

ОСОБЕННОСТИ РЕШЕНИЙ ДЛЯ ОБРАБОТКИ ТЕКСТОВЫХ ДАННЫХ

© 2019 И. Я. Львович, Н. Е. Кравцова, Ю. Л. Чупринская

*Воронежский институт высоких технологий (г. Воронеж, Россия)**В статье рассматриваются характеристики решений, связанных с обработкой текстовых данных. Дается сравнительный анализ разных программ.**Ключевые слова: программное решение, текстовые данные, обработка.*

Рассмотрим некоторые особенности существующих программных решений для обработки текстовых данных [1, 2].

Анализ существующих решений в области подсветки синтаксиса показал, что применение элемента управления RichTextBox не является выгодным решением при работе с большими объемами текста [3, 4].

В Internet опубликован тест производительности редактора, построенного на базе RichTextBox. Тестирование производилось по следующему плану:

1. Реализация 170 зарезервированных слов SQL Server;
2. Копирование и вставка 1122 строк кода из PetShop_Schema.sql.

При наборе текста, код "SELECT * FROM sometable" появляется через 5 секунд. При тестировании использовался ноутбук Pentium 4, 2000Mb ОЗУ.

Editor.NET.

Editor.NET – это продвинутый редактор кода, позволяющий встраивать очень гибкий элемент управления в приложения .NET. Он предназначен для подсветки синтаксиса более чем 30 языков программирования, таких как C, C#, Delphi, VB, Java, Xml, Html и может быть расширен для поддержки практически неограниченного числа языков. Editor.NET написан на C# и построен на классах .NET. Он поставляется с полным исходным кодом, включая код, необходимый во время разработки.

Пакет Editor.NET включает компоненты:

1. Многострочный элемент управления SyntaxEdit для редактирования.
 2. Компонент TextSource для хранения действительных данных.
 3. Анализатор для подсветки синтаксиса.
 4. 30 предопределённых синтаксических схем.
 5. Продвинутые анализаторы для C#, VB.NET, J# и XML.
 6. Дополнительный цветовой редактор.
 7. Стандартные диалоги Поиск, Замена, Перейти и Настройки.
- Особенности редактора:
1. Настраиваемая подсветка синтаксиса.
 2. Автодополнение.
 3. Фрагменты кода.
 4. Перенос слов.
 5. Выделение.
 6. Сворачивание участков кода.
 7. Поля документа.
 8. Несколько представлений одного текста [5, 6].
 9. Неограниченное количество операций отменить/вернуть.
 10. Разметка страницы.
 11. Возможно прокручивать текст в окне.
 12. Разделители содержимого.
 13. Индикаторы изменённых строк.
 14. Закладки.
 15. Операции поиска и замены.
 16. Локализация.
 17. Гипертекст.
 18. Проверка правописания.
 19. Фоновое изображение.
 20. Печать и Предварительный просмотр.
 21. Экспорт в Html, Rtf и XML.
 22. Глобальные настройки разных редакторов в приложении [7, 8].
 23. Другие опции: поддерживается отображение непечатаемых символов, подсветка соответствующих скобок и текущей строки.

1.4 MonoDevelop Editor.

Львович Игорь Яковлевич – Воронежский институт высоких технологий, д. т. н., профессор, office@vivt.ru.

Кравцова Нина Евгеньевна – Воронежский институт высоких технологий, студент, kravtrovnnin@yandex.ru.

Чупринская Юлия Леонидовна – Воронежский институт высоких технологий, студент, chuprimy9ul@yandex.ru.

Редактор кода в MonoDevelop 2.2 включает множество исправлений и новых возможностей:

1. Автосохранение документа.
2. Шаблоны кода. Поддержка шаблонов была полностью переписана и предоставляет теперь намного больше возможностей:

1. Шаблоны теперь могут иметь метки-заполнители.
2. Поддержка 'Surround With'.
3. Выделение блоков текста.
4. Подсветка кода.
5. Значительные улучшения производительности.
6. Форматирование кода.
7. Форматирование «на лету».
8. Соответствие акрониму при автозавершении кода.
9. Отображение XML документации при автозавершении кода.
10. Улучшенная поддержка vi режима.
11. Новая команда 'Dynamic Abbrev' (аналог команды Emacs).

12. Генератор кода. Команда 'Show Code Generation Window' позволяет быстро создавать шаблоны кода на основе текущего контекста. Вот некоторые примеры:

1. Создать конструктор, инициализирующий набор полей.
2. Создать свойства для полей.
3. Переопределить члены базового класса.
4. Реализовать ToString().
5. Реализовать методы проверки равенства.
6. Ввести в метод проверку null-значения параметра.

Редактор Visual Studio был полностью переписан в управляемом коде и теперь имеет новые характеристики:

1. Расширенный режим стыковки

Окна документов больше не ограничены рамками интегрированной среды разработки (IDE). Можно пристыковать окна документов к границам IDE или переместить их в любую точку рабочего стола (включая другой монитор). Если открыты два связанных окна документа, например, дизайнер и редактор Windows Form, то изменения, сделанные в одном окне будут немедленно видны в другом окне.

Окна инструментов теперь могут свободно двигаться между стыковками у границ IDE, перемещаться за пределами IDE, заполнять часть или всё пространство рамки документа. Всё это время они готовы выполнить стыковку.

2. Увеличение

В любом окне редактирования кода можно быстро выполнить увеличение или уменьшение, удерживая клавишу CTRL и вращая колёсико мыши. Можно также увеличить текстовые окна инструментов, например окно Output.

3. Выделение блоков текста

В предыдущих версиях Visual Studio была возможность выбора прямоугольного блока текста, удерживая клавишу Alt в процессе выделения с помощью мыши. Далее можно было скопировать или удалить выбранный текст. VS добавляет новые возможности к этой функции:

1. Ввод текста: Вводите текст в область выделения для вставки нового текста в каждую выбранную строку.

2. Вставка: Вставьте содержимое одного текстового блока в другой.

3. Блоки нулевой длины: Создайте вертикальное выделение нулевой ширины для многострочной вставки текста.

Эти возможности можно использовать для быстрой работы с группами выражений, такими как изменение модификаторов доступа, установка полей или добавление комментариев.

4. Иерархия вызовов

Иерархия вызовов, которая доступна в Visual C# и Visual C++, отображает некоторые части кода таким образом, что имеется возможность перемещаться по нему более эффективно:

1. Вызовы определённого метода, свойства или конструктора.

2. Реализации члена интерфейса.

3. Перегрузки виртуального или абстрактного члена.

Это помогает лучше понять, как идёт поток исполнения кода, оценить эффекты изменений, а также исследовать возможные пути исполнения, путём оценки сложных цепочек вызовов и других точек входа на разных уровнях кода.

Иерархия вызовов доступна во время разработки, в противоположность стеку вызовов, который отображается отладчиком.

Имя метода отображается на панели окна Call Hierarchy. Если развернуть узел метода, появятся подузлы Calls To и Calls From. Если раскрыть узел Calls To, то появятся все методы, которые вызывают выбранный метод. Если раскрыть узел Calls From, отобразятся все методы, которые вызывает данный метод. Можно также раскрыть подузлы узлов Calls To и

Calls From. Это позволяет перемещаться по стеку вызовов.

5. Функция Navigate To

Функцию Navigate To можно использовать для поиска символа или файла в исходном коде.

Navigate To позволяет найти определенное положение в решении или исследовать элементы решения. Это позволяет выбрать хороший набор из подходящих результатов запроса x96 10ь.

Можно найти ключевые слова, используя стиль Camel и знак подчёркивания.

6. Подсветка ссылок

При щелчке на элементе исходного кода, все экземпляры этого элемента подсвечиваются в документе.

Подсвеченные символы могут включать объявления и ссылки, а также другие элементы, которые может вернуть Find All References. Они включают имена классов, объектов, переменных, методов и свойств.

В коде Visual Basic также подсвечиваются ключевые слова для многих управляющих структур.

Для перемещения к следующему или предыдущему подсвеченному элементу, необходимо нажать CTRL+SHIFT+СТРЕЛКА ВНИЗ или CTRL+SHIFT+СТРЕЛКА ВВЕРХ.

7. Функция Generate From Usage

Функция Generate From Usage позволяет использовать классы и методы перед тем, как программист определит их. Можно создать заглушку для любого неопределённого класса, конструктора, метода, свойства, поля или перечисления, которое необходимо использовать, но ещё не было определено. Можно создать новые типы и методы, не покидая текущее положение в коде, что минимизирует вмешательство в рабочий процесс [11, 12].

Generate From Usage обеспечивает программирование в стиле «разработка через тестирование».

8. Режим предположения IntelliSense

IntelliSense теперь предоставляет два альтернативных способа автозавершения: режим завершения и режим предположения [13]. Режим предположения можно использовать в ситуациях, когда классы и методы используются перед их объявлением.

Когда открыто окно IntelliSense, можно нажать CTRL+ALT+ПРОБЕЛ чтобы переключиться между режимом завершения и предположения.

ЛИТЕРАТУРА

1. Гуськова, Л. Б. О построении автоматизированного рабочего места менеджера / Л. Б. Гуськова // Успехи современного естествознания. – 2012. – № 6. – С. 106.

2. Львович, Я. Е. Системно-деятельностный подход к процессу управления функционирования и развития вуза / Я. Е. Львович, И. Я. Львович, В. Г. Власов, В. Н. Кострова // Инновации. – 2003. – № 3. – С. 34-42.

3. Филипова, В. Н. О применении информационных технологий в туристической сфере / В. Н. Филипова // Успехи современного естествознания. – 2012. – № 6. – С. 112-113.

4. Кострова, В. Н. Оптимизация распределения ресурсов в рамках комплекса общеобразовательных учреждений / В. Н. Кострова, Я. Е. Львович, О. Н. Мосолов // Вестник Воронежского государственного технического университета. – 2007. – Т. 3. – № 8. – С. 174-176.

5. Львович, И. Я. Основы информатики / И. Я. Львович, Ю. П. Преображенский, В. В. Ермолова. – Воронеж, 2014. – 339 с.

6. Преображенский, Ю. П. О видах информационных систем в организации / Ю. П. Преображенский // Молодежь и системная модернизация страны: Сборник научных статей 3-й Международной научной конференции студентов и молодых ученых. В 4-х томах. Отв. редактор А.А. Горохов. – 2018. – С. 131-134.

7. Преображенский, Ю. П. Построение складской структуры предприятия / Ю. П. Преображенский // Молодежь и системная модернизация страны: Сборник научных статей 3-й Международной научной конференции студентов и молодых ученых. В 4-х томах. Отв. редактор А.А. Горохов. – 2018. – С. 286-290.

8. Львович, И. Я. Разработка системы учета заявок на ремонт оборудования / И. Я. Львович, А. П. Преображенский, О. Н. Чопоров // Перспективные достижения современных ученых. Техника и технологии. Одесса, 2017. – С. 48-74.

9. Преображенский, Ю. П. Проблемы управления в производственных организациях / Ю. П. Преображенский // Актуальные проблемы развития хозяйствующих субъектов, территорий и систем регионального и муниципального управления: Материалы XIII международной научно-практической конференции; Под редакцией Ю. В. Вертаковой. – 2018. – С. 208-211.

10. Преображенский, А. П. Возможности обеспечения развития предприятий / А. П. Преображенский // В мире научных открытий. – 2015. – № 10 (70). – С. 196-201.

11. Филипова, В. Н. Вопросы прогнозирования в туристическом бизнесе / В. Н. Филипова, К. В. Кайдакова, Д. С. Тарасова, А. П. Преображенский // Моделирование, оптимизация и информационные технологии. – 2013. – № 1 (1). – С. 17.

12. Преображенский, А. П. Применение статистических методов при управлении

предприятием / А. П. Преображенский, О. Н. Чопоров // Наука Красноярья. – 2017. – Т. 6. – № 1-2. – С. 273-278.

13. Choporov, O. N. The features of construction of the automated workplace for the manager that accepting the applications / O. N. Choporov, A. P. Preobrazhensky // Modern informatization problems in economics and safety Proceedings of the XX-th International Open Science Conference (Yelm, WA, USA, January 2015). Editor in Chief Dr. Sci., Prof. O. Ja. Kravets. – 2015. – С. 71-76.

FEATURES PROCESSING SOLUTIONS OF TEXT DATA

© 2019 I. Ya. Lvovich, N. E. Kravtsova, Yu. L. Chuprinskaya

Voronezh Institute of High Technologies (Voronezh, Russia)

The paper discusses the characteristics of solutions related to the processing of text data. The comparative analysis of different programs is given.

Key words: software solution, text data, processing.