.

Якщо у додатка є параметри налаштування, ви можете скористатися першим пунктом контекстного меню для відкриття відповідної сторінки у діалоговому вікні параметрів роботи Kate.

2.3.2 Область редактора

У Kate ви можете відкрити декілька документів одночасно, а також розділити область редагування на довільну кількість фреймів, такий поділ схожий, наприклад, на поділ, який застосовується у Konqueror або популярному текстовому редакторі emacs. Таким чином ви зможете переглядати декілька документів або екземплярів одного документа одночасно, це зручно, наприклад, якщо на початку вашого документа містяться визначення, у які вам доводиться часто підглядати з довідковою метою. Іншим варіантом застосування фреймів є редагування файла реалізації функцій програми з одночасним переглядом файла заголовків у іншому фреймі.

Якщо документ відкрито у декількох редакторах, зміни, внесені у одному з редакторів, негайно буде показано у всіх інших областях редагування. Це стосується як зміни тексту, так і виділення фрагмента тексту. Результати дій з пошуку та пересування курсора відображатиметься лише у поточному редакторі.

У поточній версії програми неможливо відкрити декілька екземплярів одного документа так, щоб один з екземплярів можна було змінити, залишивши інший екземпляр незмінним.

Якщо ви накажете програмі з розділити вікно редактора на два фрейми, буде створено два фрейми однакового розміру, у обох буде показано поточний документ редактора. Новий фрейм буде розташовано в нижній частині вікна (у випадку горизонтального розподілу) або у правій частині (у випадку вертикального розподілу). Фокус буде передано новому фрейму, це буде показано за допомогою блимання курсора у фреймі, який знаходиться у фокусі.

2.4 Користування сеансами

За допомогою сеансів ви зможете зберігати у Kate декілька списків файлів і налаштувань графічного інтерфейсу програми. Ви можете створити довільну кількість сеансів з назвами, а сеанси без назви або анонімні сеанси використовувати для файлів, які ви маєте намір відкрити для редагування лише один раз. У поточній версії Kate може зберігати у сеансі список відкритих файлів і загальне налаштування вікна, у майбутніх версіях Kate до сеансів може бути додано інші відомості для зберігання. За допомогою сеансів ви також зможете відкривати довільну кількість екземплярів Kate замість одного, як було б за типової поведінки програми.

Підтримка сеансів передбачена у трьох областях:

- За допомогою *параметрів командного рядка* ви зможете обирати і запускати сеанси під час запуску Kate з командного рядка.
- За допомогою *меню «Сеанси»* ви зможете перемикаати, зберігати, запускати ваші сеанси, а також керувати ними.
- За допомогою *параметрів налаштування* ви зможете визначити загальну поведінку сеансів.

Після запуску нового сеансу буде завантажено налаштування графічного інтерфейсу для **Типового сеансу**. Щоб зберегти налаштування вікон у типовому сеансі, вам слід увімкнути збереження налаштувань вікон на сторінці налаштування сеансів діалогового вікна налаштувань, а потім завантажити типовий сеанс, потрібним чином налаштувати вікно і знову зберегти сеанс.

Якщо завантажено сеанс з назвою, Kate покаже назву сеансу у заголовку вікна, який набуде вигляду *«Назва сеансу: Назва або адреса URL документа - Kate»*

Якщо ви відкриваєте файли з командного рядка за допомогою параметра `--start назва` або якщо сеанс обрано за допомогою інструменту вибору сеансів, вказаний сеанс буде завантажено перед файлами, вказаними у командному рядку. Щоб відкрити файли з командного рядка у новому сеансі без назви, налаштуйте Kate на запуск нового сеансу як типового на сторінці сеансів діалогового вікна налаштування, або скористайтесь параметром `--start` з порожнім аргументом: `''`.

Починаючи з версії Kate 2.5.1 ідентифікатор процесу програми (PID) поточного екземпляра експортується до змінної середовища `KATE_PID`. Якщо ви відкриватимете файли з вбудованого терміналу, Kate автоматично обиратиме поточний екземпляр програми, якщо за допомогою командного рядка не буде вказано інших параметрів.

2.5 Швидке відкриття

Для швидкого відкриття або переходу між файлами у Kate передбачено вбудоване вікно швидкого відкриття файлів. Відкрити це вікно можна натисканням комбінації клавіш **Ctrl+Alt+O**.

У вікні швидкого відкриття можна побачити усі відкриті у Kate документи, а також усі файли у відкритих проєктах. Щоб побачити файли проєктів, вам слід увімкнути [додаток проєктів](#).

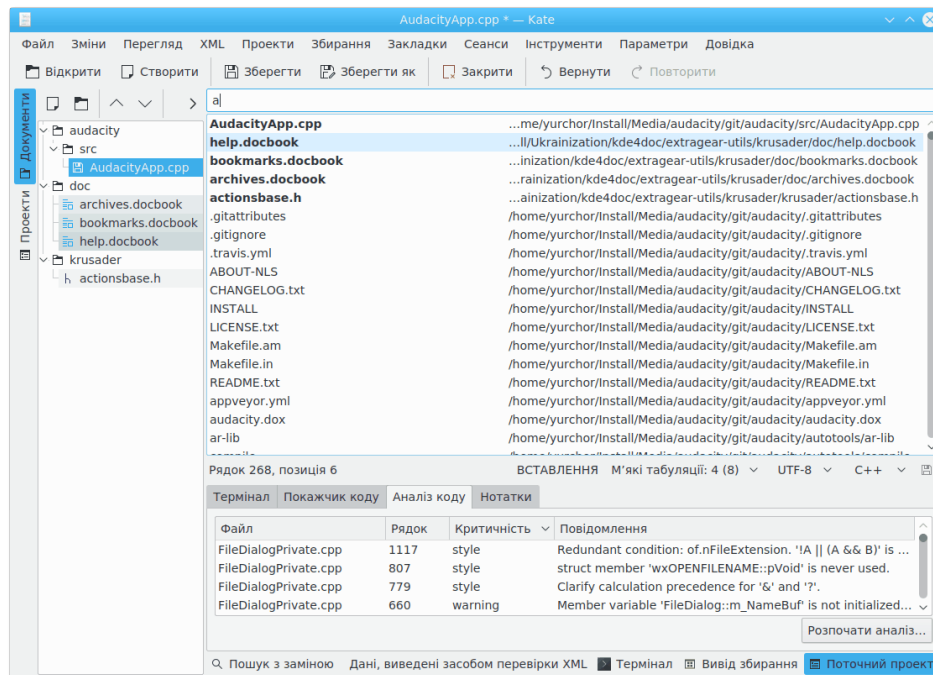
2.5.1 Користування «Швидким відкриттям»

Користування швидким відкриттям дуже просте. Після відкриття вікна просто почніть вводити назву або її частину для файла, який ви хочете відкрити, а додаток виконає фільтрування списку на основі введених даних. У відповідь на натискання **Enter** позначений файл буде відкрито, а натискання **Esc** приховає вікно швидкого відкриття.

Типово, під час фільтрування відповідність встановлюватиметься лише за назвою файла. Якщо ви хочете встановити відповідність і за шляхом, вам слід ввести «/» у полі критерію пошуку. Приклад: якщо введено «doc/index», додаток відфільтрує усі файли, у назві яких міститься «index» і які зберігаються у теці «doc».

Документи, які вже відкрито, буде позначено напівжирним шрифтом і розташовано у верхній частині списку одразу після відкриття вікна. Також додаток швидкого відкриття позначить останній відкрити документ, отже вам достатньо буде просто натиснути клавішу **Enter**, і додаток перенесе вас саме до цього документа.

2.5.2 Налаштовування «Швидкого відкриття»



У додатку швидкого відкриття передбачено декілька параметрів налаштувань. Щоб отримати доступ до цих параметрів, наведіть вказівник миші на поле введення і клацніть правою кнопкою.

У поточній версії можна встановити такі параметри:

Поточний проєкт — показувати файли лише з поточного проєкту

Усі проекти — показувати файли з усіх відкритих проектів

Нечітке фільтрування — скористатися алгоритмом нечіткої відповідності при фільтруванні файлів

Фільтрування за шаблоном — скористатися відповідністю за шаблоном при фільтруванні файлів

2.6 Як отримати довідку

2.6.1 За допомогою Kate

Цей підручник

Відкриває докладну довідку з усіх команд меню, параметрів налаштування, інструментів, діалогових вікон, додатків тощо, а також вікна Kate редактора та різноманітних концепцій, використаних у цій програмі.

Натисніть клавішу **F1** або скористайтесь пунктом меню **Довідка** → **Підручник з Kate**, щоб переглянути цей підручник.

Довідка «Що це?»

За допомогою довідки «Що це?» можна негайно отримати довідку щодо призначення окремих елементів вікон графічного інтерфейсу програми, таких як кнопки та інші ділянок вікна.

Автори намагалися створити контекстну довідку до всіх елементів вікна, для яких вона має сенс. Ви, зокрема, зможете скористатися нею у діалоговому вікні налаштування, а також у багатьох інших вікнах.

Щоб задіяти довідку «Що це?», натисніть комбінацію клавіш **Shift+F1** або скористайтесь пунктом меню **Довідка** → **Що це?**. Виконавши ці дії, ви увімкнете режим «Що це?»: вигляд вказівника миші зміниться на стрілочку зі знаком питання, ви зможете натиснути нею будь-який з елементів у вікні, щоб прочитати коротеньку довідку щодо цього елемента, якщо ця довідка доступна.

Кнопки «Довідка» у діалогових вікнах

У деяких з діалогових вікон є кнопки **Довідка**. Натискання таких кнопок призводить до запуску KHelpCenter і відкриття відповідної документації.

2.6.2 З вашими текстовими файлами

У Kate (ще) не передбачено ніяких засобів для читання пов'язаної з документом документації. Залежно від типу документа, який ви редагуєте, ви можете скористатися [вбудованим емулятором термінала](#) для перегляду відповідних сторінок довідки (man) UNIX[®] та документації info або Konqueror для перегляду іншої документації.

2.6.3 Статті щодо Kate

На домашній сторінці Kate наведено декілька [статей та настанов](#) щодо відомостей, які не належать до кола питань цього підручника.

Розділ 3

Робота у редакторі Kate

Щоб дізнатися більше про основи роботи з компонентом редактора, який є основою Kate, ознайомтеся із [главою щодо роботи з редактором KatePart підручника з KatePart](#).

Розділ 4

Робота з додатками

Anders Lund

Переклад українською: Юрій Чорноіван

Увімкнути окремі додатки можна за допомогою [діалогового вікна налаштування](#), за допомогою цього вікна можна також отримати доступ до додаткових параметрів налаштування тих додатків, які потребують налаштування.

4.1 Додатки програми Kate

Додатки Kate призначено для виконання додаткових функцій у редакторі Kate. За їх допомогою можна додавати нові меню і скорочення, а також розширювати список можливостей Kate. Ви можете встановити будь-яку бажану кількість додатків за допомогою самої програми Kate. Для цього скористайтесь діалоговим вікном налаштування Kate, яке можна відкрити за допомогою пункту меню **Параметри** → **Налаштувати Kate...** Щоб обрати потрібні додатки, перейдіть на сторінку **Програма** → **Додатки**.

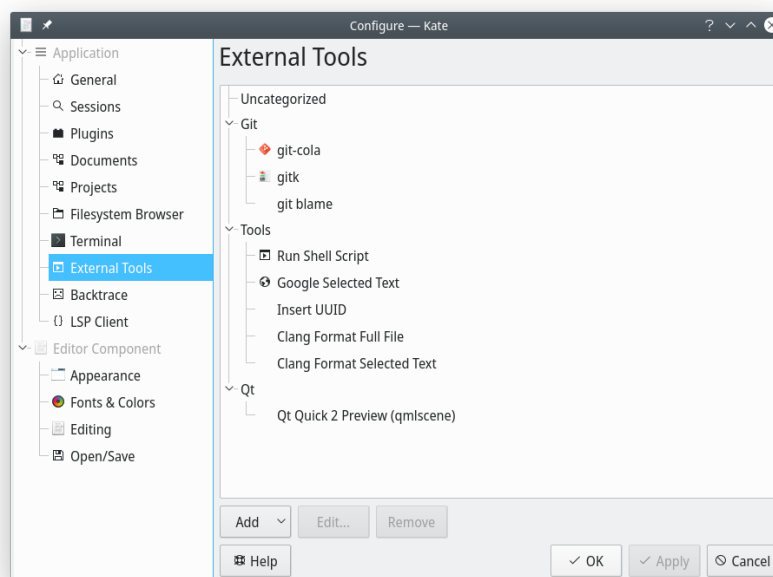
Серед можливих додатків до програми:

- [Зовнішні інструменти](#) — запуск зовнішніх інструментів і програм
- [Переглядач зворотного трасування](#) допоможе вам переглянути дані зворотного трасування C/C++
- [Додаток збирання](#) — компілює або збирає програми і обробляє повідомлення про помилки.
- [Додаток закриття за критерієм](#) — закрити групу документів на основі даних щодо спільного для них розташування у файловій системі або суфікса назви файла.
- [Піпетка](#) — показує панель попереднього перегляду для відомих назв кольорів
- [Кольорові дужки](#) — кольорові дужки для зручнішого читання
- [Додаток CTags](#) виконує пошук визначень або оголошень за допомогою CTags.
- [Попередній перегляд документів](#) — попередній перегляд документів у форматі призначення.
- [Перемикач між документами](#) — налаштування швидкого перемикання документів за допомогою натискання комбінації клавіш **Alt+Tab**.
- [Переглядач файлової системи](#) — панель засобу навігації файловою системою
- [Ієрархія документів](#) показує структуру відкритих файлів.
- [GDB](#) — надає доступ до простої графічної оболонки до GDB
- [Додаток проєктів](#) — інтеграція із Git та іншими системами керування кодом

- Replicode — конструктивістська мова програмування і середовище виконання для штучного інтелекту
- [Клієнт LSP](#) — клієнт LSP, який надає можливості навігації кодом та автоматичного доповнення коду для багатьох мов.
- [Пошук з заміною](#) — пошук з заміною у відкритих документах, теках або проєктах
- [Панель скриптів і фрагментів](#) — створення і отримання фрагментів коду та скриптів редактора
- [Клавіатурні макроси](#) — запис і відтворення клавіатурних макросів (тобто послідовностей дій з клавіатури)
- [Додаток SQL](#) — виконання запитів до баз даних SQL.
- [Переглядач символів](#) призначено для добування і показу символів з початкового коду.
- [Панель термінала](#) — термінал на основі віджета Konsole KDE
- [Фільтр тексту](#) — обробка тексту за допомогою команд з термінала
- [Доповнення XML](#) — показує список елементів, атрибутів, значень атрибутів та об'єктів XML, передбачених у DTD.
- [Додаток Перевірка XML](#) перевіряє коректність файлів XML за допомогою xmllint.
- [Навігатор компілятора](#) — інтерфейс до засобу навігації результатами збирання
- [Форматування](#) — додаток для форматування коду

4.2 Зовнішні інструменти

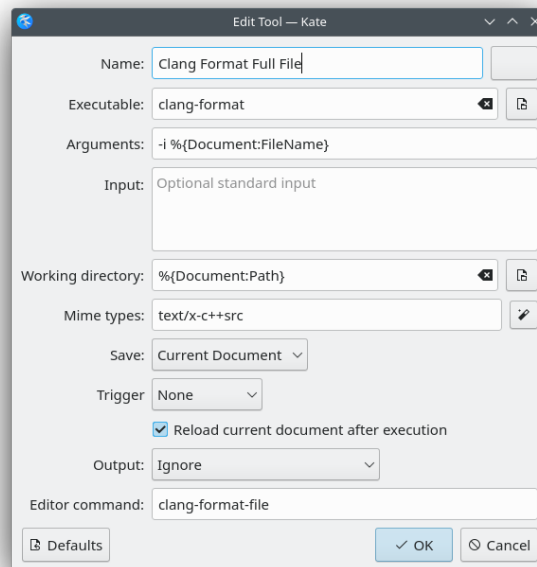
За допомогою додатка **Зовнішні інструменти** можна викликати зовнішні програми, передаючи їм дані, які пов'язано із поточним документом, наприклад його адресу, каталог, текст або позначений фрагмент тексту. Після вмикання буде показано сторінку налаштування, знімок якої наведено нижче. За допомогою цієї сторінки ви зможете змінити роботу наявних інструментів або вилучити їх. Так само, може бути додано потрібні вам інструменти. Пункти цих інструментів згодом з'являться у підменю **Зовнішні інструменти** меню **Інструменти** програми.



Додати нові зовнішні інструменти на сторінці налаштувань можна натисканням кнопки **Додати**. У відповідь буде показано контекстне меню, за допомогою якого можна або додати новий зовнішній інструмент, або додати наявний інструмент із попередньо визначеного списку, або додати нову категорію для упорядкування зовнішніх інструментів за категоріями. Так само, можна вносити зміни до наявних записів інструментів: двічі клацніть на відповідному пункті у списку або натисніть **Змінити....** Натискання кнопки **Вилучити** вилучає запис інструмента зі списку.

4.2.1 Налаштовування зовнішніх інструментів

У відповідь на вибір редагування запису програма відкриває діалогове вікно налаштування, за допомогою якого можна виконати коригування поведінки інструмента:



Як можна бачити, ви можете визначити багато елементів поведінки інструмента, зокрема:

Назва — назва інструмента, яку буде показано у меню.

Піктограма — необов'язкова піктограма, яку буде показано у меню.

Виконуваний файл — назва виконуваного файлу — можна вказати повний шлях або просто назву програми, якщо виконуваний файл зберігається в одному з каталогів, описаних у змінній PATH.

Аргументи — необов'язкові аргументи, які передаються виконуваному файлу.

Вхідні дані — необов'язкові вхідні дані, які буде передано процесу за допомогою стандартного потоку вхідних даних (stdin).

Робочий каталог — робочий каталог, з якого буде запущено інструмент. Якщо значення буде порожнім, робочим каталогом буде каталог поточного документа.

Типи MIME — якщо встановлено, інструмент ставатиме активним, лише якщо відповідним буде тип MIME поточного документа.

Збереження — якщо задіяно, не зберігає нічого, зберігає поточний документ або зберігає усі документи.

Перемикач — перемикач для виконання цього інструмента. Перемикач стосуватиметься лише поточного активного документа і застосовуватиме інструмент, лише якщо тип

MIME поточного активного документа відповідає типу MIME, який обробляє зовнішній інструмент.

Можна використовувати такі перемикачі:

Немає — типове значення; означає, що інструмент не має перемикача.

До збереження — цей перемикач призводитиме до виконання інструмента одразу перед збереженням документа.

Після збереження — цей перемикач призводитиме до застосування інструмента після збереження документа.

Перезавантажити поточний документ після виконання — корисно, якщо до поточного файла внесено зміни на диску.

Виведення — визначає призначення даних стандартного виведення (stdout). Можна вибрати один з таких варіантів: Ігнорувати, Вставити у позиції курсора, Замінити позначений фрагмент тексту, Замінити поточний документ, Дописати до поточного документа, Вставити до нового документа, Скопіювати до буфера або Показати на панелі.

Команда редактора — необов'язкова команда, яку буде використано для виклику зовнішнього інструмента за допомогою вбудованого [командного рядка](#).

Кнопку **Типові параметри** буде показано лише для інструментів, які постачаються разом із Kate. Якщо натиснути цю кнопку, усі параметри інструмента буде повернуто до типових (встановлених разом із програмою) значень.

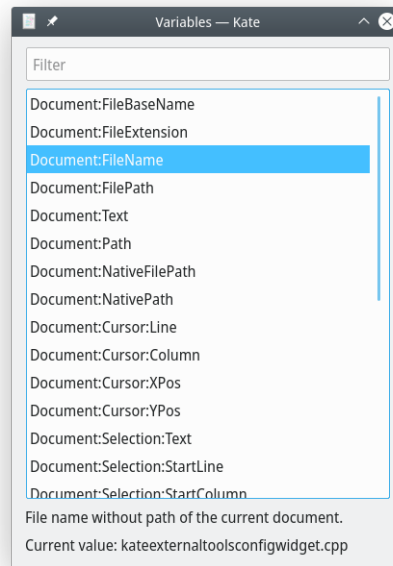
4.2.2 Розгортання змінних

У деяких з полів редагування, зокрема полях **Виконуваний файл**, **Аргументи**, **Вхідні дані** і **Робочий каталог**, передбачено підтримку змінних, які буде замінено на значення під час виклику інструмента. Це позначається піктограмою {}, яку програма показує, коли фокус передається на одне з цих полів введення (див. у червоному колі):

The image shows a configuration window for an external tool in Kate. It contains four main input fields:

- Executable:** A text box containing "git-cola" with a folder icon to its right.
- Arguments:** A text box containing "-r %){Document:Path}". A red circle highlights the curly brace "}" in the argument string.
- Input:** A text box containing "-r /home/dh/kde/kf5/src/kate/addons/externaltools".
- Working directory:** A text box containing "Uses current document path if empty" with a folder icon to its right.

Крім того, при наведенні вказівника миші на одне з таких полів для введення тексту програма показує панель підказки із поточним розгорнутим текстом. Також, натискання кнопки {} відкриває діалогове вікно, у якому наведено список усіх доступних змінних:



Ця можливість додає гнучкості у визначенні зовнішнього інструмента, оскільки усіх змінні у формі `%{...}` розгортаються під час виклику інструмента. Передбачено підтримку двох типів змінних:

- `%{назва-змінної}`
- `%{назва-змінної:<значення>}`

Якщо використано першу форму, `%{назва-змінної}`, програма просто замінить змінну її вмістом. Наприклад, змінну `%{Document:FileName}` буде замінено на назву файла поточного документа без шляху до нього. Якщо використано другу форму, `%{назва-змінної:<значення>}`, отримаємо `<значення>` як вміст. Наприклад, цим можна скористатися для розгортання змінної середовища за допомогою `%{ENV:HOME}` або можна отримати поточну дату у бажаному форматі, ось так: `%{Date:yyyy-MM-dd}`.

Ось підтримувані змінні:

`Document:FileName`: базова назва файла поточного документа без шляху і суфікса назви.

`Document:FileExtension`: суфікс назви файла поточного документа.

`Document:FileName`: назва файла поточного документа без шляху.

`Document:FilePath`: повний шлях до поточного документа, включно із назвою файла.

`Document:Text`: вміст поточного документа.

`Document:Path`: повний шлях до поточного документа без назви файла.

`Document:NativeFilePath`: шлях до документа повністю, разом із назвою файла та роздільниками шляху операційної системи (зворотними похилими рисками у Windows).

`Document:NativePath`: шлях до документа повністю, без назви файла та роздільниками шляху операційної системи (зворотними похилими рисками у Windows).

`Document:Cursor:Line`: номер рядка поточної позиції текстового курсора у поточному документі (нумерація рядків починається з 0).

`Document:Cursor:Column`: номер позиції у рядку текстового курсора у поточному документі (нумерація рядків починається з 0).

`Document:Cursor:XPos`: компонент X загальної позиції курсора на екрані.

`Document:Cursor:YPos`: компонент Y загальної позиції курсора на екрані.

`Document:Selection:Text`: позначений текст у поточному документі.

`Document:Selection:StartLine`: номер початкового рядка для позначеного фрагмента поточного документа.

`Document:Selection:StartColumn`: номер позиції позиції у рядку для позначеного фрагмента поточного документа.

`Document:Selection:EndLine`: номер кінцевого рядка для позначеного фрагмента поточного документа.

`Document:Selection:EndColumn`: номер кінцевої позиції у рядку для позначеного фрагмента поточного документа.

`Document:RowCount`: кількість рядків у поточному документі.

`Document:Variable:<змінна>`: розгорнути довільні [змінні документа](#).

`Date:Locale`: поточна дата у форматі поточної локалі.

`Date:ISO`: поточна дата (ISO).

`Date:<значення>`: поточна дата ([у форматі, вказаному рядком форматування QDateTime](#)).

`Time:Locale`: поточний час у форматі поточної локалі.

`Time:ISO`: поточний час (ISO).

`Time:<значення>`: поточний час ([рядок у форматі QDateTime](#)).

`ENV:<значення>`: доступ до змінних середовища.

`JS:<вираз>`: обробка простих інструкцій JavaScript.

`PercentEncoded:<текст>`: текст у кодуванні з використанням кодів спеціальних символів із символом відсотків.

`UUID`: створити UUID.

4.2.3 Список типових інструментів

До типового комплексу програми входять декілька інструментів. Втім, якщо у вас є ідеї щодо корисніших інструментів, будь ласка, повідомте про них на сторінці [нашого проєкту на GitLab](#), щоб розробники могли включити ці інструменти до наступних випусків програми. Усі типові інструменти, типово, буде показано на панелі списку. Втім, ви можете вносити зміни до списку, категоризувати пункти або навіть вилучити інструменти. Вилучені інструменти можна знову додати за допомогою натискання кнопки **Додати** на панелі налаштувань, як це описано вище.

GIT-COLA

git-cola — графічний клієнт **git**, за допомогою якого можна без проблем створювати записи змін і вносити зміни до сховища. Якщо програму встановлено, доступ до неї можна також отримати введенням команди **git-cola** у командному рядку.

Назва: **git-cola**
Піктограма: **git-cola**
Виконуваний файл: **git-cola**
Аргументи: **-r %\{Document:Path\}**
Команда редактора: **git-cola**

GITK

gitk — також клієнтська програма **git**, за допомогою якої можна візуалізувати журнал змін у **git**.

Назва: **gitk**
Піктограма: **git-gui**
Виконуваний файл: **gitk**
Робочий каталог: **%\{Document:Path\}**
Команда редактора: **gitk**

GIT BLAME

Запускає `git blame` для полегшення стеження за змінами у `git` для поточного файлу.

Назва: `git blame`
Виконуваний файл: `git`
Аргументи: `gui blame %{\Document:FileName}`
Збереження: поточний документ
Робочий каталог: `%{\Document:Path}`
Команда редактора: `git-blame`

ВИКОНАТИ СКРИПТ ОБОЛОНКИ

Запускає зовнішню `Konsole`, у якій виконує поточний документ. Скрипт має оголошувати у першому рядку свій інтерпретатор, ось так: `#!/шлях/інтерпретатор`.

Назва: `Run Shell Script`
Піктограма: `system-run`
Виконуваний файл: `konsole`
Аргументи: `-e sh -c "cd %{\Document:Path} && pwd && chmod -vc a+x %{\Document:FileName} && ./%{\Document:FileName} ; echo Press any key to continue. && read -n 1"`
Збереження: поточний документ
Робочий каталог: `%{\Document:Path}`
Команда редактора: `run-script`

ПОШУК У GOOGLE ПОЗНАЧЕНОГО ТЕКСТУ

Пошук позначеного фрагмента тексту за допомогою рушія пошуку `Google`.

Назва: `Google Selected Text`
Піктограма: `globe`
Виконуваний файл: `xdg-open`
Аргументи: `"https://www.google.com/search?q=%{\Document:Selection:Text}"`
Команда редактора: `google`

ВСТАВИТИ UUID

Вставляє новий `UUID` під час кожного виклику цієї дії.

Назва: `Insert UUID`
Виконуваний файл: `echo`
Аргументи: `%{\UUID}`
Виведення: Вставити у позиції курсора
Команда редактора: `uuid`

ФОРМАТУВАННЯ CLANG УСЬОГО ФАЙЛА

Виконує clang-format для поточного файлу на диску. Після обробки програма повторно завантажує документ.

Назва: Clang Format Full File
Виконуваний файл: clang-format
Аргументи: -i %\{Document:FileName\}
Робочий каталог: %\{Document:Path\}
Збереження: поточний документ
Перезавантажити: Так
Команда редактора: clang-format-file

ФОРМАТУВАННЯ CLANG ПОЗНАЧЕНОГО ТЕКСТУ

Запускає clang-format лише до позначеного тексту у поточному документі.

Назва: позначений для форматування Clang текст
Виконуваний файл: clang-format
Аргументи: -assume-filename: %\{Document:FileName\}
Робочий каталог: %\{Document:Path\}
Вхідні дані: %\{Document:Selection:Text\}
Виведення: замінити позначений фрагмент тексту
Команда редактора: clang-format-selection

ПОПЕРЕДНІЙ ПЕРЕГЛЯД QT QUICK 2 (QMLSCENE)

Попередній перегляд поточного файлу QML у qmlscene.

Назва: Qt Quick 2 Preview (qmlscene)
Виконуваний файл: qmlscene
Аргументи: %\{Document:FileName\}
Збереження: поточний документ
Робочий каталог: %\{Document:Path\}
Команда редактора: qml-preview

ФОРМАТУВАННЯ JSON УСЬОГО ФАЙЛА

Форматування усього файлу коду JSON.

Назва: JSON Format Full File
Піктограма: application-json
Виконуваний файл: jq
Аргументи: %\{Document:FileName\}
Збереження: поточний документ
Робочий каталог: %\{Document:Path\}
Виведення: Замінити поточний документ
Команда редактора: json-format-file

ФОРМАТУВАННЯ XML УСЬОГО ФАЙЛА

Форматування усього файла коду XML.

Назва: XML Format Full File
Піктограма: application-xml
Виконуваний файл: xmllint
Аргументи: --format %{\Document:FileName}
Збереження: поточний документ
Робочий каталог: %{\Document:Path}
Виведення: Замінити поточний документ
Команда редактора: xml-format-file

4.3 Додаток перегляду зворотного трасування

4.3.1 Користування додатком перегляду зворотного трасування

Цей додаток призначено в основному для розробників, а не для звичайних користувачів. За його допомогою можна переглянути дані зворотного трасування, які створено за допомогою gdb, у форматі списку у вікні інструмента Kate. Наведенням вказівника миші на пункт у цьому списку із наступним клацанням лівою кнопкою миші можна відкрити відповідний файл і перейти у ньому до відповідного рядка коду. Такий спосіб навігації працює із даними зворотного трасування, які створено не лише на вашому комп'ютері: ви можете скористатися даними інших користувачів, якщо файли коду розташовано у однакових місцях. Для роботи інструментів додатка слід проіндексувати каталоги, у яких зберігаються файли із початковим кодом програми.

Іноді може існувати декілька файлів із однаковою назвою. Приклад:

```
kdegraphics/okular/generators/dvi/config.h  
kdepim-runtime/resources/gmail/saslplugin/config.h
```

Для того, щоб зробити правильний вибір, у додатку використовуються останні дві частини адреси, у нашому прикладі такі:

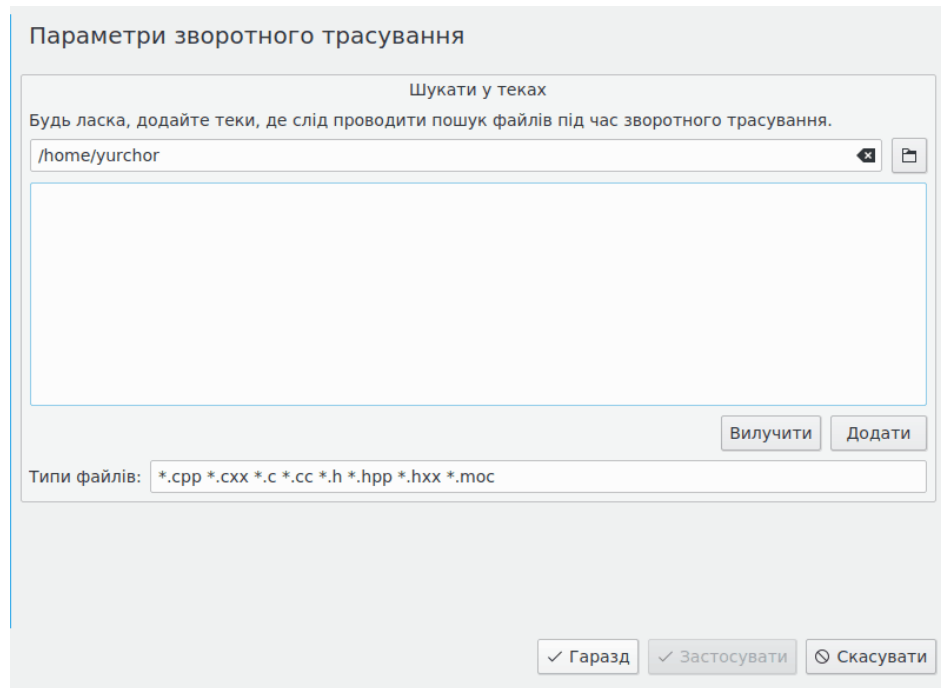
```
dvi/config.h  
saslplugin/config.h
```

Зазвичай, додаток визначає належний файл автоматично.

Одночасне індексування основної і бічних гілок розробки, звичайно ж, призведе до конфлікту.

4.3.2 Налаштування

На сторінці налаштування можна додати каталоги, у яких міститься початковий код програми.



Налаштовування шляхів у вікні інструмента перегляду зворотного трасування

Натисканням кнопки **Гаразд** можна дозволити програмі розпочати індексування. Після завершення індексування можна відкрити вікно інструмента **Перегляд зворотного трасування**.

Після цього можна завантажити дані зворотного трасування з буфера обміну даними (наприклад, після натискання кнопки **Копіювати до буфера** у DrKonqi) або з файла.

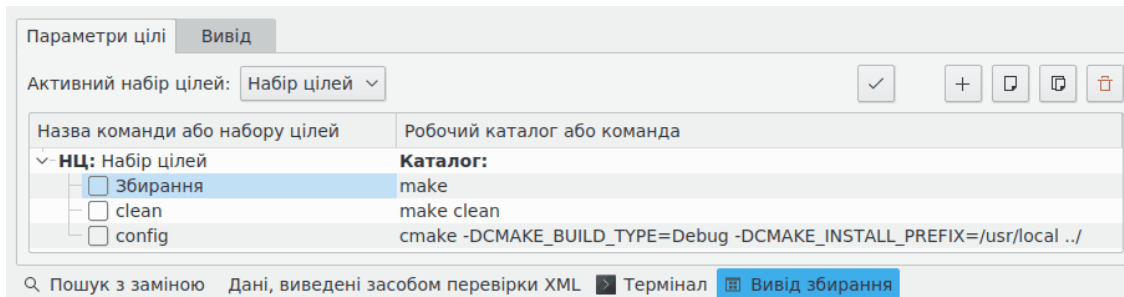
4.4 Додаток для збирання

4.4.1 Вступ

За допомогою додатка збирання можна виконувати такі дії, як збирання, очищення та компіляція проекту. Ви також можете автоматично запускати створені програми. По суті, додаток надає вам змогу налаштувати набори команд для запуску, він також може аналізувати вихідні дані у пошуку посилань на файли та певних рядків і стовпців у цих файлах. Отже, хоча додаток від початку було створено для компіляції C/C++ і здебільшого перевірено у цих мовах, додаток також може бути корисним для інших цілей і мов.

4.4.2 Користування додатком збирання

Додаток збирання додає панель інструментів **Збирання** у нижній частині вікна програми та меню **Збирання** на смужку меню. За допомогою панелі інструментів ви зможете вибрати цілі і налаштувати параметри збирання цілей, меню ж і його скороченнями можна скористатися для вибору і виконання налаштованих команд оболонки.



На панелі **Збирання** передбачено дві статичні вкладки:

- **Параметри цілі**
- **Вивід**

4.4.2.1 Вкладка «Параметри цілі»

Вкладкою «Параметри цілі» можна скористатися для налаштування різноманітних цілей збирання та визначення наборів збирання.

Набір цілей — це група команд, які можна виконувати у вказаному робочому каталозі. Набори цілей можна додати на панелі **Проекти** або **Сеанс**. Налаштовані набори цілей, додані на панелі **Проекти**, зберігаються у файлі **.kateproject.build** у кореневому каталозі проекту та відновлюються під час повторного відкриття проекту. Набори цілей, додані до **Сеанс**, зберігаються в поточних налаштуваннях сеансу Kate.

Перший рядок набору цілей містить назву набору у першому стовпці, а в другому — каталог, де мають виконуватися команди. У кожному наступному рядку ми маємо назву для команди в першому стовпці, команду збирання у другому та команду запуску в третьому стовпці. Щоб відредагувати один із них, двічі клацніть на записі або скористайтеся клавіатурним скороченням редагування (часто **F2**).

Робочий каталог

Другий стовпець першого рядка набору цілей використовується для налаштування робочого каталогу, що використовується для компіляції та виконання команд. Якщо додаток проекту ввімкнено, рядок робочого каталогу також може містити замітники для шляху основного каталогу проекту: **%B** та назви: **%b**

Команда збирання

Другий стовпець у «не перших рядках» містить команду оболонки, яку слід запустити у робочому каталозі. Зверніть увагу на слово «оболонка». Підійде майже будь-яка команда оболонки. Команда збирання може містити замітники. **%f** для поточного файла, **%d** для каталогу поточного файла та **%n** для базової назви поточного файла (назви файла без суфікса).

Виконати команду

Третій стовпець у «не перших рядках» може містити команду оболонки, яку слід виконати у фактичному терміналі у робочому каталозі цілі. Термінал відкривається як вкладка. Додаток спробує повторно використати вкладку терміналу, якщо виконано ту саму команду, а попередня програма завершила роботу.

У верхній частині вкладки **Параметри цілі** розташовано панель інструментів із фільтром цілей та такими кнопками:

-  — зібрати позначену ціль
-  — зібрати і запустити позначену ціль
-  — додати нову ціль збирання
-  — Створити набір цілей збирання
-  — копіювати команду або набір цілей
-  — вилучити поточну команду або набір цілей

4.4.2.2 Вкладка «Вивід»

На вкладці **Виведення** буде показано виведені до консолі дані під час виконання команди (збирання). Якщо у рядку виведення буде вказано розташування файла, відповідне місце можна буде натиснути. Якщо у рядку виведених даних буде показано повідомлення про помилку або попередження, рядок матиме відмінний від інших колір.

Якщо позначено пункт **Додати помилки і попередження до діагностики** на сторінці параметрів додатка у вікні параметрів Kate, на панелі діагностики буде показано повідомлення про помилки і попередження. Щоб перейти до місця попередньої помилки, натисніть комбінацію клавіш **Alt+Shift+←**. Щоб перейти до наступної помилки, натисніть комбінацію клавіш **Alt+Shift+→**.

4.4.3 Структура меню

Збирання → Вибрати ціль...

Фільтр вибору цілі фокусується на вкладці «Параметри цілі». Введення назви призведе до відфільтровування цілей, які не відповідають введеному рядку. Для навігації ієрархією цілей також можна скористатися клавішами зі стрілками. Якщо потрібну ціль вибрано, натискання **Return** або **Enter** виконає та запустить вибрану ціль.

Збирання → Зібрати позначену ціль

Зібрати останню позначену ціль. Якщо нічого не позначено, працює так само, як **Вибрати ціль...**

Збирання → Зібрати і запустити позначену ціль

Зібрати останню позначену ціль і запустити команду після успішного завершення команди збирання.

Збирання → Компілювати поточний файл

Спробувати скомпілювати поточний файл шляхом пошуку команди у можливому `compile_commands.json`.

Збирання → Зупинити

Припинити збирання цілі.

Збирання → Фокус на наступну вкладку ліворуч

Сфокусувати наступну вкладку додатка збирання ліворуч. Або відкрити панель додатка збирання, якщо її приховано.

Збирання → Фокус на наступну вкладку праворуч

Сфокусувати наступну вкладку додатка збирання праворуч. Або відкрити панель додатка збирання, якщо її приховано.

Збирання → Завантажити цілі з каталогу збирання CMake

Відкрити діалогове вікно вибору файлу і надати користувачеві можливість вибрати **CMakeCache.txt**. Якщо файл вибрано, додаток створить команду збирання цілей `stake`, яку може бути виконано у каталозі збирання для заснованого на CMake проєкту.

4.4.4 Подяки

Додаток збирання Kate створено Kåre Særs.

Особлива подяка учасниці Google Code-In 2011 Salma Sultana за написання більшої частини цього розділу.

4.5 Додаток закриття за критерієм

4.5.1 Вступ

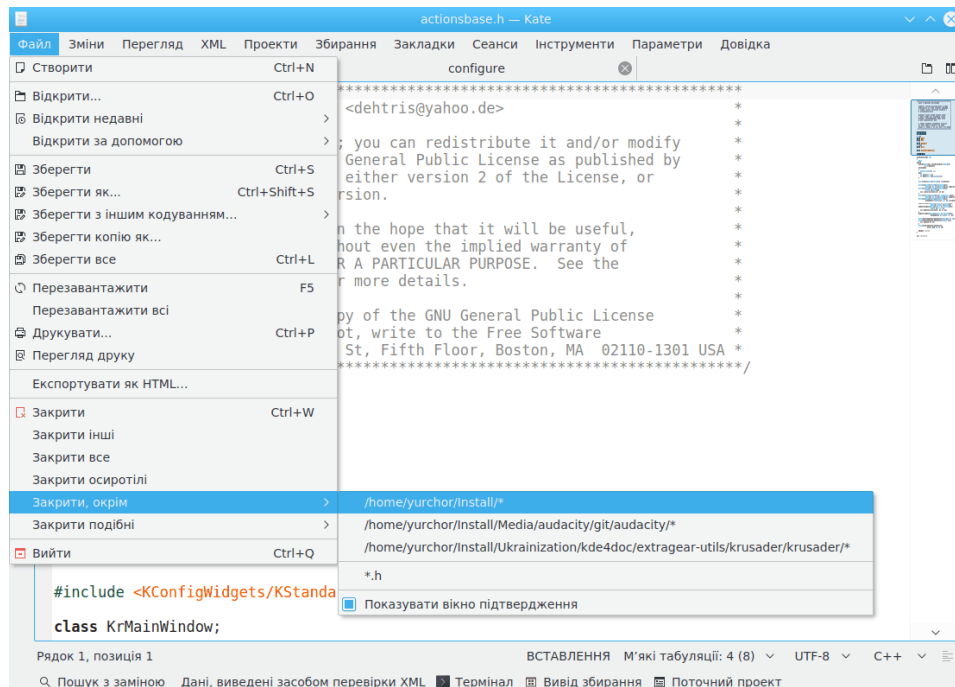
За допомогою цього додатка можна закрити групу документів на основі вказаних даних щодо суфікса назви і шляху.

4.5.2 Користування додатком закриття за критерієм

Нехай у Kate відкрито такі документи:

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

Ви зможете скористатися такими варіантами закриття документів, показаними на знімку:



Скористайтесь полем для позначки навпроти останнього пункту у списку, щоб увімкнути або вимкнути діалогове вікно підтвердження дії. Вибраний варіант буде застосовано до обох дій із закриття.

4.5.3 Структура меню

Файл → Закрити, окрім

Закрити усі відкриті документи, *окрім* тих, які за шляхом або суфіксом назви збігаються із варіантом, вибраним за допомогою підменю.

Файл → Закрити подібні

Закрити усі відкриті документи, які за шляхом або суфіксом назви збігаються із варіантом, вибраним за допомогою підменю.

4.6 Додаток піпетки

4.6.1 Вступ

За допомогою цього додатка можна скористатися вбудованою панеллю попереднього перегляду або піпеткою для кольорів у тексті (наприклад `#FFFFFF`, `white`).

Щоб завантажити цей додаток, відкрийте діалогове вікно налаштування Kate за допомогою пункту меню **Параметри** → **Налаштувати Kate...** Після цього позначте пункт **Піпетка** і закрийте діалогове вікно.

4.6.2 Налаштування

На сторінці параметрів додатка «Піпетка» у налаштуваннях Kate ви можете налаштувати вказані нижче параметри роботи додатка.

Показ попереднього перегляду кольорів із відомими назвами

Визначає, чи слід показувати панель піпетки для відомих назв кольорів (наприклад `skyblue`). Список назв кольорів наведено на [цій сторінці](#).

Вставляти перегляд після текстового кольору

Визначає, чи слід вставляти вбудовану панель попереднього перегляду після текстового запису кольору у коді.

Шістнадцяткові коди кольорів

Тут ви можете вибрати найвідповідніший варіант для кольорів у вашому коді.

4.7 Кольорові дужки

4.7.1 Вступ

Додаток кольорових дужок розфарбовує відповідні пари дужок різними кольорами для того, щоб читати текст було зручніше. Втім, розфарбуванню підлягають не усі дужки. Дужки, пару яких розташовано поза видимою частиною тексту, розфарбовано не буде. Так само, не буде розфарбовано пару дужок, якщо вона є єдиною парою у рядку.

4.7.2 Налаштування

Налаштувань для цього додатка не передбачено.

4.8 Додаток CTags

4.8.1 Вступ

CTags створює файл покажчика (або файл міток) об'єктів мови, які буде знайдено у файлах початкового коду. Після створення такого файла об'єкти можна буде швидко і просто знайти за допомогою цього додатка до Kate.

Мітка позначає об'єкт мови програмування, для якого існує запис у покажчику (або, інакше кажучи, запис покажчика для цього об'єкта).

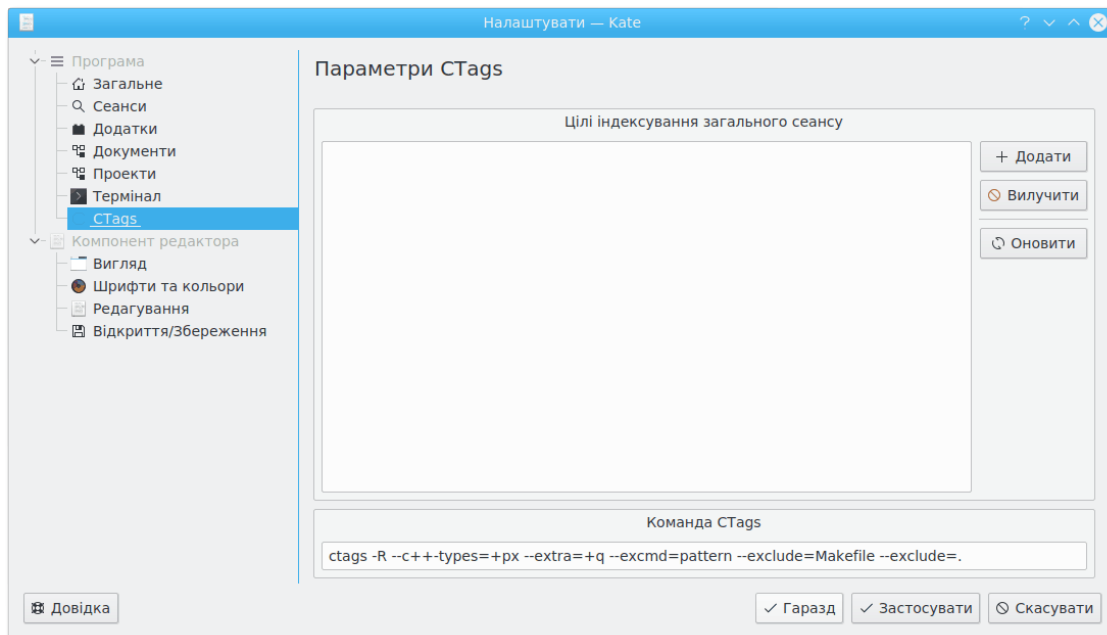
Створення міток можливе для [цих мов програмування](#).

4.8.2 Налаштування

Додаток CTags використовує два різних файли баз даних для зберігання даних покажчика.

За допомогою сторінки налаштування додатка CTags у вікні параметрів Kate ви можете додавати або вилучати записи каталогів, у яких містяться файли із початковим кодом, а також повторно створювати загальну базу даних CTags.

4.8.2.1 Загальний покажчик



Налаштовування загальної бази даних CTags

За допомогою нижньої частини сторінки налаштування ви можете вказати **Команду CTags**.

Щоб дізнатися більше про усі доступні параметри, будь ласка, ознайомтеся із сторінкою підручника (man) CTags. Прочитати цю сторінку можна у «Центрі довідки» (KHelpCenter). Крім того, можна просто ввести адресу `man:/ctags` безпосередньо у поле для адреси вікна Konqueror

Натискання кнопки **Оновити** накаже програмі розпочати індексування. Коли індексування буде завершено, закрийте діалогове вікно.

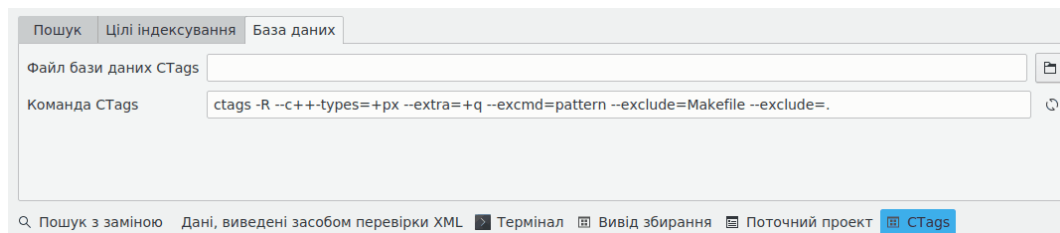
4.8.2.2 Показчик сеансу

Щоб налаштувати показчик сеансу, відкрийте панель **CTags**.

Цілі індексування

За допомогою цієї вкладки ви можете додавати або вилучати записи каталогів, які містять початковий код програми, і вручну запускати процедуру повторного створення специфічної для сеансу бази даних CTags.

База даних



Налаштовування бази даних сеансів CTags

Вибір іншого файлу бази даних CTags, налаштування команди CTags або повернення до типової команди.

4.8.3 Користування додатком CTags

Вам достатньо навести вказівник миші на об'єкт мови, зокрема функцію, символ тощо, який вам потрібен, а потім вибрати один із пунктів меню **CTags**, щоб перейти до рядка і файла, де цей об'єкт визначено чи оголошено.

Типово, із пунктами меню **CTags** не пов'язано жодних клавіатурних скорочень. Якщо вам хочеться визначити такі скорочення, скористайтеся [редактором клавіатурних скорочень](#).

Крім того, можна скористатися полем для пошуку на вкладці **Пошук** панелі CTags.

Після введення перших символів назви об'єкта у поле для пошуку програма спробує знайти відповідні об'єкти мови за назвою. Буде показано відповідні функції, класи, символи тощо разом із даними щодо їхнього типу та назви відповідного файла.

Позначте пункт у списку, щоб перейти до відповідного рядка у файлі із початковим кодом.

4.8.4 Структура меню

CTags → Повернутися на один крок

Перейти назад за записом журналу щодо останньої відвіданої мітки.

CTags → Шукати поточний текст

Відкриває вкладку **Пошук** панелі CTags і показує список усіх об'єктів мови, що відповідають поточному позначеному фрагменту тексту.

CTags → Перейти до оголошення

Якщо курсор перебуває у межах визначення об'єкта, буде відкрито документ, що містить відповідне оголошення об'єкта, активовано цей документ, а курсор буде пересунуто на початок оголошення.

CTags → Перейти до визначення

Якщо курсор перебуває у межах оголошення об'єкта, буде відкрито документ, що містить відповідне визначення об'єкта, активовано цей документ, а курсор буде пересунуто на початок визначення.

4.9 Додаток попереднього перегляду документів

4.9.1 Вступ

За допомогою цього додатка реалізовано інтерактивний попередній перегляд поточного редагованого документа у остаточному форматі на бічній панелі. Отже, редагуючи, наприклад, текст у форматі Markdown або зображення SVG, ви одразу бачитимете результати поряд із початковим текстом.

Для показу даних додаток використовує додаток KParts, який визначено як бажаний для типу MIME документа. Якщо додатка KParts для відповідного типу документа не буде знайдено, попередній перегляд не буде показано.

Щоб змінити бажаний додаток, відкрийте вікно модуля **Прив'язка файлів** програми «Системні параметри» і змініть значення **Порядок пріоритету служб** на вкладці **Вмонтовані**.

Тип MIME	Додаток KParts
Текст Markdown	KMarkdownWebViewPart або OkularPart
Зображення SVG	SVGPart
Файли інтерфейсу користувача Qt™	KUIViewerPart
Файли графів Dot	KGraphviewerPart

Табл. 4.1: Деякі доступні додатки KParts

4.9.2 Структура меню

Перегляд → Вікна інструментів → Показати Попередній перегляд

Увімкнути або вимкнути показ панелі попереднього перегляду документа Kate на бічній панелі вікна програми.

4.9.3 Інтерфейс

За допомогою кнопок у верхній частині панелі попереднього перегляду можна виконувати такі дії:

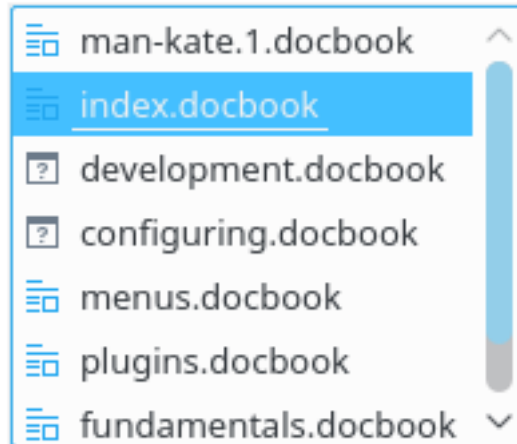
- Заблокувати попередній перегляд вказаного документа. Якщо ви скористаєтеся цим пунктом, при перемиканні фокуса перегляду на інший документ у тому самому вікні Kate попередній перегляд не буде оновлено вмістом нового документа, — у ньому лишатиметься показаним попередній, заблокований документ.
- Увімкнути або вимкнути оновлення попереднього перегляду вмісту поточного документа
- Вручну оновити попередній перегляд вмісту поточного документа
- Спадає меню із пунктами дій додатка KParts

4.10 Додаток перемикання документів

4.10.1 Структура меню

Перегляд → Попередні перегляди (Ctrl+Tab), Перегляд → Попередні перегляди (у зв. порядку) (Ctrl+Shift+Tab)

Відкриває список останніх переглянутих документів:



Тримайте натиснутою клавішу **Ctrl** і натискайте клавішу **Tab**, щоб циклічно переходити пунктами списку вперед. Додаткове натискання клавіші **Shift** надає змогу пересуватися списком у зворотньому напрямку.

Якщо утримувати натиснутою комбінацію клавіш **Ctrl+Tab**, можна скористатися клавішами **↑**, **↓**, **Home** та **End** для навігації списком. Натисканням клавіші з літерою можна циклічно переходити між усіма пунктами у списку із відповідною першою літерою. Після відпускання натиснутих клавіш програма негайно перемкнеться на поточний позначений документ зі списку.

4.11 Навігатор файловою системою

Навігатор файлової системи — це переглядач тек, за допомогою якого ви зможете відкривати файли з показаної у навігаторі теки у поточному фреймі.

4.11.1 Структура меню

Перегляд → Вікна інструментів → Показати Навігатор файлової системи

Вмикає або вимикає показ переглядача файлової системи Kate.

4.11.2 Інтерфейс

Перелічимо всі елементи навігатора файлової системи, починаючи згори:

Панель інструментів

На цій панелі містяться стандартні кнопки інструментів навігації:



Назад

За допомогою цієї кнопки можна повернути переглядач тек до перегляду попередньої теки з журналу перегляду. Якщо попередніх тек у журналі немає, цю кнопку буде вимкнено.



Вперед

За допомогою цієї кнопки можна повернути переглядач тек до перегляду наступної теки з журналу перегляду. Якщо наступних тек у журналі немає, цю кнопку буде вимкнено.



Закладки

Ця кнопка відкриває підменю, за допомогою якого ви змінити або додати закладку, а також створити нову теку закладок.



Поточна тека документа

За допомогою цієї кнопки можна перевести фокус переглядача тек на теку, де зберігається файл поточного документа, якщо це можливо. Якщо поточний документ є новим, ще не збереженим, або теку, де зберігається файл документа не може бути визначено, цю кнопку буде вимкнено.



Параметри

Стислий вигляд

За натискання цієї кнопки у переглядачі буде показано лише назви файлів.

Детальний вигляд

За натискання цієї кнопки у переглядачі буде показано дані про **Назву**, **Дату створення** і **Розмір** файлів.

Ієрархічний вигляд

Подібний до скороченого вигляду, але теки можна розгортати для перегляду їхнього вмісту.

Детальний ієрархічний вигляд

У цьому режимі також можна розгортати теки, але буде показано додаткові стовпчики даних з детального вигляду.

Показувати приховані файли

Показує файли, які зазвичай приховуються вашою операційною системою.

Автоматично синхронізувати з поточним документом

Якщо позначено цей пункт, інструмент перегляду файлової системи автоматично переходитиме до теки поточного відкритого документа у разі зміни цього документа.

Запис розташування

Показує панель послідовної навігації до поточної відкритої теки, подібно до Dolphin. Для переходу до будь-якої з тек просто клацніть на її назві. Ви також можете натиснути кнопку зі стрілочкою поряд з назвою теки, щоб відкрити список підтек цієї теки. Натискання першого з пунктів навігації з зображенням піктограми певного місця у вашій системі відкриє список панелі «Місця».

Ви можете клацнути лівою кнопкою миші праворуч від поля послідовної навігації і ввести шлях до теки, вміст якої слід показати у переглядачі тек. У рядку URL-адреси передбачено підказку у вигляді списку раніше введених адрес. Ви можете обрати одну з цих адрес за допомогою натискання кнопки зі стрілочкою, розташованої праворуч від рядка адреси.

ПІДКАЗКА

У рядку URL-адреси передбачено автоматичне доповнення назв тек. Спосіб доповнення можна обрати за допомогою контекстного меню поля адреси (викликати це меню можна за допомогою клацання правою кнопкою миші).

Перегляд тек

Це стандартне вікно перегляду тек KDE.

Поле фільтрування

За допомогою поля фільтрування ви зможете ввести шаблон для файлів, які буде показано у переглядачі тек. У шаблоні можна використовувати стандартні замітники;

декілька шаблонів слід відокремлювати один від одного пробілами. Приклад: *.crr
*.h *.moc

Щоб було показано всі файли, просто введіть у поле фільтрування символ зірочки, *.

У полі фільтрування передбачено запам'ятовування останніх 10 шаблонів, щоб скористатися одним з них, натисніть кнопку зі стрілочкою, розташовану праворуч від поля, і виберіть потрібний рядок шаблону зі списку. Ви можете вимкнути фільтр натисканням кнопки **Вилучити текст**, розташованої ліворуч від кнопки автоматичного доповнення.

4.11.3 Налаштування

Налаштувати цей додаток можна на сторінці **Навігатор файлової системи** діалогового вікна [налаштування Kate](#).

Панель інструментів

Тут можна налаштувати перелік кнопок на панелі навігатора файлової системи: пересуньте потрібні кнопки, які ви бажаєте додати на панель, до списку **Вибраних дій**, впорядкувати кнопки можна за допомогою кнопок зі стрілочками, розташованих збоку від списку.

4.12 Список документів

4.12.1 Вступ

На панелі списку документів ви побачите список всіх документів, які зараз відкрито у Kate. Поруч з позначками файлів з незбереженими змінами буде показано невеличку піктограму **дискети**, яка позначатиме їх стан.

Над списком документів розташовано панель інструментів із такими кнопками:

-  — створити документ
-  відкрити наявний документ
-  — попередній документ
-  — наступний документ
-  — зберегти поточний документ
-  — зберегти поточний документ із новою назвою

Типово список документів буде показано у **Ієрархічному режимі**, у якому програма показує структуру тек навколо поточних відкритих документів. Ним також можна скористатися у **Режимі списку**, у якому програма показує простий список всіх відкритих документів. Перемикнувшись між режимами перегляду можна клацанням правою кнопкою на списку з наступним вибором режиму зі списку **Режим перегляду**.

Якщо буде відкрито декілька файлів з однаковою назвою (розташованих у різних теках) у **Режимі списку**, перед назвою другого з цих файлів ви побачите цифру «(2)», інші файли з тією ж назвою нумеруватимуться послідовно. У контекстній підказці файла буде показано назву файла повністю, разом зі шляхом до теки, де він знаходиться, отже вам буде просто обрати потрібний вам файл зі списку.

Щоб програма показала документ у поточному активному фреймі, натисніть позначку документа у списку.

У контекстному меню буде показано деякі з пунктів дій меню **Файл**.

Крім того, передбачено пункти дій з керування файлами для перейменування та вилучення файла. За допомогою пункту **Копіювати адресу** можна скопіювати повний шлях до документа до буфера обміну даними.

Ви можете впорядковувати список у декілька різних способів за допомогою пункту **Впорядкувати за** контекстного меню, яке відкривається після клацання правою кнопкою миші, коли вказівник миші знаходитиметься у списку. Серед можливих варіантів впорядкування:

Назва документа

Список буде побудовано в алфавітному порядку назв файлів.

Адреса документа

Список буде побудовано в алфавітному порядку адрес файлів.

Порядок відкриття

Список буде побудовано за порядком відкриття документів.

Типово, у списку документів буде показано порядок роботи з документами за допомогою підсвічування у списку документів, з якими ви працювали нещодавно, кольором. Якщо документ було змінено, до кольору буде додано відповідний відтінок. Документи з якими ви працювали останніми буде показано яскравішим кольором, отже вам буде простіше знайти документи, над якими ви працюєте. Цю можливість можна вимкнути за допомогою [сторінки документів](#) діалогового вікна налаштування програми.

Типово, список документів у вікні Kate розташовано у лівій частині області редагування.

4.12.2 Структура меню

Перегляд → Попередній документ (Alt+↑)

Відкриває документ, показаний у списку документів перед поточним відкритим документом.

Перегляд → Наступний документ (Alt+↓)

Відкриває документ, показаний у списку документів після поточного відкритого документа.

Перегляд → Показати активні

Показує поточний відкритий документ у списку документів.

4.12.3 Налаштування

Затінення тла

За допомогою цього розділу ви можете увімкнути або вимкнути візуалізацію затінення останніх дій із документом та вибрати кольори, які буде використано для затінення, якщо його увімкнено.

Впорядкувати за

Тут ви можете встановити спосіб впорядкування списку документів. Ви також можете встановити порядок за допомогою контекстного меню списку документів.

Режим перегляду

За допомогою цього пункту можна вибрати один з двох варіантів показу даних на панелі документів. Якщо буде вибрано варіант **Ієрархічний перегляд**, список документів буде показано у форматі дерева, де списки документів будуть гілками пунктів тек, у яких ці документи зберігаються. Якщо ж буде вибрано варіант **Перегляд списком**, програма показуватиме плоский список документів.

Показати повний шлях

Якщо буде увімкнено режим ієрархічного перегляду, записи тек на панелі документів буде показано з повним шляхом у файловій системі у назві теки. Ці дані не буде показано у режимі перегляду списком.

Показати панель інструментів

Якщо увімкнено ієрархічний перегляд і позначено цей пункт, над списком документів буде показано панель інструментів із кнопками, подібним до кнопки **Зберегти**. Зніміть позначку з цього пункту, якщо панель інструментів має бути приховано.

Показати кнопку закриття

Якщо позначено цей пункт, Kate при наведенні на пункти відкритих документів показуватиме кнопку закриття.

4.13 Додаток GDB

4.13.1 Вступ

Додаток GDB до Kate є простою оболонкою для будь-якого засобу діагностики, у якому передбачено підтримку [Debugger Adapter Protocol](#). Це, зокрема, GNU Project Debugger або GDB, як це [описано тут](#).

ВАЖЛИВО

Користувачам додатка варто спочатку добре ознайомитися з самим GDB. Щоб дізнатися більше про те, як користуватися GDB, зверніться до [сайта GDB](#).

Увімкнути додаток GDB можна за допомогою [розділу «Додатки» діалогового вікна налаштувань Kate](#).

ПІДКАЗКА

Якщо виконуваний файл збирається за допомогою `gcc/g++`, варто скористатися під час збирання параметром командного рядка `-ggdb`.

Після виконання всіх попередніх етапів збирання відкрийте файл коду у Kate, виберіть «профіль засобу діагностики», вкажіть адресу виконуваного файла за допомогою вкладки **Параметри** панелі **Панель налагоджування** і виберіть у меню пункт **Налагоджування** → **Почати налагоджування**, щоб розпочати сеанс налагоджування.

«Профіль засобу діагностики» визначає використаний DAP-сервер (наприклад, GDB) та спосіб запуску цього сервера. Типовим випадком є запуск сервером процесу, як зазначено вище, але сервер також може встановлювати з'єднання із запущеним процесом (у цьому випадку слід вказати ідентифікатор процесу, а не виконуваний файл). Також можуть бути інші режими, специфічні для мови та DAP-сервера. Щоб дізнатися більше, див. далі додаткові відомості та подробиці щодо налаштування.

4.13.2 Структура меню і панелі інструментів

Доступ до всіх пунктів дій можна отримати за допомогою меню Kate. Крім того, скористатися значною частиною цих пунктів можна за допомогою панелі інструментів налагоджування.

Перегляд → Вікна інструментів → Панель налагоджування

Показує панель з даними, виведеними GDB, дані щодо використаного командного рядка GDB та інших параметрів.

Перегляд → Вікна інструментів → Показати Локальні і стек викликів

Показує список всіх завантажених змінних та їхніх значень, а також зворотне трасування GDB.

Налагоджування → Призначення

Підменю зі списком призначень (виконуваних файлів).

Налагоджування → Почати налагоджування

Запускає GDB для обробки певного виконуваного файла.

Налагоджування → Перервати або припинити налагоджування

Зупиняє роботу GDB.

Налагоджування → Перезапустити налагоджування

Перезапускає GDB.

Налагоджування → Встановити/Зняти точку зупинки

Встановити або зняти точку зупинки у поточному розташуванні курсора.

Налагоджування → Увійти

Виконати поточну інструкцію (буде виконано діагностику виклику функції).

Налагоджування → Переступити

Виконати поточну інструкцію (діагностику виклику функції не буде виконано).

Налагоджування → Вийти

Поновлює виконання, аж до моменту переривання програми, яка виконується.

Налагоджування → Пересунути РС

Пересунути лічильник програми (наступне виконання).

Налагоджування → Виконати до курсора

Запускає виконання програми до інструкції, у якій зараз перебуває курсор.

Налагоджування → Продовжити

Ігнорує всі точки зупинки і виконує програму аж до її переривання (у режимі успішного виконання або у режимі аварійної зупинки).

Налагоджування → Вивести значення

Виводить значення змінної, на якій перебуває курсор у тексті програми.

Параметри → Показані панелі → Додаток GDB

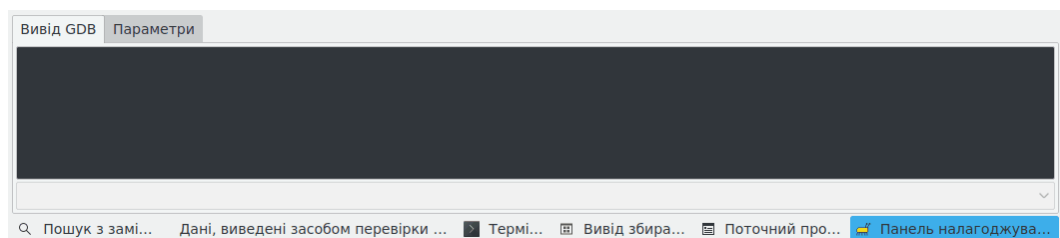
Показати панель налагоджування.

4.13.3 Панель налагоджування

На панелі **Панель налагоджування** буде показано декілька вкладок:

Вивід GDB

Містить дані, отримані від GDB, та командний рядок GDB.



На вкладці **Вивід** буде показано повідомлення, виведені під час виконання сеансу налагоджування.

Параметри

Виконуваний файл

Шлях до призначення (виконуваного файлу) для виконання діагностики.

Робочий каталог

Поточний робочий каталог, вказаний виконуваному файлу (призначенню).

Аргументи

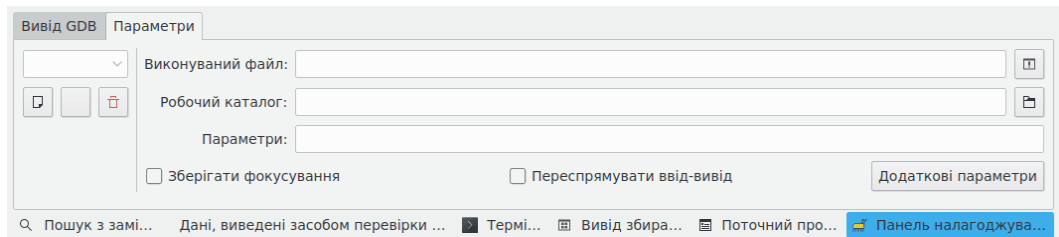
Аргументи, передані програмі.

Зберігати фокусування

Зберігає фокусування на командному рядку GDB.

Переспрямувати ввід-вивід

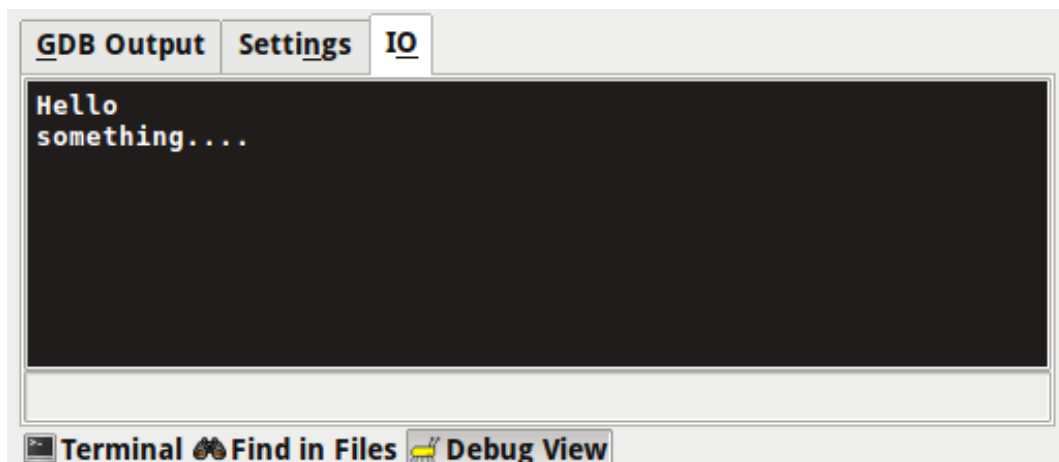
Відкриває нову вкладку **Ввід-вивід** на панелі **Панель налагоджування**. За допомогою цієї вкладки ви зможете переглядати виведені дані та вводити дані, потрібні запусненій програмі.



У діалоговому вікні **Параметри** буде показано налаштування сеансу налагоджування (діагностики).

Ввід-вивід

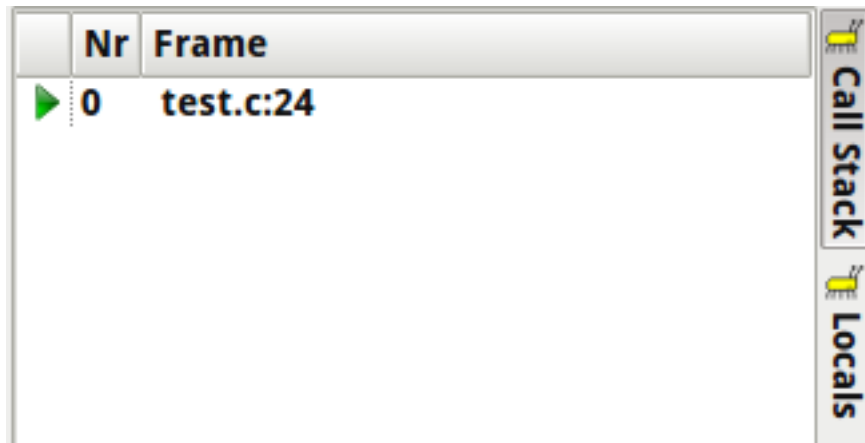
Містить область з даними, виведеними запусненою програмою, та запрошенням командного рядка, за допомогою якого ви можете вводити дані, потрібні програмі.



На вкладці **Ввід-вивід** показано повідомлення, виведені простою тестовою програмою.

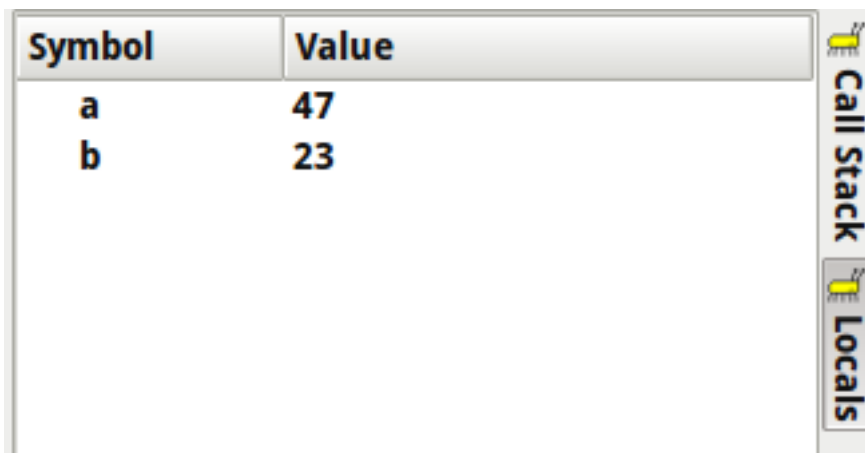
4.13.4 Стек викликів і локальні змінні

На панелі **Стек викликів** буде показано форматовані дані зворотного трасування, отримані від GDB.



Панель інструмента **Стек викликів** додатка GDB.

На панелі інструмента **Локальні змінні** буде показано список всіх завантажених зараз змінних програми та їхніх значень.



Панель інструмента **Локальні змінні** додатка GDB.

4.13.5 Подяки

Особлива подяка учаснику Google Code-In 2011 Martin Gergov за написання більшої частини цього розділу.

4.13.6 Налаштування

На сторінці налаштовування додатка визначаються «профілі засобу діагностики», які можна вибрати. Там буде показано типові налаштування (JSON), їх можна «перезаписати» наданими користувачем у подібній формі. Приклад:

```
{
  "dap": {
    "debugpy": {
      "url": "https://github.com/microsoft/debugpy",
      "run": {
        "command": ["python", "-m", "debugpy", "--listen", "${# ←
                      run.port}", "--wait-for-client"],
        "port": 0,
      }
    }
  }
}
```

Буде застосовано різні етапи перевизначення і об'єднання; налаштування користувача (буде завантажено з файла) перевизначають (внутрішні) типові налаштування і перезаписуються даними запису у налаштуваннях проєкту `.kateproject`.

4.13.6.1 Налаштовування середовища виконання

Базові відомості та специфіку можна знайти у розділі [Середовище виконання клієнта LSP](#) нижче. Але варто зауважити, що можливо виникне потреба у запуску процесу для діагностики (і сервера DAP) у «спеціальному середовищі», визначеному або змінними середовища або певним контейнером (що надає потрібні залежності та обставини для належного виконання).

Подібно до прикладу у відповідному розділі, такі налаштування може бути задано у `.kateproject`.

```
{
  // це також може бути масив об'єктів
  "exec": {
    "hostname": "foobar"
    // команда також може бути масивом рядків
    "prefix": "podman exec -i foobarcontainer",
    "mapRemoteRoot": true,
    "pathMappings": [
      // можлива будь-яка з вказаних нижче форм
      // також існує більш автоматизована альтернатива, див. нижче
      [ "/dir/on/host", "/mounted/in/container" ]
      { "localRoot": "/local/dir", "remoteRoot": "/remote/dir" }
    ]
  },
  "lspclient": {
    "servers": {
      "python": {
        // встановлюватиме відповідність або об'єднує з вищим об ←
        'єктом
        "exec": { "hostname": "foobar" },
        // обмежити цей сервер коренем цього проекту,
        // щоб його не було використано для інших проектів, які ←
        може бути відкрито
        // (інші сервери можуть вже використовувати певні корене ←
        ві проекти, але, загалом, для python це не справджує ←
        ться)
        "root": "."
      },
      "c": {
        // як вище
        "exec": { "hostname": "foobar" },
        "root": "."
      }
    }
  }
}
```

Докладніші відомості наведено у згаданому розділі, але по суті префікс `prefix` буде додано перед командним рядком сервера DAP, вказаним в іншому місці. Результат полягає у тому, що сервер запускається у вказаному контейнері, а потім, у свою чергу, також і запущений процес. `pathMapping` забезпечує перетворення шляхів до файлів між панеллю перегляду редактора та панеллю сервера DAP (контейнера), наприклад, під час роботи з встановленням точок зупинки або обробкою наданого зворотного трасування. Зауважте, що таке зіставлення є необов'язковим і може бути корисним або ні. Під час роботи з кодом C/C++, який компілюється в основній системі, символічна інформація посилається на вихідні файли в основній системі, яких взагалі не існує в іншому середовищі. Однак, в інших обставинах (наприклад, python) посилання виконується на фактичні файли середовища виконання (в іншому середовищі).

Слід пам'ятати про наведене нижче:

- У середовищі або контейнері має бути сервер DAP, який має бути налаштовано для підтримки належних дій з діагностики (отже, якщо потрібно, привілейовано, з потрібними можливостями).
- Має бути можливим обмін даними між редактором і DAP. У випадку контейнера останній має або використовувати з'єднання з мережею основної системи, або надавати додатний прив'язаний або оприлюднений порт разом із відповідним фрагментом налаштувань, як у наведеному вище прикладі (оскільки вибраний автоматично у випадку порту 0 не працюватиме).
- Вказаний виконуваний файл або ідентифікатор процесу має бути у перспективі «контейнера», а також будь-які аргументи (діагностованого виконуваного файла).

Також, як у випадку LSP, буде встановлено певні змінні середовища; `KATE_EXEC_PLUGIN` буде встановлено у значення `dap`, `KATE_EXEC_SERVER` буде встановлено у значення типу засобу діагностики або мови (наприклад, `python`), а `KATE_EXEC_PROFILE` буде встановлено у значення запису налаштувань (наприклад, `launch`)."

4.14 Додаток проєктів

4.14.1 Вступ

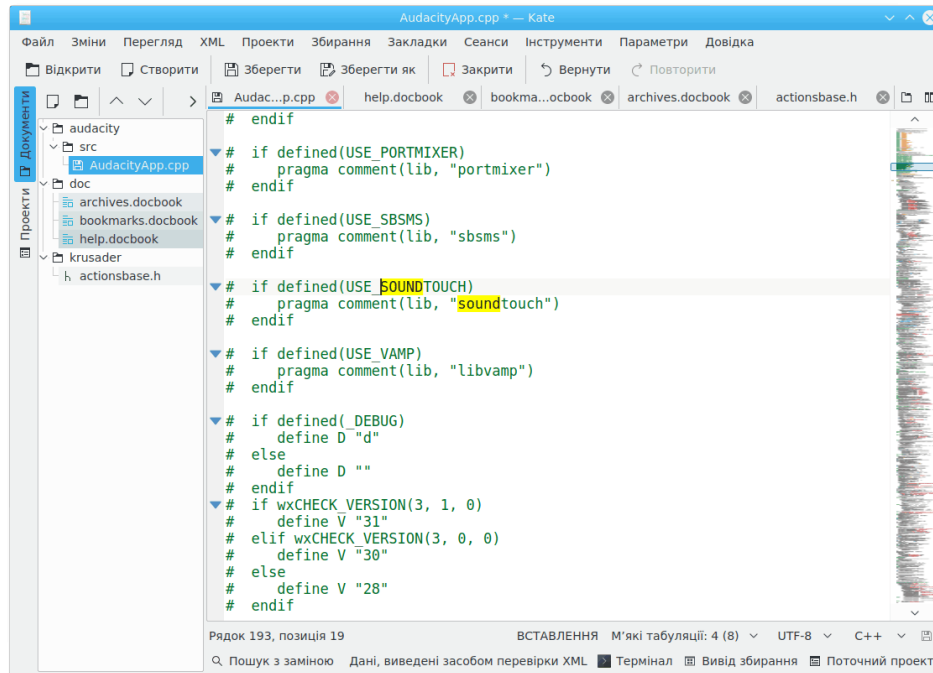
Основне призначення додатка роботи з проєктами — створення структурованого списку файлів, які належать до проєкту. Реалізовано такі можливості:

1. Надання структурованого перегляду файлів проєкту
2. Спрощення і пришвидшення відкриття проєктів і перемикання між проєктами
3. Підтримка пошуку із заміною у межах проєкту
4. Підтримка простого механізму автоматичного доповнення коду
5. Спрощення швидкого відкриття файлів у проєкті
6. Підтримка збирання проєкту

4.14.2 Структурований перегляд файлів

Після вмикання додатка роботи з проєктами за допомогою вікна налаштування Kate відкрийте файл з проєкту. У відповідь на бічний панелі буде показано список усіх проєктів та файлів у цих проєктах:

Підручник з Kate



Як можна бачити у нашому прикладі, поточним активним проектом є «Kate», а його вміст показано у форматі ієрархічного списку. Якщо натиснути пункт файлу у такому списку, програма відкриє його для редагування. Крім того, передбачено контекстне меню, за допомогою якого можна відкривати файли у інших програмах, зокрема можна відкривати файли .ui за допомогою Qt Designer.

Фільтрувати список можна ввівши частину назви файлу, який ви шукаєте у поле для пошуку, розташоване під списком.

4.14.3 Перемикання проектів

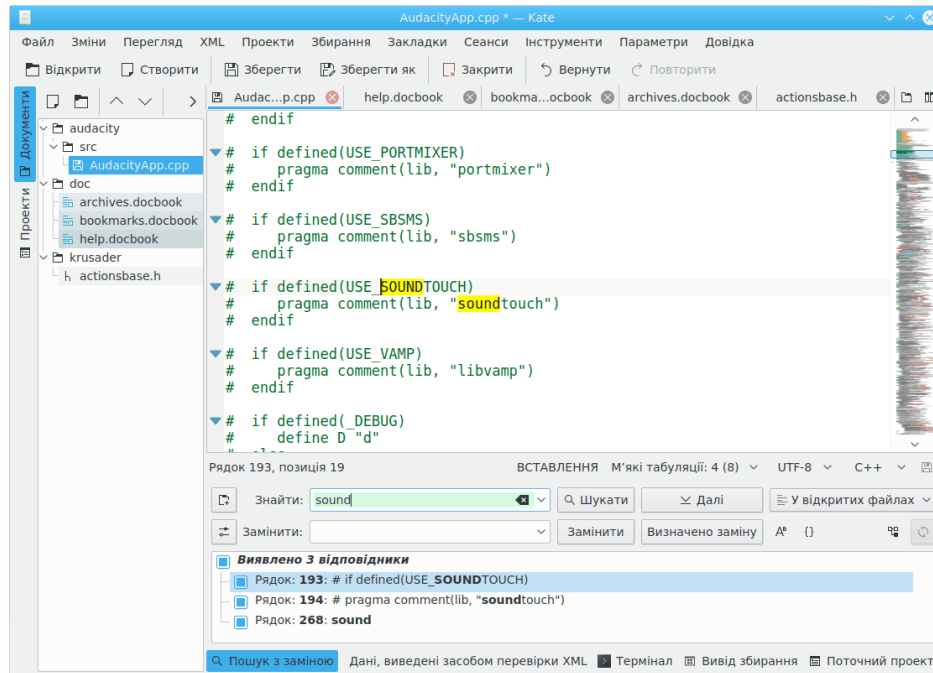
Ідея роботи додатка полягає у тому, що вам ніколи не доведеться відкривати проект вручну (таку можливість, зрештою, взагалі не передбачено). Якщо ви відкриєте якийсь файл, додаток роботи з проектами швидко виконає сканування теки, де він зберігається, і усіх її батьківських тек, і спробує знайти файл .kateproject. Якщо такий файл буде знайдено, програма автоматично завантажить відповідний проект.

Крім того, якщо ви відкриєте якийсь інший документ, що належить до іншого проекту, у Kate, додаток роботи з проектами автоматично змінить поточний проект на проект ново-відкритого файлу. Отже, без потреби у додаткових діях, завжди буде відкрито відповідний проект. Звичайно ж, ви також можете перемкнути поточний активний проект за допомогою спадного списку.

4.14.4 Пошук і заміна у проектах

У Kate передбачено додаток пошуку із заміною, сторінку якого буде показано на нижній панелі. Якщо відкрито проект, ви можете розгорнути панель пошуку із заміною і перемкнути додаток на режим пошуку із заміною у поточному проекті:

Підручник з Kate

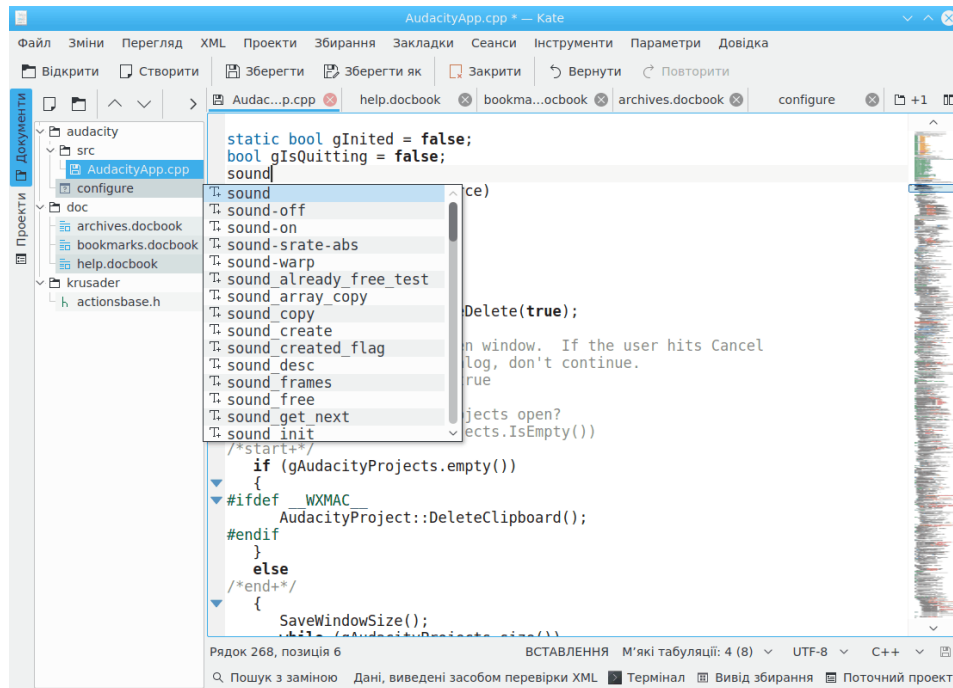


4.14.5 Просте автоматичне доповнення

Маючи дані щодо усіх файлів, які належать до проекту, додаток роботи з проектами надає можливості з автоматичного доповнення коду, реалізовані на основі CTags. Під час першого відкриття проекту CTags виконує обробку усіх файлів проекту у фоновому режимі і зберігає усі дані CTags до каталогу /tmp. Отримані дані потім використовуються для заповнення списку контекстної панелі автоматичного доповнення в Kate.

Якщо ж відповідні можливості автоматичного доповнення виявляються недоступними, Kate зможе показувати у списку автоматичного доповнення лише пункти, отримані на основі даних слів з поточного файла. Таким чином, можливості автоматичного доповнення, реалізовані у додатку роботи з проектами є набагато потужнішими за стандартні.

Підручник з Kate



Якщо у системі не буде виявлено програм CTags, Kate повідомить вас про це панелью нагадування. Крім того, варто зауважити, що файл CTags у `/tmp` спорожнюється кожного разу, коли ви завершуєте роботу Kate. Отже, додаток не засмічуватиме теку зайвими застарілими файлами.

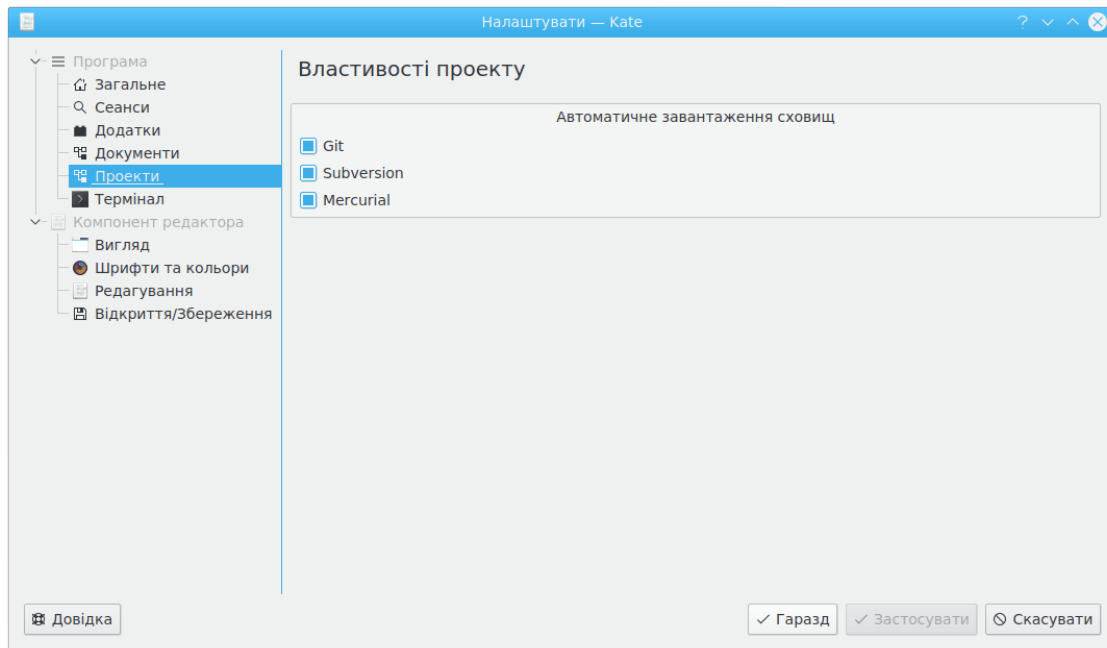
4.14.6 Підтримка збирання проекту

Ще однією можливістю є підтримка автоматичного налаштування [дodatka збирання](#).

4.14.7 Створення проектів

4.14.7.1 Автоматичне завантаження проектів

У додатку роботи з проектами передбачено можливість автоматичного завантаження. Список файлів формується на основі даних системи керування версіями. Для того, щоб скористатися цією можливістю, вам слід увімкнути відповідну систему керування версіями у параметрах програми (типово увімкнено):



4.14.7.2 Створення проєктів вручну

Вам достатньо створити файл `.kateproject` у кореневій теці проєкту. Наприклад, для «Kate» файл `.kateproject` містить такі дані:

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

Вміст файла записується із використанням синтаксису JSON. Проєкт називається «Kate», а список файлів, які у ньому містяться, отримується з даних git.

Окрім підтримки `git`, передбачено підтримку `subversion (svn)` та `mercurial (hg)`. Якщо ви не хочете, щоб додаток читав файли із системи керування версіями, ви можете викликати `kate` з командного рядка так:

```
kate /шлях/до/теки
```

або наказати додатку рекурсивно завантажити файли з каталогів так:

```
{
  "name": "Kate",
  "files": [
    {
      "directory": "kate",
      "filters": [
        "*.cpp",
        "*.h",
        "*.ui",
        "CMakeLists.txt",
        "Find*.cmake"
      ]
    }
  ]
}
```

```
    ],
    "recursive": 1,
    "hidden": 1
  }
],
"exclude_patterns" : [
  "^build/.*"
]
}
```

Тут підтеки та фільтри визначають частину проекту. Ви також можете поєднувати керування версіями та керування на основі файлів у фільтрах. Приховані файли не буде отримати, якщо для параметра

"hidden" встановлено значення 0. Параметр

"exclude_patterns" є списком вірців у формі формальних виразів, яким можна скористатися для виключення тек і файлів з ієрархії проекту. У нашому прикладі буде виключено усі файли і теки у каталозі build кореневого каталогу.

Якщо вам потрібно додати підтримку збирання, ви можете вписати до файла .kateproject такі рядки:

```
{
  "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"
      }
    ]
  }
}
```

Визначені вище цілі збирання потім буде показано у додатку збирання, розділ «**Цілі додатка проекту**». Якщо визначено масив

targets ,

build ,

clean і

install буде проігноровано. Кожен елемент у цьому масиві визначає ціль.

name є назвою цілі,

build_cmd буде використано для збирання цілі,

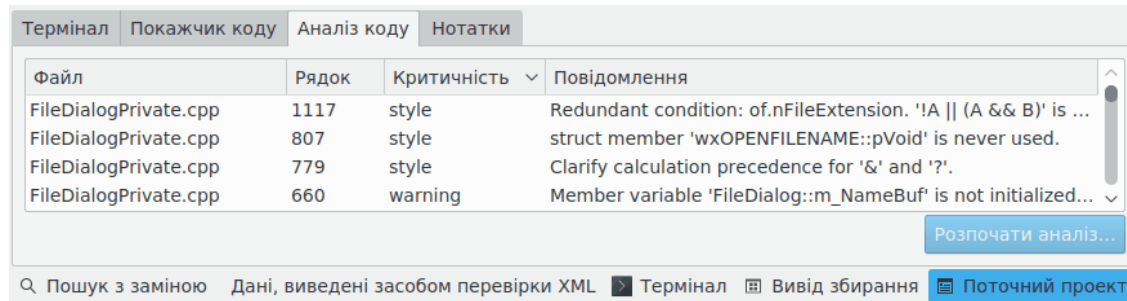
run_cmd буде використано для запуску цілі. Найважливішим записом є

directory , — це місце, де буде виконано команди.

Якщо у вашому проєкті є файл `.kateproject`, за яким стежить система керування версіями, але вам потрібно скоригувати налаштування певного робочого простору, ви можете зберегти ці зміни до окремого файла із назвою `.kateproject.local`. Вміст цього файла матиме вищий пріоритет за вміст `.kateproject`.

4.14.8 Поточний проєкт

За допомогою пункту меню **Проєкти** → **Перейти** (**Alt+1**) ви можете відкрити панель **Поточний проєкт** у нижній частині вікна редактора із такими чотирма вкладками:



Панель термінала

Запуск емулятора термінала у кореневій теці проєкту.

Покажчик коду

Після введення перших символів назви об'єкта у поле для пошуку програма спробує знайти відповідні об'єкти мови за назвою. Буде показано відповідні функції, класи, символи тощо разом із даними щодо їхнього типу та назви відповідного файла та номера рядка.

Позначте пункт у списку, щоб перейти до відповідного рядка у файлі із початковим кодом.

Аналіз коду

Натисніть пункт **Розпочати аналіз**, щоб запустити статичний аналіз коду мовами програмування C та C++ за допомогою **cppcheck** і отримати звіт із показаними назвами файлів, номерами рядків та рівнем критичності помилки (стильова, попередження тощо) та даними щодо знайденої вади.

Позначте пункт у списку, щоб перейти до відповідного рядка у файлі із початковим кодом.

Нотатки

Текст, введений на цій вкладці, буде збережено у файлі `.kateproject.notes`.

4.14.9 Меню «Проєкти»

За допомогою меню **Проєкти** ви зможете перемикатися між поточними відкритими проєктами. Відповідальним за цей пункт меню є додаток «Проєкти».

Проєкти → Назад (**Ctrl+Alt+←**)

Перемкнутися на попередній проєкт.

Проєкти → Вперед (**Ctrl+Alt+→**)

Перемкнутися на наступний проєкт.

Проєкти → Перейти до (**Alt+1**)

Відкрити панель **Поточний проєкт** у нижній частині вікна редактора.

4.15 Додаток клієнта LSP

За допомогою додатка клієнта LSP можна отримати доступ до багатьох можливостей, зокрема автоматичного доповнення коду, навігації кодом та пошуку довідкової інформації на основі протоколу сервера мов (Language Server Protocol).

Після позначення пункту додатка клієнта LSP на сторінці додатків, у діалоговому вікні налаштування Kate з'явиться нова сторінка із назвою «Клієнт LSP».

4.15.1 Структура меню

Нижче, у пояснення щодо пунктів меню, наведено відповідні команди LSP (якщо такі існують). У документації з цих команд можна знайти додаткові відомості та пояснення, які можуть залежати від мови програмування, яку використано. Термін «поточний символ» використано на позначення символу, який відповідає поточній позиції курсора і визначається на основі мови програмування та з використанням відповідної реалізації сервера обробки.

Клієнт LSP → Перейти до визначення

[textDocument/definition] Перейти до визначення поточного символу.

Клієнт LSP → Перейти до оголошення

[textDocument/declaration] Перейти до оголошення поточного символу.

Клієнт LSP → Перейти до визначення типу

[textDocument/typeDefinition] Перейти до визначення типу поточного символу.

Клієнт LSP → Знайти посилання

[textDocument/references] Знайти посилання на поточний символ.

Клієнт LSP → Знайти реалізації

[textDocument/implementation] Знайти реалізації поточного символу.

Клієнт LSP → Підсвітити

[textDocument/documentHighlight] Підсвітити посилання на поточний символ у поточному документі.

Клієнт LSP → Наведення

[textDocument/hover] Показати при наведенні дані щодо поточного символу.

Клієнт LSP → Форматувати

[textDocument/formatting] [textDocument/rangeFormatting] Форматувати поточний документ або поточний позначений фрагмент документа.

Клієнт LSP → Перейменувати

[textDocument/rename] Перейменувати поточний символ.

Клієнт LSP → Швидке виправлення

[textDocument/codeAction, workspace/executeCommand] Обчислює і застосовує швидке виправлення для діагностики у поточній позиції (або рядку).

Клієнт LSP → Показувати документацію щодо доповнення позначеного

Показати документацію щодо позначеного елемента у списку автоматичного доповнення.

Клієнт LSP → Увімкнути довідку щодо підписів із автодоповненням

Крім того, показувати довідку щодо підписів у списку автозавершення.

Клієнт LSP → Включати оголошення до посилань

Надіслати запит щодо включення оголошення поточного символу при запиті щодо посилань.

Клієнт LSP → Додавати дужки при доповненні функцій

Автоматично додати пару дужок після автоматично доповненої назви функції.

Клієнт LSP → Показувати відомості при наведенні

Показати дані, які програма показує при наведенні вказівника миші. Незалежно від значення цього параметра, запит щодо цих даних можна завжди ініціювати вручну.

Клієнт LSP → Форматувати при введенні

[document/onTypeFormatting] Форматувати частини документа, коли користувач вводить певні контрольні символи. Прикладом може бути застосування відступів рядків при введенні символу розриву рядка або іншого форматування, яке визначено на сервері LSP. Зауважте, що скрипти встановлення відступів текстового редактора можуть намагатися виконати те саме завдання (залежно від режиму роботи), тому не радимо одночасно використовувати обидва ці засоби форматування.

Клієнт LSP → Нарощувальна синхронізація документів

Надсилати для обробки сервером окремі частини документа під час редагування, а не цілий документ (якщо підтримку такого режиму обробки передбачено на сервері).

Клієнт LSP → Підсвітити місце переходу

Надати тимчасовий візуальний сигнал після виконання переходу до місця (визначення, оголошення тощо).

Клієнт LSP → Показувати діагностичні сповіщення

[textDocument/publishDiagnostics] Обробити і показати діагностичні сповіщення, які було надіслано сервером.

Клієнт LSP → Показувати підсвічування діагностики

Додати підсвічування фрагментів тексту для діапазонів, які визначено діагностикою.

Клієнт LSP → Показувати позначки діагностики

Додати позначки у документі для рядків, які визначено діагностикою.

Клієнт LSP → Перемикнути на вкладку діагностики

Перемикнути на вкладку діагностики на панелі інструментів додатків.

Клієнт LSP → Закрити усі вкладки, які не є діагностичними

Закрити усі вкладки (наприклад, вкладки посилань) на панелі інструментів додатків, які не пов'язано із діагностикою.

Клієнт LSP → Перезапустити сервер LSP

Перезапустити сервер LSP документа.

Клієнт LSP → Перезапустити усі сервери LSP

Зупинити роботу усіх серверів LSP, які згодом, якщо потрібно, буде запущено або перезапущено.

4.15.2 Підтримка переходу до символів

Клієнт LSP може допомогти вам у переході до будь-якого символу у вашому проєкті або поточному файлі. Щоб перейти до будь-якого символу у файлі, скористайтеся кнопками панелі «Схема символів клієнта LSP» у правій частині вікна Kate. На цій панелі ви побачите списки усіх символів, які було знайдено сервером у поточному документі.

4.15.2.1 Налаштовування схеми символів клієнта LSP

Типово, символи буде упорядковано за їхнім розташуванням у документі, але ви можете наказати програмі упорядкувати їх за абеткою. Для цього клацніть правою кнопкою на панелі і позначте у контекстному меню пункт «Упорядкувати за абеткою».

Типово, на панелі символи буде показано у форматі ієрархічної структури. Втім, ви можете змінити формат на звичайний для списків за допомогою контекстного меню.

4.15.2.2 Підтримка переходу до загальних символів

Щоб перейти до будь-якого символу у вашому проєкті, ви можете відкрити вікно переходу до символу за допомогою комбінації клавіш **Ctrl+Alt+P**. Одразу після відкриття вікно буде порожнім, але щойно ви щось введете з клавіатури, програма почне показувати у вікні відповідні символи. Якість відповідності, а також можливості з фільтрування залежать від використаного вами сервера. Наприклад, для clangd передбачено можливість нечіткого фільтрування, але у деяких інших серверів такої можливості не передбачено.

4.15.3 Інші можливості

Передбачено підтримку перемикавання заголовка початкового коду Clangd. Щоб перемкнути заголовок початкового коду у проєкті C або C++, скористайтеся пунктом «Перемкнути файл заголовка/коду» у контекстному меню або натисніть клавішу **F12**.

Ви можете швидко перейти до символу, навівши на нього вказівник миші і натиснувши комбінації **Ctrl** + ліва кнопка миші.

4.15.4 Налаштування

За допомогою сторінки налаштування додатка ви можете визначити типові значення для описаних вище пунктів меню. Втім, на цій сторінці є один додатковий пункт, за допомогою якого можна вказати файл налаштувань сервера. Цей файл є файлом JSON, яким можна наказати серверу LSP запускатися (і потім вести обмін даними за допомогою stdin/stdout). Для зручності, до типового пакунка включено типовий файл налаштувань, який можна вивчати на сторінці налаштування додатка. Щоб полегшити розуміння наведених нижче пояснень, ось витяг з налаштувань:

```
{
  "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.*$"
    }
  }
}

```

Зауважте, що кожен запис команди («command») може масивом або рядком (який буде поділено на записи масиву). Крім того, буде взято до уваги запис верхнього рівня «global» (поряд із «server»)(докладніший опис наведено нижче). Пошук вказаного виконуваного файлу відбуватиметься у звичний спосіб, наприклад, з використанням PATH. Якщо виконуваний файл встановлено у якомусь нетиповому місці, шлях до цього місця може бути згодом розгорнуто. Крім того, можна скористатися у записі місця (символічним) посиланням або скриптом-обгорткою, який зберігається у звичайному PATH. Як проілюстровано вище, можна також вказати «path», місце, в якому буде виконано пошук після стандартних місць.

В усіх записах «command», «root» та «path» буде виконано розгортання змінних.

Інструкцією «highlightingModeRegex» можна скористатися для прив'язування режиму підсвічування, який використовує Kate, до ідентифікатора мови сервера. Якщо не буде надано формального виразу, буде використано сам ідентифікатор мови. Якщо для запису «documentLanguageId» встановлено значення false, при відкритті документа серверу не буде передано ідентифікатора мови. Це може дати кращі результати для деяких серверів, які є точнішими у визначенні типу документа, ніж засоби на основі режиму Kate.

З наведеного вище прикладу можна зрозуміти суть. Крім того, кожен об'єкт запису сервера може мати запис «initializationOptions», який містить дані, які передаються серверу як частина методу «initialize». Якщо такий запис буде знайдено, параметр «settings» буде передано серверу за допомогою сповіщення «workspace/didChangeConfiguration». Як об'єкт JSON можна вказати «completionTriggerCharacters» або «signatureTriggerCharacters» із рядковими учасниками «exclude» і/або «include». Ці учасники буде використано для, відповідно, виключення або додавання певних символів для відповідного набору перемикачів, який надається сервером.

Застосовуються різні етапи перевизначення та об'єднання.

- Налаштування користувача (завантажені з файла) мають пріоритет над типовими (вбудованими) налаштуваннями.
- Запис «lspclient» у налаштуваннях проєкту `.kateproject` має пріоритет над усіма попередніми записами.
- Отриманий у результаті обробки запис «global» використовується як доповнення (не перевизначення) будь-якого із записів сервера.

Для кожної комбінації (root, servertime) використовується один екземпляр сервера. Якщо значення «root» вказано як абсолютний шлях, його буде використано без обробки. Якщо ж це не так, значення вважатиметься вказаним відносно каталогу «projectBase» (як його визначено у [додатку проєктів](#)), якщо такий існує, або відносно каталогу документа, якщо цього каталогу не існує. Якщо значення не вказано, і «rootIndicationFileNames» є масивом назв файлів, буде використано батьківський каталог поточного документа, який містить такий файл. Крім того, якщо «root» не вказано і «rootIndicationFilePatterns» є масивом вірців назв файлів, буде вибрано батьківський каталог поточного документа, що відповідає вірцю назви файла. Останнім резервним варіантом для «root» є домашній каталог користувача. Для будь-якого документа отриманий у результаті описаної обробки «root» визначає, чи потрібен окремий екземпляр сервера. Якщо такий екземпляр потрібен, значення «root» буде передано як «rootUri/rootPath».

Загалом, рекомендуємо не визначати «root», оскільки він не є важливим для роботи сервера (хоча, ви можете визначити його у вашому конкретному випадку). Якщо екземплярів сервера буде небагато, вони працюватимуть ефективніше і матимуть, очевидно, «ширше» поле зору, ніж у випадку використання багатьох окремих екземплярів сервера.

Як вже згадували вище, у деяких записах буде використано розгортання змінних. Належне використання цієї можливості у поєднанні із підходом із використанням «скрипту-обгортки» забезпечує належні можливості налаштування для широкого спектра застосувань. Наприклад, розглянемо сценарій із мовою розробки python. Нехай маємо декілька проєктів (наприклад, у сховищах git), у кожного з яких є власна конфігурація віртуального середовища. Якщо скористатися типовими налаштуваннями, сервер мови python не отримає даних щодо віртуального середовища. Втім, цю проблему можна розв'язати за допомогою викладеного нижче підходу. По-перше, до «Параметрів сервера користувача» додатка LSPClient слід ввести вказаний нижче фрагмент коду:

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

Запис «root» вище є відносним щодо каталогу проєкту. Його призначено для запуску окремого сервера мов для кожного з проєктів. У цьому випадку такий поділ є необхідним, оскільки кожен проєкт має своє віртуальне середовище.

`pylsp_in_env` — невеличкий «скрипт-обгортка», який має бути збережено до одного з каталогів зі змінної `PATH` із таким вмістом (вміст слід скоригувати за вашими потребами):

```
#!/bin/bash
cd $1
# запустити сервер (python-lsp-server) у віртуальному середовищі
# (тобто із заданими змінними віртуального середовища)
# тому командуємо source для віртуального середовища
source XYZ
# сервер і аргументи можуть бути різними
exec myserver
```

Це лише один приклад більш загального шаблону, з яким, можливо, працювати трохи зручніше, як описано в розділі [Середовище виконання](#) нижче.

4.15.4.1 Налаштування сервера LSP

У кожного окремо взятого сервера LSP є власний спосіб налаштування. У ньому можуть використовуватися специфічні засоби для мови або інструменти, наприклад `tox.ini` (для python), `.clang-format` для форматування у стилі C++. Такі налаштування може бути потім використано іншими (не пов'язаними із LSP) інструментами (зокрема `tox` або `clang-format`). Крім того, деякі сервери LSP завантажують налаштування з нетипових файлів (наприклад, `.cccls`). Далі, нетипові налаштування сервера можна передавати за допомогою LSP (протоколу), див. згадані раніше записи «`initializationOptions`» і «`settings`» у налаштуваннях сервера.

Оскільки застосовуються різні рівні перевизначення і об'єднання налаштувань, у наведеному нижче прикладі визначені користувачем специфічні налаштування клієнта коригують деякі з налаштувань `python-language-server`.

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

На жаль, засоби налаштування серверів LSP часто не мають належної докладної документації. Лише вивчення коду сервера надає змогу ознайомитися із підходами до налаштування та переліком доступних параметрів. Зокрема, у наведеному вище прикладі у сервері передбачено підтримку ширшого спектра параметрів у «`settings`». У [документації до іншого клієнта LSP](#) можна знайти приклади для серверів інших мов та відповідні параметри, які доволі просто перетворити на налаштування JSON, які використовуються у нашому клієнті, і які у загальних рисах описано вище.

4.15.4.2 Форматування при збереженні для сервера LSP

Ви можете увімкнути «форматування при збереженні» у параметрах LSP діалогового вікна налаштування програми.

4.15.4.3 Придушення діагностики на сервері LSP

Може так статися, що діагностичні дані не є аж надто корисними. Діагностика може бути зайвою, особливо, якщо діагностичних повідомлень надто багато (а також вони належать до одного типу). У деяких випадках, потік повідомлень можна скоригувати за мовою програмування (на сервері). Наприклад, [механізм налаштування clangd](#) уможливорює коригування деяких аспектів діагностики. Втім, загалом, спосіб коригування не завжди очевидний, або коригування взагалі неможливе у бажаний спосіб через обмеження роботи сервера або вади.

Через це, у додатку передбачено підтримку придушення діагностичних повідомлень, подібне до, наприклад, придушення у `valgrind`. Можна виконати найточніше налаштування за допомогою ключа «`suppressions`» у (об'єднаних) налаштуваннях JSON.

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

Кожен (коректне) правило може мати довільну назву і бути визначеним масивом довжини 2 або 3, у якому вказано формальний вираз для встановлення відповідності (повній) назві файлу, формальний вираз для встановлення відповідності діагностичному тексту та необов'язковий формальний вираз для встановлення відповідності (діапазону початкового коду) тексту, до якого застосовується діагностика.

Окрім вказаних вище точних налаштувань, у контекстному меню вкладки діагностики також передбачено підтримку додавання та вилучення придушень. При цьому відповідність діагностичному тексту встановлюється точно на загальному рівні (будь-який файл) або на локальному рівні (певний вказаний файл). Дані придушень зберігаються у налаштуваннях сеансу і завантажуються з них же.

4.15.4.4 Діагностика сервера LSP

Одна річ описати як налаштувати (нетиповий) сервер LSP для будь-якої окремої мови програмування, інша ж — зробити так, щоб сервер працював без будь-яких проблем. На щастя, зазвичай, маємо безпроблемну роботу сервера. Втім, іноді можуть виникати проблеми або із певними «тупими» помилками у налаштуваннях або фундаментальні проблеми із самим сервером. Другий варіант може проявляти себе як повідомлення про невдалі спроби запустити сервер, які з'являються на вкладці виведення даних Kate. Втім, такі повідомлення є лише повідомленнями верхнього рівня або повідомленням поступу обробки, а не докладними діагностичними повідомленнями, і навіть можуть не містити даних щодо того, що є, фактично, іншим процесом (сервером LSP).

Звичайним способом діагностики є додавання певних прапорців до команди запуску (сервера мови), які увімкнуть (додаткове) журналювання (до якогось файлу або стандартного виведення помилок), якщо цього не зроблено типово. Якщо потім запустити Kate з терміналу, можна отримати додаткові дані щодо джерела помилок.

Також дані можна отримати вивченням обміну даними між клієнтом LSP Kate і сервером LSP. Знову ж таки, на останньому, зазвичай, є засоби трасування. Крім того, клієнт LSP надає додаткове діагностичне трасування (до `stderr`), якщо викликати Kate із належним чином експортованою змінною середовища `LSPCLIENT_DEBUG=1`.

4.15.4.5 Налаштування середовища виконання

Наведений вище приклад `virtualenv python` — це лише один із прикладів «середовища виконання», яке працює окремо від звичайного середовища основної системи. Цього можна

досягти за допомогою різних налаштувань змінних (наприклад, `virtualenv`), або налаштування `(s)chroot` (перемикання на інший каталог як новий кореневий каталог), контейнера (наприклад, `podman`, `docker`) або `ssh`-сеансу на іншу основну систему. У кожному випадку «інше середовище» визначається «префіксом виконання». Тобто, деяку програму можна викликати/запустити в іншому середовищі за допомогою «префікса» (програми та аргументів), доданого до передбачуваного виклику. Наприклад, `podman exec -i назва_контейнера або ssh користувач@вузол`.

Зокрема, як і в попередньому прикладі `virtualenv`, можна вибрати/потрібно запустити LSP-сервер в окремому середовищі (наприклад, контейнері з усіма залежностями, необхідними для певного проєкту). «Ручний» підхід, описаний вище, має недолік у тому, що він замінює стандартний командний рядок LSP, який потім доводиться вказувати та дублювати знову в скрипті-обгортці. Також у деяких інших згаданих вище прикладах «простір назв шляху» вузла (як його бачить редактор) може відрізнитися від простору назв середовища. Щоб вирішити ці питання більш систематичним способом (ніж «нетиповим» підходом), можна вказати деякі додаткові налаштування.

Наприклад, наведений нижче код можна вказати у налаштуваннях `.kateproject`. Очевидно, «фіктивні» коментарі включати не слід.

```
{
  // це також може бути масив об'єктів
  "exec": {
    "hostname": "foobar"
    // the command could also be an array of string
    "prefix": "podman exec -i foobarcontainer",
    "mapRemoteRoot": true,
    "pathMappings": [
      // either of the following forms are possible
      // a more automagic alternative exists as well, see later/ ←
      below
      [ "/dir/on/host", "/mounted/in/container" ]
      { "localRoot": "/local/dir", "remoteRoot": "/remote/dir" }
    ]
  },
  "lspclient": {
    "servers": {
      "python": {
        // встановлюватиме відповідність або об'єднує з вищим об ←
        'ектом
        "exec": { "hostname": "foobar" },
        // обмежити цей сервер коренем цього проєкту,
        // щоб його не було використано для інших проєктів, які ←
        може бути відкрито
        // (інші сервери можуть вже використовувати певні корене ←
        ві проєкти, але, загалом, для python це не справджує ←
        ться)
        "root": "."
      },
      "c": {
        // as above
        "exec": { "hostname": "foobar" },
        "root": "."
      }
    }
  }
}
```

Отже, що відбувається в результаті вищезазначеного? Як згадувалося, частина `lspclient` вищезгаданого об'єднується із загальними налаштуваннями, отже, буде знайдено розділ `exec`

(для вказаних мов). Буде виконано пошук іншого об'єкта (який вказує на відповідну `hostname`) у розділі `exec` або `lspclient`, і відповідний об'єкт слугуватиме основою для об'єднання. В результаті, командний рядок LSP-сервера (для C та python) буде додано до вказаного (з підставленням змінних) префікса, і тому він буде виконаний у заданому контейнері. Звичайно, контейнер має бути створено, він має перебувати у належному запущеному стані та бути обладнаним належними LSP-серверами. Можливо, у вас виникла спокуса використати розділ `global` (у межах `lspclient`). Це також могло б спрацювати, але тоді *всі* LSP-сервери запустилися б з цим префіксом, включаючи ті, що використовуються, наприклад, для Markdown, Bash-скриптів або JSON. Більш імовірно, що на звичайній основній системі це усе є (якщо ним користуються). Тож, як завжди, це залежить від ваших конкретних налаштувань.

Однак, LSP-сервер тепер може стежити за (віддаленими) шляхами, відмінні від (локальних), які бачить та використовує редактор. Вказані `pathMappings` використовуються для перетворення між ними в межах зв'язку із LSP-сервером. Це, звичайно, відбувається у межах можливого. Зрозуміло, що не всі локальні шляхи мають віддалене представлення, але шляхи, яких немає, не будуть видимі (сервером) і не призведуть до проблем. І навпаки, (віддалений) сервер тепер може бачити та надавати посилання у «віддаленому кореневому каталозі», які не є очевидно/легко представленими у локальній системі. Якщо увімкнено `mapRemoteRoot`, віддалений кореневий каталог буде неявним чином відображено на «локальну адресу» `exec://щось/`, яку потім буде оброблено простим протоколом КІО. Для останнього, по суті, буде використано (наприклад) `podman exec -i якийсь_контейнер cat якийсь_файл` для копіювання з віддаленого на локальний комп'ютер (та інші подібні варіації з використанням інструментів з набору `coreutils`). Достатньо сказати, що його не призначено для загального використання, і він не претендує на жодну продуктивність, але його достатньо, щоб швидко та легко завантажити файл, на який посилаються, до редактора.

Тепер, як легко бачити, можна скористатися спрощеною версією наведених вище налаштувань (без `hostname` або `pathMappings`) для обробки `virtualenv` без дублювання командного рядка сервера (у скрипті-обгортці).

Наступне побачити не просто, тому ми нагадуємо про це явним чином.

- Як `hostname`, так і власне `prefix` визначають середовище виконання (перше — за назвою, друге — за вмістом). У екземплярі процесу редактора одне й те саме `hostname` не повинно бути пов'язане з різними `prefix`, оскільки це призводить до невизначеної поведінки (без потреби у діагностиці).
- Обидва значення, `prefix` і `pathMappings`, підлягають розгортанню змінних (у редакторі) (але з урахуванням попереднього запису).
- Під час виконання також встановлюються деякі змінні середовища, які можна використовувати для (тонкого) налаштування поведінки «засобу запуску з префіксом» (хоча знову ж таки, зверніть увагу на перший пункт). Зокрема, `KATE_EXEC_PLUGIN` встановлюється у значення `lspclient`, а `KATE_EXEC_SERVER` встановлюється у значення ідентифікатора сервера (наприклад, `python`).
- Зокрема, для параметра `KATE_EXEC_INSPECT` також буде встановлено значення 1. Це повідомляє «засобу запуску з префіксом», що отримувач даних (додаток Kate) приймає деякі дані поза діапазоном/протоколом. Це надає змогу засобу запуску використовувати деякі засоби для визначення прив'язки шляхів (наприклад, `podman inspect`) та передавати їх у відповідному форматі. Потім це буде вилучено з потоку, який в іншому випадку відповідає протоколу LSP, та використано для розширення будь-якої визначеної прив'язки шляхів. Це може слугувати альтернативою явному визначенню, наприклад, прив'язки монтувань (знову ж таки, дублювання визначення контейнера). Конкретніше, замість цього можна було б використовувати наведений вище фрагмент коду;

```
{
  // ...
  "exec": {
    "hostname": "foobar"
    // небажаний повтор назви, але допоміжний скрипт зрозумілий і ←
    простий
  }
}
```

```

        "prefix": "exec_inspect.sh foobarcontainer podman exec -i ↵
            foobarcontainer",
        "mapRemoteRoot": true,
        "pathMappings": []
    }
    // ...
}

```

І останнє, але не менш важливе: що робити, якщо підхід «резервного КІО» не є адекватним? На практиці часто можна «монтувати» віддалений кореневий каталог до локальної файлової системи, а потім вказати останній у `pathMappings`. Для початку, добре відома файлова система `sshfs` (fuse) може монтувати (справді) віддалену систему до локальної. Для контейнера (podman/docker) у більшості налаштувань (драйвера файлової системи) передбачено підтримку наступного трюку;

```

$ rootdir=/proc/`podman inspect --format '{{.State.Pid}}` назва_контейне ↵
  pa`/root
# з деякими символічними посиланнями можуть бути проблеми, нижче наведен ↵
  о альтернативу
$ sudo mount --bind $rootdir /somewhere/containername

```

Якщо цей підхід виявиться невдалим, можна звернутися до `sshfs` або його підмножини для монтування кореневого каталогу віддаленої файлової системи або контейнера;

```

# див. відповідні сторінки підручника; для справжніх (незашифрованих) пр ↵
  отоколу обробки файлів достатньо sftp-server
$ socat 'exec:podman exec -i containername /usr/lib/openssh/sftp-server'
  'exec:sshfs -o transform_symlinks -o passive \:/ /десь/назва_контейне ↵
  pa'
# ...
$ fusermount -u /десь/назва_контейнера

```

Зауважте, що таке монтування кореневого каталогу віддаленої системи до файлової системи основної системи, ймовірно, слід вказувати вручну і явно у розділі `pathMappings` (за винятком дуже інтелектуальних засобів запуску з префіксом, який впорядковує усе це автоматично).

4.16 Пошук з заміною

4.16.1 Вступ

За допомогою додатка «Пошук з заміною» Kate ви зможете шукати фрагмент тексту або текст за [формальним виразом](#) у декількох файлах одночасно. Пошук може виконуватися у всіх відкритих файлах або всіх файлах у одному каталозі і, якщо потрібно, у всіх його підкаталогах чи у активному файлі. Шукати можна з фільтруванням за назвою файлів, наприклад шукати лише файли з певним суфіксом назви.

4.16.2 Інтерфейс

4.16.2.1 Запит з пошуку

У верхній частині панелі інструмента пошуку у файлах завжди буде показано такі пункти:



Ви можете відкрити довільну кількість панелей пошуку одночасно. Просто натисніть кнопку створення вкладки у верхній лівій частині панелі інструмента пошуку, і програма відкриє нову вкладку, за допомогою якої можна буде виконати пошук за новими критеріями.



За допомогою кнопки, розташованої у правому верхньому куті панелі пошуку у файлах, ви можете перемикати нижню половину панелі між режимом показу додаткових параметрів пошуку у теці та режимом показу результатів пошуку.

Знайти

За допомогою цього поля можна вказати критерій пошуку. Ви можете ввести звичайний фрагмент тексту або формальний вираз, якщо увімкнено пошук за формальними виразами.

Замінити (поле для введення тексту)

Текст рядка заміни буде додано замість тексту у полі **Знайти**.

Шукати

Після того, як параметри пошуку буде налаштовано, достатньо натиснути кнопку **Шукати**, щоб пошук було розпочато. Щоб розпочати пошук у новій вкладці, натисніть і утримуйте клавішу **Ctrl** перед натисканням кнопки **Шукати**. Ви також можете просто натиснути клавішу **Enter** після введення тексту до поля **Знайти** (так само, натискання комбінації клавіш **Ctrl+Enter** призведе до виконання пошуку у новій вкладці).

Замінити

Після того, як параметри пошуку буде налаштовано, достатньо натиснути кнопку **Замінити**, щоб програма замінила текст, вказаний у полі **Знайти**, текстом, вказаним у полі **Замінити**. Ви також можете просто натиснути клавішу **Enter** після введення тексту-замінника у полі **Замінити**.

Далі

Перейти до наступного відповідника вашого критерію пошуку, якщо потрібно, у іншому файлі.

Замінити у позначених

Те саме, що і **Замінити**, але заміну буде виконано лише у файлах, які було позначено на розташованій нижче панелі.

4.16.2.2 Параметри пошуку у теці

Ці пункти буде показано під раніше згаданими пунктами параметрів пошуку. Якщо замість них буде показано результати пошуку, натисніть кнопку  , щоб програма їх показала.

Шукати у

Тут ви можете вибрати один з трьох варіантів. Позначте пункт **Відкриті файли**, щоб пошук було виконано у файлах, які відкрито у Kate. За допомогою позначення пункту **у теці** можна наказати програмі виконати пошук у теці і, якщо потрібно, її підтеках. Пунктом **Поточний файл** можна скористатися для пошуку у поточному активному файлі.

Якщо завантажено додаток **Проекти**, ви зможете вибрати область пошуку: **Поточний проект** або **Усі відкриті проекти**.

Враховувати регістр

Виконувати пошук лише точних відповідників комбінації великих та малих літер, вказаної у полі запиту пошуку.

Формальні вирази

Надає вам змогу використовувати **формальні вирази** замість звичайного тексту у записах.

Розгорнути результати

Показати усі пункти знайдених фрагментів у кожному з файлів, а не просто список файлів, у яких міститься шуканий фрагмент.

Тека

Ви можете вказати адресу теки, у якій слід виконати пошук. Наприклад, ви можете ввести `~/development/kde/kate/`, якщо пошук слід виконати у початковому коді Kate. Цим пунктом можна буде скористатися лише у режимі **у теці**.

Діалогове вікно відкриття файлів

Натисніть цю кнопку, щоб вказати теку за допомогою діалогового вікна перегляду ієрархії тек. Цим пунктом можна буде скористатися лише у режимі **теки**.



Натисніть цю кнопку, щоб змінити значення у полі **Тека** на батьківську теку поточної позначеної теки. Цим пунктом можна буде скористатися лише у режимі **теки**.



За допомогою цієї кнопки можна змінити значення у полі **Тека** на адресу теки, у якій зберігається поточний відкритий документ. Цим пунктом можна буде скористатися лише у режимі **теки**.

Фільтр

За допомогою цього поля можна наказати програмі виконувати пошук лише у тих файлах, назви яких відповідають певному шаблону. Наприклад, якщо потрібно виконати пошук лише у файлах початкових кодів мовою C++, вкажіть `*.cpp`. Щоб пошук було виконано лише у файлах, назви яких починаються на `kate`, вкажіть `kate*`. Можна вказувати декілька фільтрів, відокремлених комами (,). Щоб виключити деякі результати з пошуку додайте символ мінус (-) перед записом у списку. Цим пунктом не можна буде скористатися у режимі **відкритих файлів**.

Виключити

Варіант, який є протилежним до **Фільтр**. Забороняє Kate шукати у файлах, назви яких відповідають певному зразку. Так само, як і у полі **Фільтр**, ви можете вказати декілька зразків, відокремлених комою (,). Цим пунктом можна буде скористатися, якщо пошук виконується у режимі **у відкритих файлах**.

Рекурсивний

Якщо буде позначено цей пункт, Kate виконуватиме пошук у всіх підтеках вказаної вами теки. Цим пунктом можна буде скористатися лише у режимі **теки**.

Шукати у прихованих

Якщо буде позначено цей пункт, Kate виконуватиме пошук у файлах та теках, які позначено як приховані у вашій операційній системі. Цим пунктом можна буде скористатися лише у режимі **теки**.

Переходити за символічними посиланнями

Типово, інструмент пошуку у файлах не здійснює переходів за **символічними посиланнями**. Якщо буде позначено цей пункт, під час пошуку здійснюватиметься перехід за символічними посиланнями та пошук у файлах, на які вони посилаються. Цим пунктом можна буде скористатися лише у режимі **теки**.

ЗАСТЕРЕЖЕННЯ

Символічні посилання можуть бути посиланнями на теку, яка є батьківською щодо тієї, у якій виконується пошук, або на теку, у якій міститься символічне посилання на її батьківську теку. Якщо у теці для пошуку є таке посилання, Kate виконуватиме перехід за кожним з посилань і пошук у відповідній теці, отже пошук не буде завершено у автоматичному режимі.

Включати двійкові файли

Якщо буде позначено цей пункт, Kate не виконуватиме пошук у файлах, які не є текстовими файлами.

4.16.2.3 Результати пошуку

Результати пошуку буде показано під параметрами пошуку. Якщо замість них показано параметри пошуку у теці, просто натисніть кнопку , щоб програма їх показала. Їх також буде автоматично показано після завершення пошуку.

На панелі результатів пошуку буде показано список файлів, у яких міститься рядок пошуку. Поряд з назвою кожного з файлів буде у дужках вказано кількість знайдених відповідників.

Щоб переглянути список відповідників у файлі, просто натисніть кнопку розгортання (стрілку), розташовану ліворуч від пункту з назвою файла. Якщо вами було позначено пункт **Розгорнути результати**, список буде показано вже розгорнутим. У списку буде показано дані щодо номера рядка та частину вмісту рядка, у якій знайдений фрагмент буде позначено напівжирним шрифтом.

Щоб відкрити файл, у якому було виявлено рядок, просто двічі клацніть лівою кнопкою миші на його пункті. Kate відкриє файл, якщо його ще не відкрито. Перейти до відповідника можна подвійним клацанням лівою кнопкою миші на пункті відповідника у списку знайдених у файлі відповідників.

4.16.3 Структура меню

Зміни → Пошук у файлах (Ctrl+Alt+F)

Відкриває панель засобу «Пошук з заміною».

Зміни → Перейти до наступного відповідника

Перейти до наступного відповідника під час пошуку за допомогою додатку «Пошук з заміною».

Зміни → Перейти до попереднього відповідника

Перейти до попереднього відповідника під час пошуку за допомогою додатку «Пошук з заміною».

Перегляд → Вікна інструментів → Показати панель пошуку із заміною

Вмикає або вимикає показ інструмента **Пошук з заміною** Kate.

4.17 Скрипти і фрагменти тексту Kate

4.17.1 Вступ

За допомогою додатка скриптів і фрагментів Kate ви можете додавати власні елементи автоматизації та керувати ними. Це може варіюватися від статичних багаторазових фрагментів звичайного тексту до складних шаблонів з нетиповими автоматизованими та скриптованими полями, а також повністю нетиповими скриптами JavaScript для зміни вмісту вашого документа.

Здебільшого, ці типи вмісту обробляються однаково. Тому для зручності читання в наступному обговоренні не розрізняються статичні фрагменти коду, фрагменти зі скриптами та скрипти. Натомість термін «фрагмент» використовується для позначення всіх цих елементів тексту.

У додатку також передбачено підтримку призначення клавіатурних скорочень та вставлення фрагментів шляхом автодоповнення коду. Крім того, ви можете отримати з інтернету збірку фрагментів (так звані сховища).

4.17.2 Структура меню

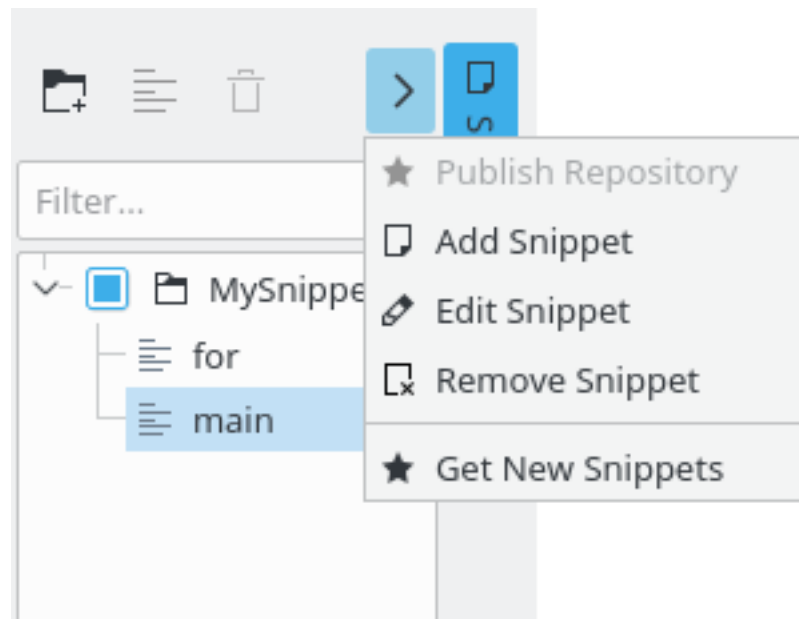
Перегляд → Вікна інструментів → Показати Фрагменти тексту

Показує панель фрагментів з усіма фрагментами у вашому сховищі, які відповідають типу поточного відкритого файлу.

Інструменти → Створити фрагмент

Створити фрагмент, тобто придатний до повторного використання фрагмент тексту, який можна буде потім вставити у довільну частину будь-якого документа, або невеличкий скрипт.

4.17.3 Панель фрагментів



Панель для фрагментів тексту Kate.

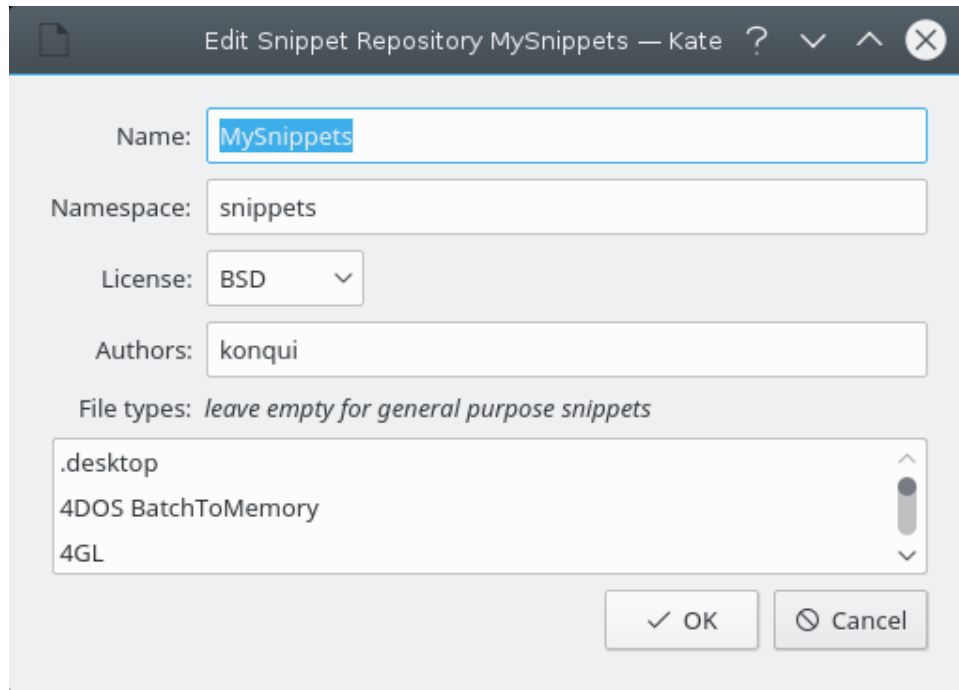
На цій панелі буде показано список сховищ фрагментів разом з пунктами, призначеними для створення власних сховищ, отримання даних з інтернету та завантаження даних з локального файлу. Поряд з пунктом кожного зі сховищ буде показано поле для позначки, за допомогою якого можна буде увімкнути або вимкнути сховище. Також передбачено кнопки для редагування та видалення вже створених сховищ.

4.17.3.1 Завантаження файлів сховища фрагментів

Ви можете отримати сховища фрагментів з інтернету. Просто натисніть кнопку **Отримати нові фрагменти** і програма покаже вікно зі списком сховищ фрагментів. Після отримання потрібного вам сховища не забудьте позначити його пункт, щоб згодом скористатися вмістом сховища.

4.17.3.2 Створення і редагування сховищ фрагментів

Щоб створити нове сховище фрагментів, натисніть кнопку **Додати сховище**. У відповідь має бути відкрито діалогове вікно, у якому програма попросить вас вказати назву файлу фрагментів, умови ліцензування цього файлу та його автора. Після визначення бажаних параметрів натисніть кнопку **Гаразд**, щоб відкрити вікно редактора фрагментів.



Інтерфейс редактора сховища.

У редакторі даних сховища фрагментів можна визначити такі поля:

Name

Буде показано у списку фрагментів панелі інструментів, а також використовуватиметься під час пошуку варіантів для автодоповнення.

Простір назв

Префікс, що використовується під час доповнення коду.

Ліцензія

Виберіть умови ліцензування даних вашого сховища фрагментів.

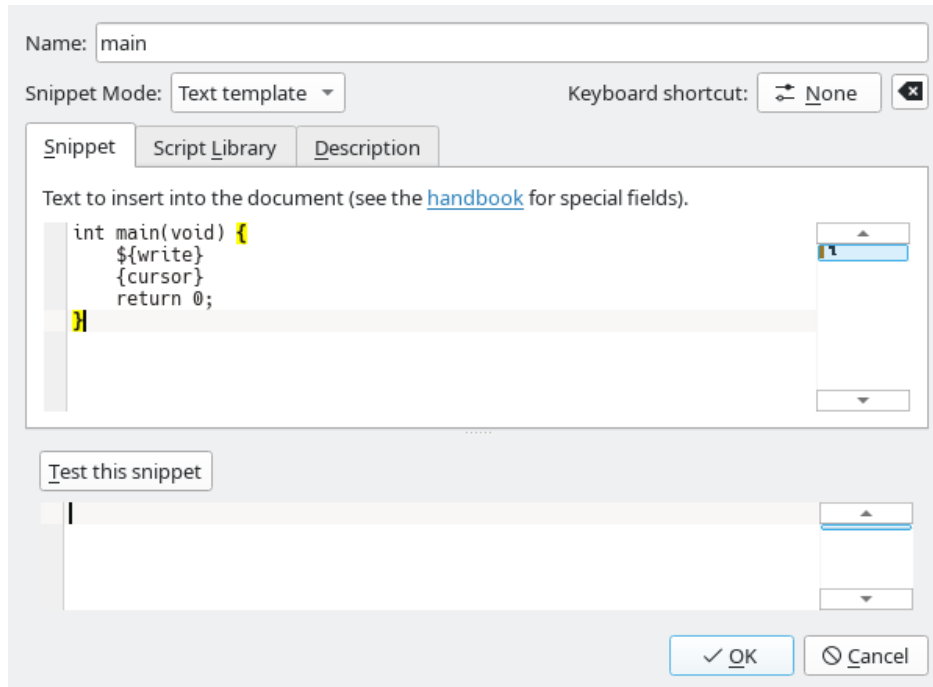
Автори

Вкажіть імена авторів файла фрагментів.

Типи файлів

Виберіть типи файлів, до яких будуть застосовуватися дані зі сховища. Типово встановлено значення «*», тобто сховище застосовуватиметься до всіх файлів. Ви можете змінити тип, наприклад визначити C++, або вибрати значення зі списку, який відкривається натисканням кнопки пунктів. Ви можете вказати декілька типів файлів, якщо натиснете клавішу **Shift** під час їхнього додавання.

4.17.3.3 Створення і редагування фрагментів



Інтерфейс редактора фрагментів.

Назва

Назву буде показано у списку фрагментів і у списку автозавершення (вона є обов'язковою).

Тип

Фрагменти коду можуть бути визначені як текстові шаблони або скрипти. Обидва типи надають змогу використовувати функції JavaScript (докладніше див. нижче), і тому можуть бути використані для досягнення однакових результатів. Однак, як правило, текстові шаблони більше пасують, якщо ви здебільшого хочете *вставляти* текст, тоді як скрипти часто є простішим рішенням, якщо ви хочете *змінити* текст).

ПРИМІТКА

Kate має бути зібрано з KTextEditor версії 6.15.0 або новішої, щоб можна було скористатися цією можливістю. У попередніх версіях можна користуватися лише текстовими шаблонами.

Скорочення

Натискання цієї комбінації клавіш призведе до вставлення (або запуску) фрагмента до поточного документа.

Фрагмент (тип Шаблон тексту)

Текст, який ваш фрагмент вставлятиме до документа.

Фрагмент може містити придатні до редагування поля. Циклічний перехід цими полями можна здійснювати натисканням клавіші **Tab**. У тексті шаблону можна використовувати такі вирази для створення полів:

\${назва_поля} створює просте придатне до редагування поле. Під час редагування усі наступні вставки тієї самої *назви_поля* будуть дзеркальними копіями вмісту першого поля.

ПІДКАЗКА

У полях можуть міститися будь-які символи, окрім символу завершальної фігурної дужки (}).

Щоб вставити `{текст}` без змін до фрагмента, екрануйте символ `$` зворотною похилою рискою: `\${текст}`. Після цього текст не перетвориться у поле. Щоб вставити сам символ зворотної похилої риски перед полем, екрануйте його: `\\${поле}`, `\\\\${поле}` тощо.

`{функція(інше_поле1, інше_поле2, ...)}` — цим можна скористатися для створення поля, яке міститиме результат виконання функції JavaScript під час кожного редагування, і яке буде замінено на повернуте функцією значення. Докладнішу інформацію можна знайти на вкладці **Скрипти**.

`{назва_поля=типове значення}` встановлює типове значення поля. Вмістом фрагмента *типове значення* може бути будь-який вираз JavaScript. Скористайтеся лапками (`{назва_поля="текст"}`) для визначення фіксованого рядка як типового значення.

ПІДКАЗКА

Використовуючи типове значення (`{назва_поля=типове}`), пам'ятайте, що типове значення обчислюється негайно, коли фрагмент коду вставляється в документ, і воно не оновлюється пізніше, коли поля змінюються.

Ви можете посилатися на інші поля у типових значеннях, якщо їх визначено до типового значення, яке буде обчислено. Однак це надасть вам доступ лише до типових значень цих полів.

Якщо не визначено нестандартного типового значення, типовим значенням буде назва поля.

`{cursor}` позначає кінцеву позицію курсора після того, як усі інші поля будуть заповнені. Вставлення тексту у цю позицію завершить редагування. Крім того, після натискання **Esc** для завершення редагування курсор перейде у цю позицію.

Фрагмент (тип Скрипт)

Код JavaScript для обробки цього фрагмента

Якщо цей код містить оператор **return**, повернутий рядок буде вставлено у поточну позицію курсора. Ви також можете скористатися [програмним інтерфейсом для скриптів Kate](#) для безпосереднього внесення змін до документа.

Крім того, ваш код може використовувати функції, визначені на вкладці **Бібліотека скриптів**, які є спільними для всіх фрагментів коду у сховищі.

Бібліотека скриптів

Допоміжні функції JavaScript, якими можна скористатися у ваших фрагментах. Ці функції є спільними для усіх фрагментів у сховищі.

Ви можете скористатися [програмним інтерфейсом роботи зі скриптами Kate](#) для отримання позначеного фрагмента тексту, усього тексту документа, назви файлу та інших даних за допомогою відповідних методів об'єктів `document` та `view`. Щоб дізнатися більше, зверніться до документації із програмного інтерфейсу для роботи зі скриптами.

Якщо функцію викликають у полі фрагмента **Текстового шаблону**, *повернуте* нею значення буде вставлено у текст.

ПРИМІТКА

Використання функцій у фрагмента Текстовий шаблон:

- Функції можуть отримати доступ до актуального вмісту усіх полів через об'єкт `fields`: скористайтеся `fields.my_field` або `fields["my_field"]`. Під час використання функції як типового значення для поля доступні лише поля, визначені у попередньому тексті.
- Поля можна передавати як аргументи функціям. Назву поля можна використовувати безпосередньо, якщо воно є дійсним ідентифікатором JavaScript і не існує змінної з такою назвою. Якщо це не так, використовуйте об'єкт `fields`: `${func(field)}`, `${func(fields.document)}`, `${func(fields["my field"])}`
- Поля функцій (наприклад, `${func()}`) переобчислюється щоразу, коли змінюється вміст будь-якого поля. Однак це не стосується викликів функцій із типовими значеннями: вони обчислюються лише один раз, коли фрагмент коду вставляється в документ.
- Якщо ви користуватиметеся складнішими скриптами, важливо розуміти, що *спочатку* програма вставляє фрагмент коду до документа, а вже *потім* обробляє функції. Наприклад, якщо функція отримує текст у рядку, де було вставлено фрагмент, цей текст також міститиме `${викликФункції()}`. Там, де це призводить до ускладнень, варто скористатися типом фрагмента Скрипт.
- Для спрощення загортання поточного позначеного тексту у теги скористайтеся такою інструкцією: `<strong>${view.selectedText()}</strong>`
- Пам'ятайте, що у поле буде вставлено лише *повернуте значення* функції.

Description

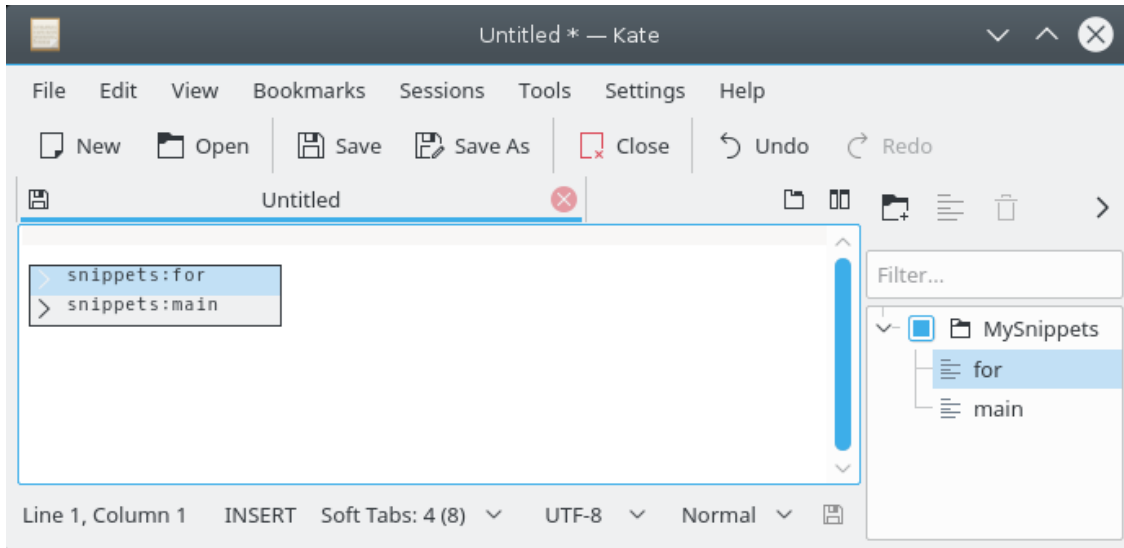
Необов'язковий опис призначення фрагмента. Його буде показано на панелях підказок. Опис може містити базове форматування HTML.

Приклад використання

У наступному прикладі ми викликаємо скрипт, який вставляє позначений текст або, якщо нічого не позначено, типовий текст у теги (тип фрагмента Скрипт):

```
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 Користування фрагментами



Вибір зі списку фрагментів.

Викликати фрагменти можна у декілька способів:

- Натисканням пункту фрагмента на панелі інструментів.
- За допомогою клавіатурного скорочення, якщо ви призначили таке.
- Під час написання тексту чи коду ви можете натиснути комбінацію клавіш **Ctrl+Пробіл** і програма покаже зручне вікно зі списком всіх фрагментів, за допомогою якого ви зможете вибрати потрібний вам фрагмент. Натискання цієї комбінації клавіш надає доступ до функціональних можливостей, подібних до автоматичного доповнення коду.

Якщо у фрагменті містяться змінні (окрім змінної `${cursor}`) курсор буде автоматично пересунуто до першого екземпляра таких змінних, після чого програма перейде у стан очікування на введення тексту. Після введення відповідного значення натисніть клавішу **Tab**, щоб перейти до наступної змінної.

4.17.5 Подяки

Розділ щодо роботи з фрагментами у Kate було написано Джозефом Венігером (Joseph Wenninger).

Особлива подяка учаснику Google Code-In 2011 Martin Gergov за написання більшої частини цього розділу.

4.18 Додаток клавіатурних макросів

4.18.1 Вступ

Запис і відтворення клавіатурних макросів (тобто послідовностей дій з клавіатури).

4.18.2 Основні прийоми користування

4.18.2.1 Розпочати записування клавіатурного макроса:

Інструменти → Клавіатурні макроси → Записати макрос... (Ctrl+Shift+K).

Додаток записуватиме усі натискання клавіш, аж доки ви не завершите записування.

4.18.2.2 Щоб завершити записування, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Завершити записування макроса (Ctrl+Shift+K).

Додаток припинить записування натискань клавіш і збереже послідовність як поточний макрос.

4.18.2.3 Щоб скасувати записування, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Скасувати записування макроса (Ctrl+Alt+Shift+K).

Додаток припинить записування натискань клавіш, але поточний макрос лишиться незмінним.

4.18.2.4 Щоб відтворити поточний макрос, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Відтворити макрос (Ctrl+Alt+K).

Додаток відтворить поточний макрос.

Команда `kmplay` без аргументів також відтворює поточний макрос.

4.18.3 Іменовані макроси

Передбачено можливість збереження клавіатурних макросів із вказаною користувачем назвою.

Іменовані макроси зберігаються між сеансами Kate. Сховищем макросів є файл `keyboardmacros.json` у каталозі даних користувача Kate (зазвичай, цим каталогом є `~/.local/share/kate/`).

4.18.3.1 Щоб зберегти поточний макрос, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Зберегти поточний макрос (Alt+Shift+K).

Додаток попросить вас ввести назву і збереже макрос із відповідною назвою.

Команда `kmsave` `назва` зберігає поточний макрос із використанням назви `назва`.

4.18.3.2 Щоб завантажити збережений макрос як поточний, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Завантажити іменований макрос...

Додаток показуватиме список збережених макросів у підменю. Вибір пункту призведе до завантаження відповідного макроса як поточного.

Команда `kmload` `назва` завантажить макрос, який було збережено під назвою `назва`, як поточний.

4.18.3.3 Щоб відтворити збережений макрос без його завантаження, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Відтворити іменований макрос...

Додаток показуватиме список збережених макросів у підменю. Вибір пункту призведе до відтворення відповідного макроса без його завантаження.

Зауважте, що кожен збережений макрос є дією, яка є частиною поточної збірки дій, отже з макросом можна пов'язати клавіатурне скорочення за допомогою інтерфейсу пункту меню **Параметри → Налаштувати клавіатурні скорочення...**

Команда `kmplay` назва відтворить макрос, який було збережено під назвою `назва`, без його завантаження.

4.18.3.4 Щоб витерти (тобто вилучити) збережений макрос, скористайтесь пунктом меню:

Інструменти → Клавіатурні макроси → Витерти іменований макрос...

Додаток показуватиме список збережених макросів у підменю. Вибір пункту призведе до витирання (тобто вилучення) відповідного макроса.

Команда `kmwipe` назва витре макрос, який було збережено під назвою `назва`.

4.18.3.5 Підказки щодо команд:

Зауважте, що після префікса `km` усі ці команди мають різні літери, тому ви можете ефективно викликати їх за допомогою системи доповнення команд у відповідь на натискання клавіші `Tab`!

4.18.4 Обмеження

У поточній версії клавіатурні макроси не відтворюються належним чином, якщо використано певні типи віджетів графічного інтерфейсу: наприклад, `QMenu`, `QuickOpenLineEdit` або `TabSwitcherTreeView`. Автор не певен щодо причини, але здогадка наступна: ці віджети працюють із подіями клавіатури нестандартно.

4.19 Додаток SQL

4.19.1 Вступ

Структурована мова запитів (SQL) — спеціалізована мова, призначена для оновлення, вилучення та отримання даних з баз даних.

За допомогою додатка SQL Kate ви зможете виконувати такі дії:

- створювати бази даних;
- встановлювати з'єднання з уже створеною базою даних;
- вставляти дані у базу даних і вилучати дані з бази даних;
- надсилати запити до бази даних;
- переглядати результати у форматі таблиці.

4.19.2 Встановлення з'єднання з базою даних

Виберіть пункт **Додати з'єднання** з меню **SQL** або відповідний пункт на панелі інструментів, а потім виберіть драйвер бази даних Qt™, який слід використовувати (ви можете вибрати QSQLITE, QMYSQL3, QMYSQL, QODBC3, QODBC, QPSQL7 або QPSQL). Якщо потрібного вам драйвера немає у списку, вам слід встановити його. Після визначення драйвера натисніть кнопку **Далі**.

Якщо у вибраній вами базі даних для зберігання даних використовується файл, просто вкажіть розташування цього файлу і натисніть кнопку **Далі**. Якщо для роботи з базою даних потрібне з'єднання з сервером, вам слід вказати назву вузла сервера, ім'я вашого користувача та пароль, а також будь-які інші дані, потрібні для роботи відповідного драйвера. Після визначення всіх потрібних даних натисніть кнопку **Далі**.

Нарешті, надайте вашому з'єднанню назву і натисніть кнопку **Завершити**.

4.19.3 Виконання запитів

4.19.3.1 INSERT/DELETE/UPDATE

За допомогою додатка SQL ви можете вставляти, вилучати і оновлювати дані так само, як це можна робити із командного рядка чи програми-скрипту. Просто введіть запит на панелі інструментів і натисніть кнопку **Виконати запит** або скористайтесь пунктом меню **SQL** → **Виконати запит (Ctrl+E)**.

Example 4.1 Декілька прикладів запитів

INSERT

```
INSERT INTO назва_таблиці ("запис1", "запис2", "запис3", "запис4", "запис5")
VALUES ("значення1", "значення2", "значення3", "значення4", "значення5")
```

DELETE

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

UPDATE

```
UPDATE назва_таблиці SET "запис1" = "текст", "запис2" = "текст", "запис3" = "текст", "запис4" = "текст", "запис5" = "текст"
```

4.19.3.2 SELECT

Після виконання запиту **SELECT** ви зможете переглянути результати у таблиці, яку буде показано на панелі **Вивід даних SQL** у нижній частині вікна Kate або у текстовому форматі на панелі **Текстовий вивід SQL**. Якщо під час виконання запиту станеться помилка, ви зможете переглянути відповідне повідомлення на панелі текстового виводу.

Example 4.2 Приклад запиту SELECT

```
SELECT * FROM назва_таблиці
```

На панелі інструментів **Вивід даних SQL** розташовано декілька кнопок:

Ширина стовпчиків за вмістом

Змінити ширину стовпчиків відповідно до вмісту цих стовпчиків.

Висота рядків за вмістом

Змінити висоту рядків відповідно до вмісту цих рядків.

Копіювати

Позначити весь вміст таблиці і скопіювати його до буфера обміну даними.

Експортувати

Експортувати весь вміст таблиці до файла, буфера обміну даними або поточного документа у форматі тексту, розділеного комами (CSV).

Спорожнити

Вилучає всі дані з перегляду таблиці.

Змінити кольори, які буде використано під час показу таблиці, можна за допомогою розділу **SQL** вікна, яке відкривається за допомогою пункту меню **Параметри** → **Налаштувати Kate...**

4.19.4 Навігація

Навігацію вашою базою даних можна здійснювати за допомогою панелі інструментів **Схеми бази даних**, розташованої у вікні ліворуч. Перелік показаних на цій панелі даних залежатиме від використаного вами драйвера бази даних.

Щоб оновити вміст списку, наведіть вказівник миші на вільне місце панелі інструментів і клацніть правою кнопкою миші, потім виберіть у контекстному меню пункт **Оновити**. Щоб створити запит щодо якогось з записів у списку, наведіть вказівник миші на цей запис, клацніть правою кнопкою миші і виберіть у контекстному меню пункт **Створити**, а потім виберіть тип запиту (**SELECT**, **UPDATE**, **INSERT** або **DELETE**) з підменю.

4.19.5 Структура меню

SQL → Додати з'єднання...

Додає нове з'єднання за допомогою одного з драйверів баз даних.

SQL → Вилучити з'єднання

Вилучає позначене з'єднання.

SQL → Змінити параметри з'єднання...

Змінити параметри поточного з'єднання.

З'єднання

Всі створені вами з'єднання з базами даних буде показано у списку між пунктами меню **Змінити параметри з'єднання** та **Виконати запит**. Щоб надіслати запити до відповідної бази даних або внести зміни до цієї бази даних, просто виберіть один з цих пунктів.

SQL → Виконати запит (Ctrl+E)

Виконати ваш запит.

4.19.6 Подяки

Додаток SQL було створено Marco Mentasti.

Особлива подяка учаснику Google Code-In 2011 Ömer Faruk ORUÇ за написання більшої частини цього розділу.

4.20 Додаток перегляду символів

4.20.1 Користування додатком закриття за критерієм

Цей додаток надає змогу розробникам переглядати символи (функції, макроси і структури) під час редагування початкового коду.

Клацанням на пунктах обробленої інформації ви можете без проблем здійснювати навігацію кодом.

У поточній версії передбачено підтримку таких мов:

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

Список можливостей:

- Режим списку/дерева

- Вмикання або вимикання упорядковування

- Приховування або показ параметрів функцій

- Розгортання або згортання ієрархічних списків

- Автоматичне оновлення у відповідь на зміну у документах

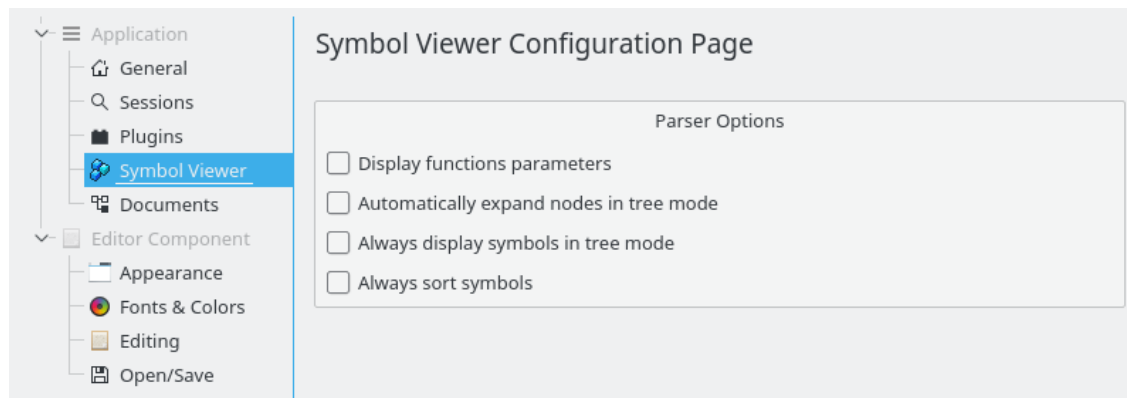
- Обробку коду засновано на засобах підсвічування синтаксичних конструкцій з KDE Frameworks.

4.20.2 Структура меню

Перегляд → Вікна інструментів → Показати список символів (Ctrl+\\)

Увімкнути або вимкнути показ панелі списку символів Kate. На цій панелі буде показано список функцій, макросів і структур у початковому коді активного документа.

4.20.3 Налаштування



Вибір параметрів типового обробника

4.21 Додаток перегляду інструмента термінала

Вбудований емулятор термінала є копією термінала з програми Konsole, його додано для зручності. Доступ до емулятора можна отримати за допомогою пункту меню **Перегляд** → **Вікна інструментів** → **Показати панель термінала**, це вікно завжди отримуватиме фокус після увімкнення його показу. Крім того, якщо буде позначено пункт **Автоматично синхронізувати термінал з поточним документом, за можливості**, поточний каталог у терміналі буде за можливості змінено на каталог поточного документа після показу або внесення змін у поточний документ.

Типовим розташуванням емулятора термінала у вікні Kate, є нижня частина вікна, під областю редагування.

Ви можете налаштувати Konsole за допомогою контекстного меню, яке можна викликати клацанням правою кнопкою миші. Докладніші відомості можна знайти у [підручнику з Konsole](#).

Відповідальним за роботу емулятора термінала є додаток перегляду інструмента термінала.

4.21.1 Структура меню

Перегляд → Вікна інструментів → Показати панель термінала

Вмикає або вимикає показ вбудованого емулятора термінала.

Після першого активування буде створено термінал.

Якщо буде показано панель емулятора термінала, їй буде передано фокус, і ви зможете негайно перейти до набору команд. Якщо буде позначено пункт **Автоматично синхронізувати термінал з поточним документом, за можливості** на сторінці **Термінал головного діалогового вікна налаштування**, у сеансі командної оболонки каталог автоматично змінюватиметься на каталог активного документа, якщо активним документом є локальний файл.

Інструменти → Спрямувати через канал у консоль

Передає поточний виділений текст у рядок введення вбудованого емулятора термінала. Символ нового рядка у кінець рядка додано не буде.

Інструменти → Синхронізувати термінал з поточним документом

Позначення цього пункту призведе до того, що вбудований термінал автоматично змінюватиме свій поточний каталог на каталог активного документа.

Крім того, ви можете налаштувати Kate так, щоб термінал було завжди синхронізовано з поточним документом. Докладніше про це у розділі Розділ 4.21.2.

Інструменти → Запустити поточний документ

Запустити поточний документ у Konsole. Можна запускати лише локальні документи. Kate покаже попередження до запуску документа, оскільки ця дія може бути серйозною загрозою безпеки.

Інструменти → Фокус на панель термінала/Зняти фокус з панелі термінала

Перемикає фокус з поточного документа на термінал, і навпаки.

4.21.2 Налаштування

Налаштувати додаток панелі термінала можна за допомогою сторінки **Термінал діалогового вікна налаштування програми**.

Можна вибрати один з таких варіантів:

Автоматично синхронізувати термінал з поточним документом, за можливості

Якщо буде позначено цей пункт, вбудований термінал змінюватиме поточний каталог на каталог активного документа після запуску і після того, як новий документ отримуватиме фокус. Якщо пункт не буде позначено, вам слід буде виконувати всі дії з навігації у терміналі вручну.

Змінити значення змінної середовища EDITOR на «kate -b»

За допомогою цього пункту можна визначити змінну середовища EDITOR так, щоб програми, що працюють у вбудованому терміналі і автоматично відкривають файл у редакторі, відкривали його у Kate замість типового текстового редактора командної оболонки. Ви не зможете продовжувати користування таким терміналом, аж доки редагований файл не буде закрито у Kate з надсиланням програмі, якій потрібно було редагування, повідомлення про завершення редагування файла.

Приховувати Konsole у відповідь на натискання «Esc»

Уможливило закриття вбудованого термінала натисканням клавіші **Esc**. Може спричинити проблеми із програмами, які працюють у терміналі і використовують клавішу **Esc**, наприклад vim. Такі програми можна вказати за допомогою поля для введення тексту, розташованого нижче. Пункти у списку програм слід відокремлювати комами.

4.22 Додаток для фільтрування тексту

4.22.1 Користування додатком фільтрування тексту

Цим додатком можна скористатися для обробки позначеного тексту за допомогою команд термінала. Вхідними даними команди буде позначений фрагмент тексту. Виведені дані буде або використано як заміник позначеного фрагмента, або скопійовано до буфера обміну, залежно від визначених користувачем налаштувань.

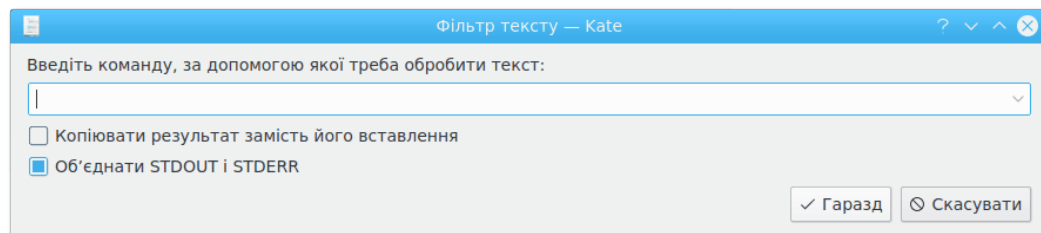
Приклади:

- **less /etc/fstab** — вставити вміст цього файла або скопіювати вміст до буфера обміну даними
- **wc** — порахувати кількість рядків, слів і символів у позначеному фрагменті тексту і вставити ці дані до документа або скопіювати їх до буфера обміну даними
- **sort** — упорядкувати рядки у позначеному фрагменті тексту і вставити результат до документа або скопіювати його до буфера обміну даними

4.22.2 Структура меню

Інструменти → Фільтрувати текст (Ctrl+\)

Відкриває діалогове вікно фільтрування тексту:



Введіть до поля спадного списку команду оболонки або виберіть раніше введену команду зі списку журналу.

Копіювати результат замість його вставлення

Копіювати результат до буфера обміну даними без внесення змін до документа.

Об'єднати STDOUT і STDERR

Якщо позначити цей пункт, виведення до каналів STDOUT та STDERR буде об'єднано і програма не повідомлятиме про помилки. Якщо пункт не буде позначено, програма показуватиме дані, виведені до STDERR, за допомогою панелі пасивного повідомлення.

4.23 Перевірка XML

За допомогою цього додатка можна перевірити належний формат і коректність файлів XML.

За допомогою цього додатка можна перевірити коректність поточного файла. Список попереджень і помилок буде показано у нижній частині головного вікна Kate. Ви можете натиснути повідомлення про помилку, щоб перевести фокус програми на відповідне місце у тексті документа. Якщо у тексті документа є тег «DOCTYPE» для перевірки документа на коректність буде використано саме вказаний цим тегом DTD. Пошук файла DTD буде проводитися відносно розташування файла поточного документа, тобто, якщо тим документа вказано як «DTD/xhtml1-transitional.dtd», а файл документа знаходиться за адресою `/home/peter/test.xml`, програма очікуватиме, що файл DTD знаходиться за адресою `/home/peter/DTD/xhtml1-transitional.dtd`. Додатком підтримується і зовнішні DTD, вказані за допомогою `http`-адреси.

Якщо у файлі не встановлено параметр `doctype`, файл буде перевірено на належне форматування.

Щоб дізнатися більше про перевірку XML, відвідайте [офіційні сторінки W3C](#), присвячені XML.

Для виконання своїх завдань цей додаток викликає зовнішню програму **xmllint**, яка є частиною **libxml2**. Якщо цю програму у вашій системі не було встановлено належним чином, цей додаток не працюватиме.

Щоб завантажити цей додаток, відкрийте діалогове вікно налаштування Kate за допомогою пункту меню **Параметри** → **Налаштувати Kate...** Після цього позначте пункт **Перевірка XML** на сторінці **Програма/Додатки** і закрийте діалогове вікно.

4.23.1 Структура меню

XML → Перевірити XML

За допомогою цього пункту меню можна запустити перевірку, яку було описано раніше.

4.23.2 Подяки

Авторські права на додаток до Kate «Перевірка XML» належать Daniel Naber daniel.naber@t-online.de, ©2002

Авторські права на документацію до додатка належать Daniel Naber, ©2002

4.24 Доповнення XML

Цей додаток надаватиме вам підказки щодо того, які з елементів є дозволеними на певній позиції у файлі XML, відповідно до обмежень DTD цього файла. Програма покаже вам список можливих елементів, атрибутів, значень атрибутів або об'єктів, залежно від позиції курсора (наприклад, якщо символом, що знаходиться ліворуч від курсора, є «&», буде показано список всіх об'єктів). Додаток також може завершувати найближчий незакритий тег, розташований ліворуч.

Для роботи додатка потрібен файл DTD у форматі XML, створений програмою на Perl **dtddparse**. DTD у такому форматі називається «мета-DTD». Деякі з мета-DTD встановлюються разом з програмою. Їх можна знайти у підкаталозі **katexmltools/** каталогу **qtpaths --paths GenericDataLocation**, який також є типовим під час використання функції додатка **Призначити мета-DTD...** Щоб мати змогу створювати ваші власні мета-DTD, вам слід отримати програму **dtddparse** за адресою <http://dtddparse.sourceforge.net>.

4.24.1 Як користуватися

Щоб завантажити цей додаток, відкрийте діалогове вікно налаштування Kate за допомогою пункту меню **Параметри** → **Налаштувати Kate...** Після цього позначте пункт **Перевірка XML** на сторінці **Програма/Додатки** і закрийте діалогове вікно. Після цього скористайтеся пунктом меню **XML** → **Призначити мета-DTD...** Якщо у документі не міститься означення «DOCTYPE» або тип документа є невідомим, вам доведеться вказати файл мета-DTD у файловій системі. Якщо ж усе гаразд, буде автоматично завантажено мета-DTD, який відповідає DOCTYPE поточного документа.

Тепер ви зможете скористатися додатком, після набору таких символів у тексті:

< (клавіша знак «менше»)

Натискання цієї клавіші відкриє список можливих елементів, якщо курсор ще не знаходиться усередині теґу. Зауважте, що у поточній версії ви ще не можете скористатися цією можливістю для вставки елемента найвищого рівня (наприклад «<html>»).

</ (клавіша знак «менше» + клавіша похилої риски)

Введення цих символів призведе до виводу панелі з пропозицією завершення поточного елемента (найближчого елемента ліворуч від курсора). Натисніть **Enter**, щоб прийняти пропозицію. На відміну від пункту меню **Завершити елемент**, комбінація символів працюватиме лише у разі призначення DTD.

" (клавіша лапок)

Після введення лапок буде відкрито список можливих значень атрибута (якщо такий існує), якщо курсор знаходиться всередині теґу.

(клавіша пробілу)

Після введення цього символу додаток відкриє список можливих атрибутів для поточного елемента, якщо курсор знаходиться всередині теґу.

& (клавіша амперсанда)

За допомогою цієї клавіші можна викликати список іменованих об'єктів.

4.24.2 Можливості і обмеження

Ви можете перевірити всі функціональні можливості і обмеження завантаженням файла **katexmltools/testcases.xml** у каталозі **qtpaths --paths GenericDataLocation** до Kate з наступним виконанням настанов з цього файла.

4.24.3 Структура меню

XML → **Вставити елемент...** (Ctrl+Enter)

Відкриє діалогове вікно, за допомогою якого ви зможете вставити елемент XML. Символи < та >, а також завершальний теґ буде вставлено автоматично. Якщо ви під час вибору цього пункту меню було позначено фрагмент тексту, позначений текст буде вставлено між початковим та завершальним теґами. За допомогою цього діалогового вікна також можна буде завершувати запис всіх елементів, які вставляються у поточній позиції курсора, якщо ви призначили загальний DTD за допомогою пункту меню **Призначити мета-DTD...**

XML → Завершити елемент (Ctrl+<)

Якщо ви скористаєтеся цим пунктом меню, програма виконає у вашому документі пошук теґу, який ще не було завершено, і вставить до тексту відповідний завершальний теґ. Пошук буде розпочато з позиції курсора і продовжено у напрямку ліворуч. Якщо програмі не вдасться знайти жодного незакритого теґу, ніяких дій зі зміни тексту виконано не буде.

XML → Призначити мета-DTD...

За допомогою цього пункту меню ви зможете повідомити додатку, який з файлів мета-DTD слід використовувати для поточного документа. Зауважте, що додаток не зберігатиме даних вашого вибору. Вам доведеться повторити процедуру визначення після наступного запуску Kate.

4.24.4 Подяки

Авторські права на додаток до Kate «Доповнення XML» належать Daniel Naber daniel.naber@t-online.de, 2001, 2002.

Версія для KDE SC 4, © Tomáš Trnka, 2010

Авторські права на документацію до додатка належать Daniel Naber, 2001, 2002.

4.25 Додаток навігації результатами збирання

Додаток навігатора компілятора надає вам змогу використовувати навігатор компілятора з самої програми Kate. Навігатор компілятора зазвичай обмежений лише одним файлом, і його важко використовувати для перегляду коду файла у вашому проєкті. Цей додаток призначено для спрощення цього процесу шляхом автоматичного визначення команд компіляції для вашого файла.

4.25.1 Користування

Спочатку вам треба відкрити вкладку навігатора компілятора. Натисніть комбінацію клавіш **Ctrl+Alt+I**, щоб відкрити панель команд і пошукайте пункт «Відкрити поточний файл у навігаторі компілятора», позначте його і натисніть клавішу **Enter**. Ви можете додати відповідну кнопку на панель інструментів або пов'язати з цією дією клавіатурне скорочення, щоб спростити доступ до інструмента.

Потім вам потрібно налаштувати локальний екземпляр навігатора компілятора або скористатися загальнодоступним екземпляром <https://godbolt.org/>. Див. <https://github.com/compiler-explorer/compiler-explorer>. Щойно у вас з'явиться екземпляр, клацніть на спадному списку **Варіант** на вкладці навігатора компілятора, яку ви раніше відкрили, і виберіть **Змінити адресу...**, а потім введіть адресу сервера навігатора компілятора.

Далі вам знадобиться `compile_commands.json`, якщо вам потрібно переглянути `codegen` для файла у вашому проєкті. В іншому випадку ви можете пропустити цю частину. Потрібного результату можна легко досягти, якщо ви використовуєте сучасну систему збирання, як-от CMake або Meson. Для CMake достатньо передати

`-DCMAKE_EXPORT_COMPILE_COMMANDS=ON` під час виклику `cmake`.

`compile_commands.json` слід розмістити в корені вашого проєкту. Закрийте та знову відкрийте вкладку навігатора компілятора у Kate. Після повторного відкриття вкладки має бути показано команди компіляції, необхідні для збирання файла

Якщо ви просто використовуєте навігатор компілятора, щоб побачити `codegen` для чогось тимчасового, ви можете просто записати команди компіляції для вашого файла та натиснути кнопку **Компілювати**. Зауважте, що підтримки автоматичної перекомпіляції під час редагування не передбачено. Вам потрібно натискати «Компілювати» кожного разу, коли ви хочете знову зібрати файл

Після того, як файл буде скомпільовано, ви побачите файл codegen праворуч. Ви можете клацнути правою кнопкою миші та вибрати **Показати пов'язаний код**, щоб побачити codegen для певного рядка. Подібно до правої панелі, ви можете клацнути правою кнопкою миші та вибрати **Гортати до джерела**, щоб переглянути початковий вихідний код для певного рядка codegen

4.26 Додаток форматування

За допомогою додатка форматування можна формувати код. У додатку передбачено збереження журналу скасування дій з документом та розташування курсора користувача під час форматування коду, щоб форматування коду не заважало роботі. Це особливо важливо для автоматичного форматування під час збереження.

4.26.1 Користування

Надає змогу користувачеві формувати код у два способи:

- Вручну за допомогою пункту «Формувати документ»
- Автоматично при збереженні

ПІДКАЗКА

Ви можете пошукати відповідний пункт на панелі команд і пов'язати з ним клавіатурне скорочення для спрощення доступу до інструмента

4.26.2 Підтримувані мови і засоби форматування

Поточний список підтримуваних мов і засобів форматування:

- 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 Налаштовування

Додаток можна налаштувати у два способи:

- На загальному рівні за допомогою вікна налаштувань програми
- На рівні проекту за допомогою файла **.kateproject**

При читанні налаштувань додаток спочатку намагається прочитати дані з файла **.kateproject**, а потім виконує читання загальних налаштувань. Приклад:

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

Код вище:

- вмикає «форматування при збереженні» на загальному рівні
- визначає «jq», як засіб форматування для JSON
- вимикає «форматування при збереженні» для cmake-format і autopep8

Щоб налаштувати форматування проекту, спочатку створіть файл **.kateproject**, а потім додайте до нього об'єкт

“formatting”. В об'єкті

“formatting” ви можете вказати параметри, як це показано у попередньому прикладі. Приклад:

ПРИМІТКА

Для того, щоб додаток працював, вам треба увімкнути [додаток проєктів](#)

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

4.26.4 Тимчасове вимикання форматування при збереженні

При роботі часто виникатиме потреба у тимчасовому вимиканні форматування з певної причини. Зробити це можна за допомогою меню — **Інструменти** → **Форматувати при збереженні**. Також відповідну дію можна виконати за допомогою **Панелі команд**.

Розділ 5

Додаткові інструменти редагування

Щоб дізнатися більше про розширені засоби редагування, які включено до Kate, ознайомтеся із главою щодо додаткових засобів редагування підручника з [KatePart](#).

Розділ 6

Розширення можливостей Kate

T.C. Hollingsworth
Переклад українською: Юрій Чорноіван

6.1 Вступ

Подібно до всіх текстових редакторів з широкими можливостями, у Kate передбачено певні шляхи до розширення функціональних можливостей програми. Ви можете [створювати про-сті скрипти](#), що реалізують додаткові можливості, за допомогою JavaScript або розширювати можливості самого редактора за допомогою додатків до програми Kate, написаних мовою C++. Нарешті, вдосконаливши ваш екземпляр Kate, ви можете [долучитися до розробників](#) і поділитися вашими удосконаленнями з іншими користувачами!

6.2 Як працювати з підсвічуванням синтаксису

Щоб дізнатися більше про додавання визначень підсвічування синтаксичних конструкцій та внесення змін до вже визначених записів підсвічування, ознайомтеся із [розділом щодо роботи з підсвічуванням синтаксичних конструкцій](#) у главі [щодо розробки коду програм у підручнику з KatePart](#).

6.3 Створення скриптів мовою JavaScript

Щоб дізнатися більше про роботу зі скриптами JavaScript, ознайомтеся із [розділом щодо роботи зі скриптами JavaScript](#) у главі [щодо розробки](#) у підручнику з KatePart.

6.4 Додатки до програми Kate (C++)

[Додатки до програми Kate](#) розширюють можливості самого редактора Kate у будь-який спосіб, який можна собі уявити, за допомогою тієї самої мови програмування, якою написано Kate, C++.

Для початку ознайомтеся з [підручником щодо написання додатків на сайті Kate](#).

Розділ 7

Режим введення VI

Щоб дізнатися більше про режим введення VI у Kate, ознайомтеся із [главою щодо режиму введення VI](#) у підручнику з KatePart.

Розділ 8

Пункти меню

8.1 Меню «Файл»

Файл → Створити (Ctrl+N)

За допомогою цього пункту можна створити новий документ у вікні редагування. У списку **Документи**, розташованому ліворуч, новий файл буде мати назву *Без назви*.

Файл → Відкрити... (Ctrl+O)

Відкриває стандартне діалогове вікно KDE **Відкрити файл**. Скористайтесь областю перегляду файлів, щоб обрати потрібний вам файл, а потім натисніть кнопку **Відкрити**, щоб відкрити його.

Файл → Відкрити недавні

Це клавіатурне скорочення призначено для відкриття документів, які ви зберігали нещодавно. Натискання цього пункту відкриє список, розташований збоку від меню, у якому ви побачите назви декількох збережених нещодавно файлів. Натискання позначки одного з цих файлів відкриє відповідний файл у ця програма, якщо файл все ще зберігається за старою адресою.

Файл → Відкрити за допомогою

У цьому підменю ви побачите список програм, які можуть працювати з типом MIME поточного документа. Натискання відповідного пункту призведе до відкриття поточного документа у вказаній програмі.

Крім того, за допомогою пункту **Інша...** можна відкрити діалогове вікно, за допомогою якого ви зможете обрати іншу програму для відкриття активного файла. Ваш файл залишиться при цьому відкритим і у Kate.

Файл → Зберегти (Ctrl+S)

За допомогою цієї дії можна зберегти поточний документ. Якщо цей документ вже було збережено, ця дія призведе до перезапису попереднього збереженого файла без погодження з користувачем. Якщо документ зберігається вперше, буде відкрито діалогове вікно збереження (його описано нижче).

Файл → Зберегти як... (Ctrl+Shift+S)

За допомогою цього пункту можна зберегти файл з новою назвою. Ця дія виконується за посередництвом діалогового вікна, описаного раніше у розділі цього підручника щодо дії **Відкрити**.

Файл → Зберегти з іншим кодуванням

Зберегти документ із новою назвою файла і іншим кодуванням.

Файл → Зберегти копію як

Зберегти копію документа у файлі з новою назвою та продовжити редагування початкового документа.

Файл → Зберегти все (Ctrl+L)

За допомогою цього пункту можна зберегти всі змінені відкрити файли.

Файл → Перезавантажити (F5)

Перезавантажує поточний файл з диска. Цією командою зручно користуватися, якщо інша програма або процес змінили файл, який ви відкрили у ця програма.

Файл → Перезавантажити всі

Перезавантажує всі відкриті файли.

Файл → Друкувати... (Ctrl+P)

Відкриває просте діалогове вікно друку, яке надає користувачеві можливість вказати що, де і як друкувати.

Файл → Експортувати як HTML

Зберегти поточний відкритий документ як файл HTML, який буде форматовано з використанням поточного режиму підсвічування синтаксичних конструкцій та параметрів схеми кольорів.

Файл → Закрити (Ctrl+W)

За допомогою цього пункту можна закрити активний файл. Якщо у файлі є зміни, які не було збережено, програма запитає вас про їх долю до того, як Kate закриє файл.

Файл → Закрити інші

Закрити інші відкриті документи.

Файл → Закрити всі

За допомогою цього пункту можна закрити всі файли, які ви відкрили у Kate.

Файл → Закрити осиротілі

Закрити всі документи у списку файлів, якщо ці документи неможливо повторно відкрити під час запуску, оскільки до них вже немає доступу.

Файл → Вийти (Ctrl+Q)

Використання цього пункту призведе до завершення роботи Kate і закриття всіх файлів, які ви редагували. Якщо у цих файлах були незбережені вами зміни, програма запитає вас про те, чи слід зберегти ці зміни у файлах.

8.2 Меню «Зміни»

Зміни → Вернути (Ctrl+Z)

Скасувати останню команду редагування (введення тексту, копіювання, вирізання тощо)

ПРИМІТКА

За допомогою цього пункту можна скасувати одразу декілька команд редагування, які належать до одного типу, наприклад введення символів.

Зміни → Повторити (Ctrl+Shift+Z)

Цей пункт дозволить вам повторити найостаннішу зміну (якщо така була) виконану за допомогою пункту «Вернути».

Зміни → Вирізати (Ctrl+X)

Ця команда копіює поточний вибраний фрагмент до буфера і вилучає його з тексту. Буфер — елемент системи, який працює як фоновий процес для переносу даних між програмами.

Зміни → Копіювати (Ctrl+C)

Цей пункт меню призводить до копіювання вибраного тексту до буфера, отже, ви зможете вставити його у іншому місці. Буфер — елемент системи, який працює як фоновий процес для переносу даних між програмами.

Зміни → Вставити (Ctrl+V)

Цей пункт меню дасть вам змогу вставити перший пункт буфера обміну даними у позицію курсора. Буфер — елемент, який працює як фоновий процес для переносу даних між програмами.

ПРИМІТКА

Якщо буде увімкнено перезapis виділеного тексту, вставлений текст замінить собою будь-який виділений текст, якщо такий існує у вашому документі.

Зміни → Вставити позначене (Ctrl+Shift+Ins)

За допомогою цього пункту можна вставити до тексту раніше позначений вміст [буфера позначення мишею](#). Позначте який текст за допомогою вказівника миші і вставте його до поточного відкритого файлу, скориставшись цим пунктом.

Зміни → Поміняти місцями із вмістом буфера обміну даними

За допомогою цього пункту можна поміняти місцями позначений фрагмент текст із вмістом [буфера обміну даними](#).

Файл → Журнал буфера даних

У цьому підменю буде показано список з початкових фрагментів блоків тексту, які було скопійовано до буфера обміну даними. Вибір певного пункту у цьому підменю призводитиме до вставлення відповідного блоку у поточний відкритий файл.

Зміни → Копіювати як HTML

Копіювати позначений фрагмент тексту як код HTML, який буде форматовано з використанням поточного режиму підсвічування синтаксичних конструкцій та параметрів схеми кольорів.

Зміни → Вибрати все (Ctrl+A)

Використання цього пункту меню призведе до виділення всього документа. Це буває дуже корисно, якщо ви бажаєте скопіювати весь файл до іншої програми.

Зміни → Скасувати вибір (Ctrl+Shift+A)

Скасовує виділення (якщо таке було) будь-якого тексту у редакторі.

Зміни → Режим прямокутного вибору (Ctrl+Shift+B)

Перемикає режим вибору. У режимі вибору **БЛК**, коли на панелі стану показано рядок **[БЛК]**, ви зможете робити вибір по вертикалі, наприклад, вибирати стовпчики з 5 по 10 у рядках з 9 по 15.

Зміни → Режими введення

Перемикає програм у подібний до *vi* модальний режим редагування. У цьому режимі підтримуються найвживаніші команди і пересування курсора зі звичайного і візуального режимів *vim*, також передбачено можливість вмикання панелі стану режиму *vi*. На цій панелі стану буде показано команди під час їх введення, вивід команд і поточний режим. Поведінку програми у цьому режимі можна налаштувати у розділі [Режим вводу Vi](#) на сторінці **Редагування** діалогового вікна параметрів ця програма.

Зміни → Режим перезапису (Ins)

Перемикає ввід у програмі між режимами Вставки і Заміни. Якщо програма перебуває у режимі **ВСТ**, символ, який ви введете буде записано за позицією курсора. У режимі **ЗАМ** введені символи перезапишуть поточні символи, якщо курсор було розташовано всередині тексту. На панелі стану поточний режим введення буде показано у вигляді записів **ВСТ** або **ЗАМ**.

Зміни → Пошук... (Ctrl+F)

Цей пункт відкриває додаткову панель пошуку, розташовану внизу вікна редактора. Ліворуч на цій панелі ви побачите піктограму для закриття панелі, за якою розташовано невеличке поле для введення шаблону пошуку.

Пошук буде розпочато негайно після того, як ви почнете вводити символи вашого шаблону пошуку. Якщо буде знайдено відповідник у тексті, його буде підсвічено, а колір тла поля запису буде змінено на світло-зелений. Якщо шаблон пошуку не відповідає жодному з рядків тексту, програма продемонструє це зміною кольору тла поля запису на світло-червоний.

Скористайтесь кнопкою  or  для переходу до наступного або попереднього відповідника пошуку у документі.

Відповідники у документі буде підсвічено навіть після закриття панелі пошуку. Щоб зняти підсвічування, натисніть клавішу **Esc**.

Ви можете визначитися з тим, чи слід виконувати пошук з врахуванням регістру символів. Позначення пункту  обмежить варіанти відповідності записами з точною відповідністю регістру для кожного з символів у ключі пошуку.

Натисніть кнопку  на правому краю додаткової панелі пошуку, щоб перемкнути панель у стан потужного пошуку і заміни.

Зміни → Знайти варіанти → Знайти далі (F3)

Повторює останню операцію з пошуку, якщо така виконувалася, без відкриття панелі нагромаджувального пошуку, пошук виконується у напрямку до кінця документа, починаючи з позиції курсора.

Зміни → Знайти варіанти → Знайти позаду (Shift+F3)

Повторює останню операцію з пошуку, якщо така виконувалася, без відкриття панелі нагромаджувального пошуку, пошук виконується у напрямку до початку документа, а не до його кінця.

Зміни → Знайти варіанти → Знайти вибране (Ctrl+H)

Знаходить наступний елемент у виділеному тексті.

Зміни → Знайти варіанти → Знайти вибране позаду (Ctrl+Shift+H)

Знаходить попередній елемент у виділеному тексті.

Зміни → Замінити... (Ctrl+R)

Цей пункт відкриє панель потужного пошуку і заміни. У лівій верхній частині цієї панелі ви побачите піктограму, призначену для закриття панелі, поряд з якою буде невеличке поле для введення шаблону тексту.

Ви можете керувати режимом пошуку вибором одного з режимів **Простий текст**, **Цілі слова**, **Керівні послідовності** і **Формальний вираз** за допомогою спадного списку.

Якщо позначено принаймні один з пунктів **Керівні послідовності** або **Формальні вирази**, програма увімкне пункт меню **Додати...** у нижній частині контекстного меню текстових панелей. Це надасть вам змогу додавати керівні послідовності або формальні вирази у шаблони пошуку або заміни за допомогою списку шаблонів.

Скористайтесь кнопкою  or  для переходу до наступного або попереднього відповідника пошуку у документі.

Текст, на який слід буде замінити ключ пошуку, слід ввести у поле для введення тексту з міткою **Замінити**, після чого слід натиснути кнопку **Замінити**, щоб замінити лише

поточний підсвічений елемент, або кнопку **Замінити всі**, щоб замінити ключ пошуку у всьому документі.

Ви можете змінити поведінку програми під час пошуку з заміною позначенням відповідних пунктів у правій частині панелі. За допомогою кнопки  можна обмежити пошук елементами, у яких реєстр (верхній або нижній) кожного символу збігається з ключем пошуку. За допомогою кнопки  можна обмежити пошук з заміною лише позначеним фрагментом тексту. Якщо ви позначите пункт **Знайти всі**, програма позначить кольором всі відповідники ключа пошуку у документі і покаже кількість знайдених відповідників на невеличкій контекстній панелі.

Натисніть кнопку , розташовану у правій частині панелі потужного пошуку і заміни, щоб перемкнутися на звичайну нагромаджувальну панель пошуку.

Зміни → Перейти → Перейти до відповідної дужки (Ctrl+6)

Пересунути курсор до відповідної парної початкової чи кінцевої дужки.

Зміни → Перейти → Вибрати до відповідної дужки (Ctrl+Shift+6)

Позначити текст між відповідними початковою і кінцевою дужками.

Зміни → Перейти → Перейти до попереднього зміненого рядка

Рядки, які було змінено з часу відкриття файлу, називаються зміненими рядками. За допомогою цього пункту можна перейти до попереднього зміненого рядка.

Зміни → Перейти → Перейти до наступного зміненого рядка

Рядки, які було змінено з часу відкриття файлу, називаються зміненими рядками. За допомогою цього пункту можна перейти до наступного зміненого рядка.

Зміни → Перейти → Перейти до рядка... (Ctrl+G)

За допомогою цього пункту можна відкрити панель переходу вниз вікна. Цією панелью можна скористатися для переведення курсора на певний рядок (який визначається номером) у документі. Номер рядка можна ввести безпосереднім набором на клавіатурі або за допомогою стрілочок вгору і вниз у керуванні лічильника збоку від поля для введення тексту. Маленька стрілочка вгору збільшує значення номера рядка, а стрілочка вниз збільшує його. Закрити панель можна за допомогою натискання піктограми у лівій частині панелі.

8.3 Меню «Перегляд»

Перегляд → Нове вікно

Відкриває ще один екземпляр програми Kate. Новий екземпляр буде ідентичним попередньому екземпляру.

Перегляд → Наступна вкладка (Alt+→)

Активує наступну вкладку на панелі вкладок.

Перегляд → Попередня вкладка (Alt+←)

Активує попередню вкладку на панелі вкладок.

Перегляд → Повторно відкрити останні закриті документи (Ctrl+Shift+T)

Повторно відкриває останній закритий документ або документи.

Перегляд → Швидко відкрити (Ctrl+Alt+O)

Показує поле для пошуку та список відкритих у редакторі файлів. Введення тексту до поля пошуку призведе до фільтрування списку за назвами та адресами документів. Для навігації списком документів можна використовувати клавіші зі стрілкою ↑ і ↓. Після натискання клавіші **Enter** або подвійного клацання лівою кнопкою миші на пункті списку додаток перемкне перегляд на документ, який було позначено у списку. Це спрощує перемикання між документами, якщо у редакторі відкрито дуже багато документів.

Виконати цю дію можна за допомогою натискання кнопки  у верхній правій частині вікна редактора.

Перегляд → Розділений перегляд → Попередній розділ перегляду (Shift+F8)

Перевести фокус на попередній перегляд документа, якщо ви розділили вікно редактора на декілька областей перегляду.

Перегляд → Розділений перегляд → Наступний розділ перегляду (F8)

Перевести фокус на наступний перегляд документа, якщо ви розділили вікно редактора на декілька областей перегляду.

Перегляд → Розділений перегляд → Лівий розділ перегляду

Пересунути фокус на ліву панель розділеного перегляду з використанням, якщо потрібно для усування неоднозначності, позиції курсора.

Перегляд → Розділений перегляд → Правий розділ перегляду

Пересунути фокус на праву панель розділеного перегляду з використанням, якщо потрібно для усування неоднозначності, позиції курсора.

Перегляд → Розділений перегляд → Верхній розділ перегляду

Пересунути фокус на верхню панель розділеного перегляду з використанням, якщо потрібно для усування неоднозначності, позиції курсора.

Перегляд → Розділений перегляд → Нижній розділ перегляду

Пересунути фокус на нижню панель розділеного перегляду з використанням, якщо потрібно для усування неоднозначності, позиції курсора.

Перегляд → Розділити перегляд → Розбити вертикально (Ctrl+Shift+L)

За допомогою цього пункту можна розбити поточний фрейм (яким може бути головна область редагування) на два фрейми однакових розмірів, — новостворений фрейм знаходитиметься ліворуч від поточного фрейма. Фокус буде переведено на новий фрейм, у цьому фреймі також буде показано той самий документ, що і у початковому фреймі до розбиття.

Див. також [Робота з багатовіконним інтерфейсом Kate](#).

Перегляд → Розділити перегляд → Розбити горизонтально (Ctrl+Shift+T)

За допомогою цього пункту можна розбити поточний фрейм (яким може бути головна область редагування) на два фрейми однакових розмірів, — новостворений фрейм знаходитиметься у нижній половині попередньої області фрейма. Фокус буде переведено на новий фрейм, у цьому фреймі також буде показано той самий документ, що і у початковому фреймі до розбиття.

Див. також [Робота з багатовіконним інтерфейсом Kate](#)

Перегляд → Розділений перегляд → Пересунути документ до нового горизонтального поділу

Вертикально розбиває поточний перегляд на два і пересунути поточний активний документ на праву панель.

Див. також [Робота з багатовіконним інтерфейсом Kate](#).

Перегляд → Розділений перегляд → Пересунути документ до нового горизонтального поділу

Розбиває поточний перегляд на два перегляди вертикально і пересунути поточний активний документ на нижню панель.

Див. також [Робота з багатовіконним інтерфейсом Kate](#)

Перегляд → Розділений перегляд → Перемкнути орієнтацію

Перемикання між горизонтальним і вертикальним поділом.

Перегляд → Розділити перегляд → Закрити поточний перегляд (Ctrl+Shift+R)

Закрити активну область редагування (її можна визначити за курсором, що блимає). Якщо у програмі відкрито лише одну область редагування, цей пункт буде вимкнено.

Закриття фрейма не закривати документа, — доступ до нього можна буде отримати за допомогою [меню Перегляд](#), а також списку файлів.

Див. також [Робота з багатовіконним інтерфейсом Kate](#)

Перегляд → Розділений перегляд → Закрити неактивні панелі перегляду

Закрити усі області перегляду, окрім активної (тієї, у якій блимає курсор). Якщо у програмі відкрито лише одну область редагування, цей пункт буде вимкнено.

Закриття фрейма не закривати документа, — доступ до нього можна буде отримати за допомогою [меню Перегляд](#), а також списку файлів.

Перегляд → Розділений перегляд → Приховати неактивні панелі перегляду

Приховує усі панелі розділеного перегляду, окрім поточної активної.

Перегляд → Розділений перегляд → Пересунути розділювач ліворуч

Якщо увімкнено режим розділеного перегляду, за допомогою цього пункту можна по-сунути межу між двома розділеними вертикально документами ліворуч.

Перегляд → Розділений перегляд → Пересунути розділювач праворуч

Якщо увімкнено режим розділеного перегляду, за допомогою цього пункту можна по-сунути межу між двома розділеними вертикально документами праворуч.

Перегляд → Розділений перегляд → Пересунути розділювач вгору

Якщо увімкнено режим розділеного перегляду, за допомогою цього пункту можна по-сунути межу між двома розділеними горизонтально документами вгору.

Перегляд → Розділений перегляд → Пересунути розділювач вниз

Якщо увімкнено режим розділеного перегляду, за допомогою цього пункту можна по-сунути межу між двома розділеними горизонтально документами донизу.

ПРИМІТКА

Доступ до частини типових дій із панелями розділеного перегляду у меню **Перегляд** **Розділений перегляд** можна отримати за допомогою кнопки  , розташованої у правому верхньому куті вікна редактора.

Перегляд → Перегляди інструментів

Перегляд → Вікна інструментів → Показати бічні панелі (Ctrl+Alt+Shift+F)

Вмикає або вимикає показ рядків кнопок бічної панелі. Ця команда не вплине на показ віджетів вмісту бічної панелі, всі бічні панелі, які були видимими, залишаться видимими. Якщо ви призначили клавіатурні скорочення для команд, яким відповідають кнопки, ці скорочення, звичайно ж, продовжуватимуть працювати.

Перегляд → Вікна інструментів → Показати *назва додатка*

Список усіх увімкнених додатків. Скористайтесь полем для позначки перед пунктом додатка, щоб увімкнути чи вимкнути показ відповідного вікна інструмента.

Перегляд → Перемкнутись до командного рядка (F7)

За допомогою цієї команди можна вмикати або вимикати показ **вбудованого інструменту командного рядка**.

Перегляд → Збільшити шрифт (Ctrl++)

Збільшує розмір шрифту.

Перегляд → Зменшити шрифт (Ctrl+-)

Зменшити розмір шрифту.

Перегляд → Схема

У цьому меню ви знайдете список всіх доступних схем кольорів. За допомогою пунктів меню ви зможете змінити схему для поточного перегляду. Щоб змінити типову схему, вам слід скористатися сторінкою **Шрифти та кольори** діалогового вікна налаштування.

Перегляд → Перенесення слів → Динамічне перенесення слів (F10)

Вмикає або вимикає динамічне перенесення рядків у поточному перегляді документа. За допомогою динамічного перенесення рядків можна зробити видимим весь текст документа: вам не потрібно буде використовувати горизонтальну смужку гортання, — рядки документа, за потреби, буде розбито на декілька рядків для перегляду.

Перегляд → Перенесення слів → Помітки динамічного перенесення слів

За допомогою цього пункту ви зможете обрати те, за яких умов і яким чином слід показувати помітки динамічного перенесення слів. Цей пункт доступний, лише якщо позначено пункт **Динамічне перенесення слів**.

Перегляд → Перенесення слів → Показувати помітки статичного перенесення слів

Вмикає або вимикає показ вертикальної лінії, яка позначає позицію максимальної довжини рядка для перенесення. Цю довжину можна налаштувати за допомогою **діалогового вікна налаштування**. Для того, щоб скористатися цією можливістю, для показу документа слід використовувати моноширинний шрифт.

Перегляд → Межі → Показувати рамку для піктограм (F6)

Це пункт-перемикач. Якщо його позначено у лівій частині активного редактора буде показано рамку для піктограм, якщо позначку знято — рамки для піктограм показано не буде. На рамці для піктограм буде показано всі позначені рядки у редакторі.

Перегляд → Межі → Показати номери рядків (F11)

Це пункт-перемикач. Якщо його буде позначено у лівій частині вікна активного редактора буде показано панель з номерами рядків, якщо позначку буде знято — цю панель буде сховано.

Перегляд → Межі → Показувати позначки на смужці гортання

Вмикає або вимикає візуальне представлення закладок (або інших позначок) на вертикальній смужці гортання. Якщо цей пункт буде позначено, позначки буде показано тоненькими рисками кольору позначки на смужці гортання, наведення вказівника миші на таку риску з наступним клацанням середньою кнопкою миші призведе до гортання області перегляду до позиції поряд з позначкою.

Перегляд → Межі → Показувати мінікарту на смужці гортання

За допомогою цього пункту можна замінити смужку гортання візуальним представленням вмісту поточного документа. Докладніший опис мінікарти на смужці гортання можна знайти у **розділі щодо мінікарти підручника з KatePart**.

Перегляд → Згортання коду

Пункти цього меню стосуються **згортання коду**:

Показувати маркери згортання (F9)

Вмикає або вимикає показ панелі маркерів згортання у лівій частині області перегляду.

Згорнути поточний вузол

Згортає область, у якій перебуває курсор.

Розгорнути поточний вузол

Розгортає область, у якій перебуває курсор.

Згорнути вузли найвищого рівня (Ctrl+Shift+-)

Згортає усі області верхнього рівня у документі. Натисніть стрілочку-трикутник, спрямовану праворуч, щоб розгорнути усі області верхнього рівня.

Розгорнути вузли найвищого рівня (Ctrl+Shift++)

Розгортає всі області верхнього рівня у документі.

Показувати недруковані пробіли

Показати або приховати рамку навколо непридатних до друку пробілів.

8.4 Меню «Закладки»

Під описаними тут пунктами буде по одному пункту для кожної з закладок у поточному документі. Текст цих пунктів визначатиметься декількома першими словам позначеного рядка. Вибір одного з таких пунктів призведе до переведення курсора на початок відповідного рядка. Область перегляду редактора буде прогорнуто так, щоб зробити належний рядок видимим.

Закладки → Встановити закладку (Ctrl+B)

Встановлює або вилучає закладку з поточного рядка активного документа (якщо там уже була закладка, її буде вилучено, якщо ж закладки не було, буде встановлено).

Закладки → Очистити всі закладки

За допомогою цього пункту можна вилучити з документа всі позначки, а також список позначок, який буде показано внизу цього меню.

Закладки → Попередня (Alt+PgUp)

За допомогою цього пункту можна перевести курсор на початок першого ж рядка вище за текстом, на якому встановлено закладку. У пункті меню буде показано номер рядка і перші символи тексту у рядку з закладкою. Цей пункт меню стане доступним, лише якщо вище за текстом від рядка з курсором існує закладка.

Закладки → Наступна (Alt+PgDown)

За допомогою цього пункту можна перевести курсор на початок першого ж рядка нижче за текстом, на якому встановлено закладку. У пункті меню буде показано номер рядка і перші символи тексту у рядку з закладкою. Цей пункт меню стане доступним, лише якщо нижче за текстом від рядка з курсором існує закладка.

8.5 Меню «Сеанси»

У цьому меню містяться пункти, призначені для користування і керування сеансами Kate. Докладніші відомості про сеанси можна отримати з розділу [Користування сеансами](#).

Сеанси → Створити

Створює новий порожній сеанс. Всі поточні відкриті файли буде закрито.

Сеанси → Відкрити сеанс...

Відкрити існуючий сеанс. Обрати сеанс ви зможете за допомогою діалогового вікна вибору сеансу.

Сеанси → Швидке відкриття сеансу

За допомогою цього меню ви зможете відкрити існуючий сеанс.

Сеанси → Зберегти сеанс

Зберегти поточний сеанс. Якщо сеанс є анонімним, програма запитає вас про назву сеансу.

Сеанси → Зберегти сеанс як...

Зберегти поточний сеанс з новою назвою. Програма запитає вас про назву, яку слід для цього використати.

Сеанси → Керування сеансами...

Відкриває діалогове вікно інструменту керування сеансами, за допомогою якого ви зможете перейменувати і вилучати сеанси.

8.6 Меню «Інструменти»

Інструменти → Режим тільки для читання

Переводить поточний документ у режим «лише для читання». Цей режим призначено для запобігання будь-якому додаванню тексту та будь-якій зміні форматування документа.

Інструменти → Режим

Тут ви зможете обрати схему типу файлу, якою ви бажаєте скористатися у активному документі. Вказаний вами режим перевизначить для поточного документа загальний режим [типу файлів](#), встановлений за допомогою вікна, що відкривається пунктом меню **Параметри → Налаштувати Kate...**, на вкладці «Режими і типи файлів».

Інструменти → Підсвічування

За допомогою цього пункту ви можете обрати схему підсвічування, яка потрібна для вашого активного документа. Цей параметр перевизначить глобальний режим підсвічування, встановлений у вікні, яке відкривається за допомогою пункту меню **Параметри → Налаштувати редактор...**, але лише для поточного документа.

Інструменти → Відступ

За допомогою цього пункту ви можете вибрати [стиль відступу](#), яку ви бажаєте використати для вашого активного документа. Цей параметр перевизначить глобальний режим відступів, встановлений у вікні, яке відкривається за допомогою пункту меню **Параметри → Налаштувати редактор...**, але лише для поточного документа.

Інструменти → Кодування

За допомогою цього пункту ви можете перевизначити типове кодування, встановлене у вікні, яке відкривається за допомогою пункту меню **Параметри → Налаштувати редактор...**, на сторінці **Відкрити/зберегти**, тобто встановити інше кодування для вашого поточного документа. Кодування, яке ви встановите за допомогою цього пункту, буде діяти лише у межах поточного документа.

Інструменти → Кінець рядка

За допомогою цього пункту ви можете обрати режим завершення рядків, який ви бажаєте використати для вашого активного документа. Цей параметр перевизначить глобальний режим завершення рядків, встановлений у вікні, яке відкривається за допомогою пункту меню **Параметри → Налаштувати редактор...**, але лише для поточного документа.

Інструменти → Додати позначку порядку байтів (BOM)

Після позначення цього пункту ви зможете явним чином додати позначку порядку байтів до документів у кодуванні Unicode. Позначка порядку байтів (BOM) — це символ Unicode, що використовується для визначення порядку байтів текстового файла або потоку даних. Докладніше про неї можна дізнатися зі статті [Позначка порядку байтів](#).

Інструменти → Скрипти

У цьому підменю ви знайдете список всіх дій, пов'язаних з виконанням скриптів. Внести зміни до списку доволі просто: достатньо [створити власні скрипти](#). За допомогою цих скриптів користувач може удосконалити Kate власними інструментами.

[У документації з KatePart можна знайти повний список скриптів.](#)

Інструменти → Увімкнути завершення коду (Ctrl+Пробіл)

Вручну викликати завершення команд за допомогою скорочення, яке прив'язане до цієї дії.

Інструменти → Завершення слів

За допомогою пунктів **Повторно використати слово нижче (Ctrl+9)** і **Повторно використати слово нижче (Ctrl+8)** можна наказати програмі завершувати поточні слова на основі подібних слів, розташованих у напрямку кінця чи початку документа відносно поточної позиції курсора. **Завершення оболонки** відкриває панель завершення з відповідними пунктами.

Інструменти → Перевірка правопису → Автоматична перевірка правопису (Ctrl+Shift+O)

Якщо буде позначено пункт **Автоматична перевірка правопису**, помилкові слова у тексті буде підкреслено на льоту.

Інструменти → Перевірка правопису → Перевірка правопису...

Відкриває вікно програми перевірки правопису. Цю програму спеціально розроблено для того, щоб допомогти вам у пошуку і виправленні всіх помилок правопису.

Щоб дізнатися більше про те, як користуватися програмою перевірки правопису KDE, ознайомтеся з [розділом Перевірка правопису підручника з основ роботи у KDE](#).

Інструменти → Перевірка правопису → Перевірка правопису (від курсора)...

За допомогою цього пункту можна розпочати перевірку правопису, але не з початку документа, а з місця, де зараз знаходиться курсор.

Інструменти → Перевірка правопису → Перевірка правопису вибраного...

Перевіряє правопис у вибраному.

Інструменти → Перевірка правопису → Змінити словник

Відкриває у нижній частині вікна редактора спадний список з усіма можливими словниками для перевірки правопису. За допомогою цього списку вам буде простіше перемикатися між словниками перевірки правопису, наприклад, у документах, написаних різними мовами.

8.7 Меню «Параметри» і «Довідка»

У Kate передбачено типові для KDE пункти меню **Параметри** і **Довідка**. Щоб дізнатися більше, ознайомтеся з розділами щодо [меню «Параметри»](#) та [меню «Довідка»](#) підручника з основ роботи у KDE, окрім цих додаткових пунктів.

Параметри → Тема кольорів

Скористатися іншою схемою кольорів із набору загальносистемних схем кольорів, керування яким можна здійснювати за допомогою модуля «Системних параметрів» [Кольори](#).

Параметри → Показати вкладки

Вкладки можна пересувати за допомогою перетягування із одночасним затисканням лівої кнопки миші, для них передбачено пункти дій контекстного меню для закриття документів, копіювання шляху до буфера обміну даними та відкриття теки документа у програмі для керування файлами. За допомогою кнопки **Швидко відкрити** можна

без проблем перемикатися між документами. Натисніть кнопку  (клацніть лівою кнопкою миші), щоб відкрити меню дій із пунктами з меню **ПереглядРозділений перегляд**.

Параметри → Показувати шлях на смужці заголовка

Якщо позначено, буде показано повний шлях до активного документа. Якщо ж пункт не позначено, буде показано лише назву файла. Цей пункт корисний, якщо ви одночасно редагуєте декілька файлів із однаковою назвою, які розташовано у різних каталогах.

Розділ 9

Як налаштувати Kate

Anders Lund
Переклад українською: Юрій Чорноіван

9.1 Огляд

У Kate передбачено декілька інструментів, за допомогою яких ви зможете налаштувати програму бажаним чином. Серед найважливіших інструментів:

Діалогове вікно налаштування

Основний інструмент налаштування, за його допомогою ви зможете налаштувати програму Kate, компонент редактора і додатки до програми.

Меню Параметри

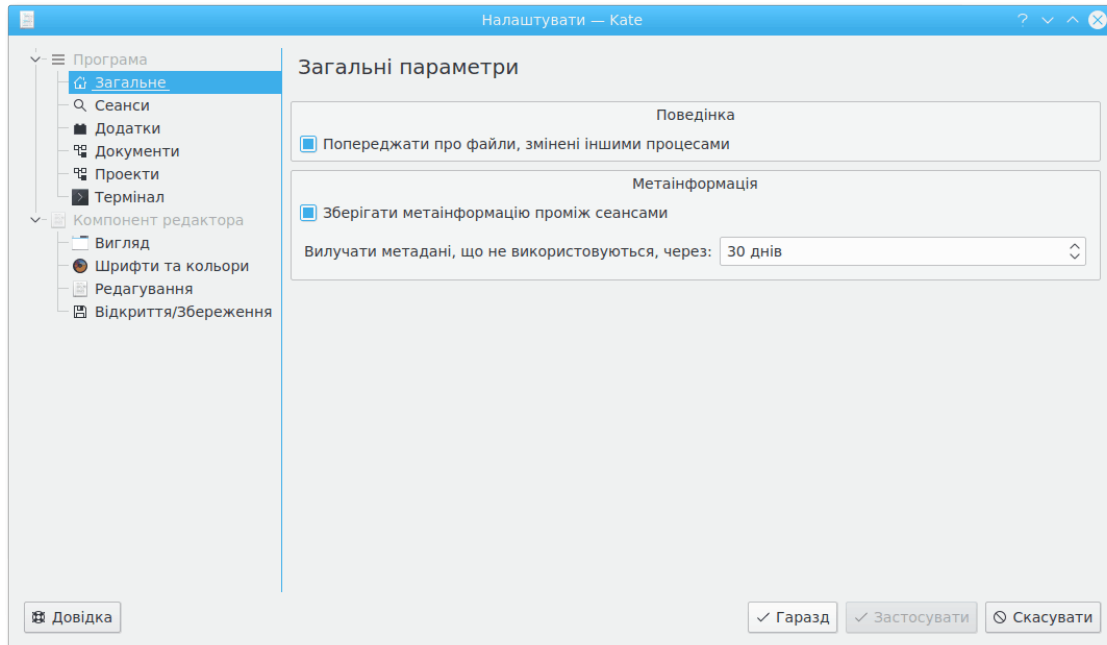
За допомогою цього меню можна змінити найвживаніші параметри, а також відкрити діалогові вікна налаштування.

Меню Перегляд

За допомогою цього меню ви зможете розділити поточний фрейм на частини, а також увімкнути або вимкнути показ панелей піктограм і номерів рядків для поточного документа, який ви редагуєте.

Вбудований термінал використовує параметри, визначені вами у «Системних параметрах», крім того, ви зможете налаштувати цей термінал за допомогою контекстного меню, відкрити яке можна клацанням правою кнопкою миші.

9.2 Основне діалогове вікно налаштування



У вікні налаштування Kate ліворуч розташовано деревоподібний список розділів, праворуч же буде показано сторінку налаштування, що відповідає позначеному розділу.

Розділи налаштування поділено на дві групи, а саме:

- [Налаштування програми](#)
- [Налаштування компонента редактора](#)

9.3 Налаштування програми Kate

У цій групі зібрано сторінки, які стосуються налаштування основної програми Kate.

9.3.1 Загальне

У цьому розділі зібрано декілька загальних параметрів Kate

Поведінка

Відкривати кожен документ у власному вікні

Якщо позначено, кожне документ буде відкрито у власному вікні. Якщо не позначено, кожен документ буде відкрито у новій вкладці поточного вікна.

Перемикання на панель виведення, якщо маємо повідомлення типу

За допомогою цього пункту можна налаштувати умови, за яких Kate має показувати панель виведення залежно від типу результату дії.

Можна вибрати між такими варіантами: **Ніколи**, **Помилка** (при помилці), **Попередження** (при попередженні чи критичніших випадках), **Інформація** (при появі інформації чи критичніших повідомлень) і **Журнал** (при появі повідомлень журналу чи критичніших повідомлень).

Окреме вікно для обробки файлів, які змінено сторонніми програмами

Якщо позначити цей пункт, Kate сповіщатиме вас модальним діалоговим вікном про усі файли, які було змінено у зовнішній програмі, коли головне вікно програми отримуватиме фокус. Ви зможете одночасно виконувати одну і ту саму дію з групою з декількох файлів: перезавантажити, зберегти або відкинути змінені файли.

Якщо пункт не позначено, Kate питатиме вас про подальші дії щодо кожного зміненого файла окремо і лише тоді, коли панель перегляду файла отримуватиме фокусування.

Швидке відкривання

Режим відповідності

Встановлює режим списку для інструмента [Швидке відкривання](#). Відповідність може бути встановлено за назвою або шляхом до файла.

Режим списку

Встановлює режим списку для інструмента [Швидке відкривання](#). Можна вибрати один із таких варіантів: **Файли поточного проєкту** та **Файли усіх проєктів**.

Вкладки

Обмеження кількості вкладок

Встановлює максимальну кількість вкладок. Виберіть варіант **Без обмежень**, якщо ви не хочете обмежувати цю кількість.

Автоприховування вкладок

Якщо позначено, панель вкладок буде автоматично приховано, якщо відкрито лише один документ.

Показувати кнопку закриття

Якщо позначено, на кожній вкладці буде показано кнопку закриття.

Розширювати вкладки

Якщо позначено, вкладки займатимуть максимум доступного місця.

Подвійне клацання відкриває новий документ

Якщо позначено, подвійне клацання відкриває документ.

Клацання середньою закриває документ

Якщо позначено, клацання середньою закриває документ.

Дозволити гортання вкладок

Якщо позначено, уможлиблює гортання на смужці вкладок, якщо кількість вкладок стає великою.

Обривати текст вкладок трьома крапками

Якщо позначено, текст заголовка вкладки може бути обірвано трьома крапками, якщо він є надто довгим.

Вкладки

Натиснуто кнопку назад

Надає змогу вибрати дію у відповідь на натискання кнопки «Назад» миші: **Попередня вкладка** і **Назад у журналі**.

Натиснуто кнопку вперед

Надає змогу вибрати дію у відповідь на натискання кнопки «Вперед» миші: **Наступна вкладка** і **Вперед у журналі**.

9.3.2 Сеанс

У цьому розділі містяться параметри, що стосуються [використання сеансів](#).

Параметри запуску програми

Тут ви можете обрати бажаний спосіб, у який Kate поводитиметься під час запуску. Цей параметр може бути перевизначено дії, які виконуватиме програма у відповідь на [рядок команди](#).

Почати новий сеанс

Якщо обрати цей варіант, Kate після запуску відкриє новий сеанс без назви.

Завантажити останній сеанс

За вибору цього варіанта Kate після запуску використовуватиме останній відкритий сеанс. Це добрий вибір, якщо ви бажаєте завжди використовувати один і той самий сеанс або рідко змінюєте сеанси.

Вручну вибрати сеанс

За вибору цього варіанта Kate покаже невеличке діалогове вікно, за допомогою якого ви зможете обрати бажаний сеанс або завантажити типовий сеанс, якщо жодного сеансу ще не збережено. Це типова поведінка. Чудовий вибір, якщо ви часто використовуєте різні сеанси.

Параметри запуску/завершення програми

Виберіть режим поведінки Kate при завершенні роботи програми. Можна визначити, які дані Kate має **Автоматично зберігати і відновлювати**.

Новостворені незбережені файли

Якщо буде позначено цей пункт, Kate автоматично зберігатиме усі новостворені незбережені файли.

Файли із незбереженими змінами

За допомогою цього пункту можна налаштувати Kate на автоматичне збереження усіх файлів із незбереженими змінами під час завершення роботи програми.

Завершувати роботу Kate, якщо закрито вікно останнього файла

Якщо позначено, Kate завершуватиме роботу, якщо закрито останній редагований файл. Якщо не позначено, програма продовжуватиме роботу із порожньою сторінкою, з якої ви можете розпочати створення нового файла.

Елементи сеансу

Включити конфігурацію вікон

Якщо буде позначено цей пункт, Kate зберігатиме налаштування вікна кожного сеансу.

Зберігати метаінформацію проміж сеансами

Якщо буде позначено цей пункт, Kate зберігатиме метадані, такі як закладки і налаштування сеансу, навіть якщо ви закриєте ваші документи. Дані буде використано, якщо документ до часу, коли його буде знову відкрито, не буде змінено.

Вилучати метадані, що не використовуються, через

Тут ви можете встановити максимальну кількість днів, протягом яких зберігатимуться метадані для файлів, які не відкриватимуться у програмі. Цей пункт допоможе вам зберігати невеликий розмір бази метаданих.

Зміни, внесені до даних сеансу (відкриті файли і, якщо увімкнено, налаштування вікон) зберігатимуться завжди.

9.3.3 Додатки

На цій сторінці ви знайдете список встановлених додатків до програми Kate. На ній буде показано назву кожного з додатків і його короткий опис. Щоб увімкнути додаток, вам слід позначити відповідний пункт у списку.

Якщо додаток надає можливості налаштування, доступ до розділу з налаштуваннями буде відкрито на дочірній панелі цієї сторінки.

Докладніший опис доступних додатків можна знайти у розділі розділ 4.

9.3.4 Налаштування компонента редактора

Щоб дізнатися більше про цей розділ діалогового вікна налаштування, ознайомтеся із розділом щодо налаштування компонента редактора з підручника щодо [KatePart](#).

9.3.5 Налаштування змінних документа

Щоб дізнатися більше про використання змінних документа у Kate, ознайомтеся із розділом щодо налаштування змінних документа у підручнику з [KatePart](#).

Розділ 10

Подяки і ліцензія

Kate. Авторські права на програму належать команді розробників Kate, ©2000–2005
КОМАНДА РОЗРОБНИКІВ КАТЕ:

Christoph Cullmann cullmann@kde.org

Керівник проекту і розробник ядра

Anders Lund anders@alweb.dk

Основний розробник, підсвічування синтаксису Perl, документація

Joseph Wenninger kde@jowenn.at

Основний розробник, підсвічування синтаксису

Michael Bartl michael.bartl@chello.at

Головний розробник

Phlip phlip_cpp@my-deja.com

Компілятор проекту

Waldo Bastian bastian@kde.org

Класна система буферів

Matt Newell newellm@proaxis.com

Тестування...

Michael McCallum gholam@extra.co.nz

Головний розробник

Jochen Wilhemly digisnap@cs.tu-berlin.de

Автор KWrite

Michael Koch koch@kde.org

портування KWrite на KParts

Christian Gebauer gebauer@bigfoot.com

Не вказано

Simon Hausmann hausmann@kde.org

Не вказано

Glen Parker glenebob@nwlinc.com

Журнал дій у KWrite, інтеграція з KSpell

Підручник з Kate

Scott Manson sdmanson@alltel.net

Підтримка підсвічування синтаксису XML у KWrite

John Firebaugh jfirebaugh@kde.org

Різноманітні латки

Dominik Haumann dhdev@gmx.de

Розробник, майстер підсвічування програми

СВІЙ ВНЕСОК ЗРОБИЛИ І БАГАТО ІНШИХ ЛЮДЕЙ:

Matteo Merli merlim@libero.it

Підсвічування синтаксису для файлів Spec RPM, Diff та інші речі

Rocky Scaletta rocky@purdue.edu

Підсвічування синтаксису для VHDL

Yury Lebedev

Підсвічування синтаксису для SQL

Chris Ross

Підсвічування синтаксису для Ferite

Nick Roux

Підсвічування синтаксису для ILERPG

John Firebaugh

Підсвічування синтаксису для Java™ та багато інших речей

Carsten Niehaus

Підсвічування синтаксису для L^AT_EX

Per Wigren

Підсвічування синтаксису для Makefiles та Python

Jan Fritz

Підсвічування синтаксису для Python

Daniel Naber

Невеличкі виправлення вад, додаток XML

Авторські права на документацію до програми належать Seth Rothberg sethmr@bellatlantic.org, ©2000, 2001

Авторські права на документацію до програми належать Anders Lund anders@alweb.dk, ©2002, 2003, 2005

Переклад українською: Юрій Чорноіван yurchor@ukr.net

Цей документ поширюється за умов дотримання [GNU Free Documentation License](#).

Ця програма поширюється за умов дотримання [GNU General Public License](#).

Додаток А

Формальні вирази

Щоб дізнатися більше про використання формальних виразів у Kate, ознайомтеся із [додатком щодо формальних виразів у підручнику з KatePart](#).

Додаток Б

Предметний покажчик

Емулятор терміналу, [77](#)
Головне вікно, [14](#)
Область редагування, [15](#)
Список документів, [38](#)
налаштування
 параметри
 уподобання, [100](#)