

Я.В. Баранов

студент кафедры вычислительных систем и сетей

А.В. Аксенов

научный руководитель

ЭФФЕКТИВНАЯ ОРГАНИЗАЦИЯ И УПРАВЛЕНИЕ ФРАГМЕНТАМИ ИСХОДНОГО КОДА

Предложена программа, повышающая производительность труда программиста за счёт быстрого и удобного доступа к фрагментам исходного кода. В действительности, программист часто использует ранее написанный код в новых проектах. Если таких проектов много и их исходный код занимает тысячи строк – поиск необходимых блоков не эффективен. Более того, проекты часто хранятся в разных директориях, а исходный код – в множестве различных файлов. Поиск вручную по ним весьма затруднителен. Поэтому я разработал программу CodePaster, которая представляет встроенную базу фрагментов исходного кода и поисковик по ним.

Программа написана на языке C++ с использованием библиотеки Qt. Работа происходит большей частью через глобальные горячие клавиши. Есть свой значок в области уведомлений и главное контекстное меню, связанное с ним. При запуске под Microsoft Windows 7 программа выглядит так, как показано на рис. 1.

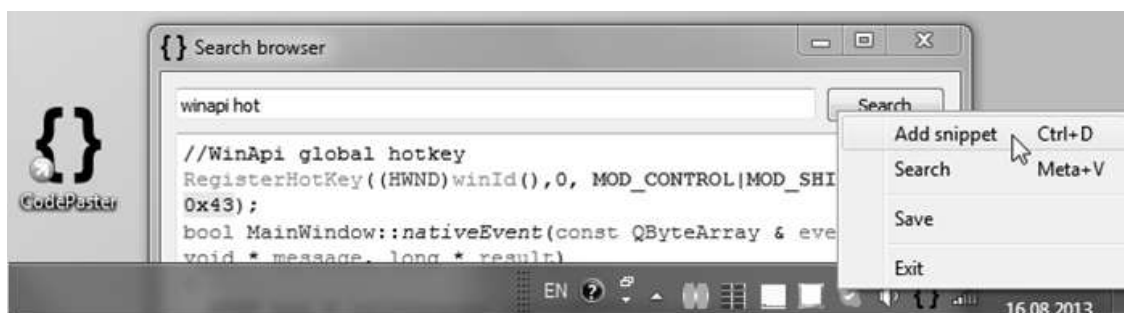


Рис. 1. Значок программы в области уведомления, контекстное меню и окно поиска

CodePaster распространяется по лицензии GNU General Public License v3 (Универсальная общественная лицензия GNU). Среди поддерживаемых операционных систем – тройка самых популярных для настольных персональных компьютеров систем - Microsoft Windows, Linux и Mac OS.

Программа рассчитана на запуск вместе с операционной системой (через автозагрузку) и постоянное пребывание в фоновом режиме. Работать с программой можно двумя способами – через глобальные горячие клавиши, или же через контекстное меню. Предпочтителен первый, так как он значительно повышает скорость работы и позволяет использовать CodePaster совместно с любой средой разработки.

На рисунке 2 представлена наглядная схема взаимодействия с базой данных. База на жёстком диске представлена в одном файле, который содержит иерархическую модель. Для её реализации используется расширяемый язык разметки (XML). Система управления базой данных (СУБД) написана специально для этой программы на языке C++.

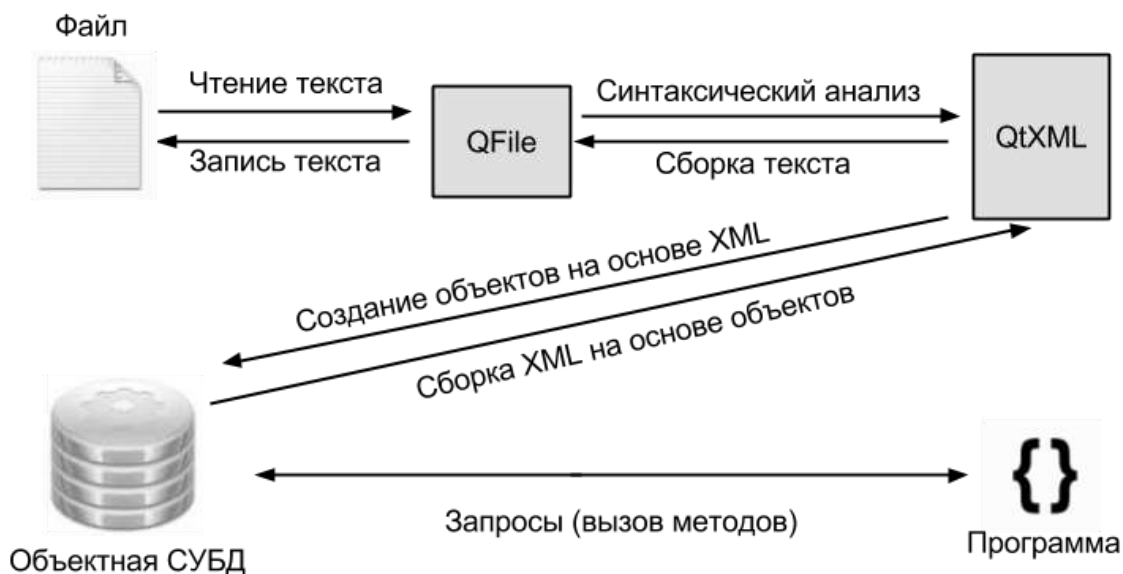


Рис. 2. Схема взаимодействия приложения с базой данных

При открытии программы происходит однократное побайтовое считывание XML-файла. Файл содержит текст в кодировке UTF-8 (Unicode Transformation Format). Его можно легко открыть и отредактировать в стороннем текстовом редакторе (например, в блокноте). Текст содержит корневой дескриптор codeBase, который содержит в себе множество дочерних дескрипторов code. Каждый такой дескриптор представляет фрагмент исходного кода и содержит два атрибута – для ключевых слов и для даты добавления фрагмента. А тело дескриптора – сам фрагмент кода, при этом специальные символы (например, угловые скобки) заменены соответствующими XML-сущностями.

После считывания файла мы имеем некоторый XML-документ в текстовом виде. Средствами модуля QtXML происходит синтаксический анализ текста, чтобы получить DOM – объектную модель документа. Если анализ текста прошёл без ошибок, мы можем далее проанализировать непосредственно наш XML-документ. Программа проходит по каждому дескриптору code и создаёт на его основе C++ - объект Code, переводя XML - атрибуты в поля объекта.

Таким образом, мы имеем вектор из объектов Code – это и есть наша база данных. В оперативной памяти через СУБД происходит работа с ней из программы – добавление новых фрагментов кода, просмотр существующих, удаление. При сохранении базы программа преобразует её из объектной модели обратно в текстовый вид и сохраняет в единый XML-файл. Сохранение происходит автоматически с определённым периодом (пять минут), а также по закрытию программы.

Для программы была создана специальная поисковая система, которая обрабатывает некоторую строку-запрос и ищет в базе фрагменты кода, которые ей удовлетворяют. При этом пробел между словами в запросе служит логическим умножением (операция И), позволяя сужать список найденных фрагментов кода. Алгоритм поиска работает следующим образом – программа в цикле проходит по всем объектам Code и для каждого объекта решает, включить его в выдачу или нет. Для включения фрагмента кода в нём (или же в его ключевых словах) должны содержаться все слова из строки запроса.

Программа показала себя эффективным средством решения задачи оптимизации рутинных действий при написании кода.