

Использование программы «Трепатор» для написания статей в ручном режиме (Полусказка)

Иосиф Каршенбойм.
www.iosifk.narod.ru

Сказка – ложь, да в ней намек...

Сталкивались ли Вы с теми из наших коллег, которые на предложение написать статью для технического журнала отвечают примерно так: «Написать красиво программу мы можем, а вот статью – нет!» Сейчас здесь будет показано, что написать статью - это очень просто. Причем показано будет именно для специалистов, хорошо разбирающихся в программировании на C++.

Введение.

Автору этой статьи часто приходится читать статьи, написанные начинающими авторами. Так же приходится делать рецензии на статьи, написанные коллегами. И вот что оказалось интересного в этом деле. Естественно, что молодые писатели пишут статьи по-старинке, вручную. Поэтому тексты таких статей оказываются слабо структурированными и читать их порой достаточно тяжело. Автор уже давно отошел от подобной практики. Если внимательный читатель посмотрит на тексты, написанные автором, то он заметит жесткое структурирование текстов. Статьи написаны строго в определенном порядке. А именно в том порядке, как автор был обучен для подачи заявок на изобретения во ВНИИ Государственной Патентной Экспертизы. Далее вопрос о структурировании текстов статей будет рассмотрен более подробно, а пока мы ограничимся только одним примером. Вот только один пример того, как надо располагать текст. Сначала описывается объект, потом – действия над ним. Сначала описывается причина и только потом – следствие, которое порождается этой причиной. Так и только так, а не в обратном порядке. «Потому что вокруг было холодно, оказалось, что это зима...» - вот частая ошибка начинающих. Так как же достигается правильный стиль изложения материала, о котором говорит автор, когда техническая статья читается так же легко, как и занимательный рассказ.

Далее, эта статья продолжит пост, отправленный автором в телеконференцию сегодня, 01.04.2006г. В дебатах тогда произошли горячие обсуждения о том, как правильно надо писать статьи. И в этом посте и только в тот день автор впервые сообщил широкой общественности о том, что для написания статей он использует программу, которую он назвал «Трепатор» (ударение надо делать на букве «А»). Что же это такое? Впервые идея об автоматической генерации статей была вычитана автором из книги по языку программирования Форт. С тех пор прошло много лет и вот, наконец, эта идея была успешно реализована. Для программы исходными данными являются ключевые слова, такие как: «микроконтроллер», «линкер», «порт» и так далее. Огромная база данных, пополняемая из импортных и отечественных публикаций, позволяет автоматически и довольно быстро генерировать любые тексты, по содержанию и по стилю практически ничем не отличимые от многих других статей, написанных «писателями среднего качества». А при выполнении программой функции «высокое качество», компьютер автоматически сканирует Интернет, выполняет поиск файлов по заданной тематике, делает перевод документации и автоматически пополняет свою базу данных. Когда база

данных превышает требуемый по заданной тематике объем, программа автоматически переходит в режим формирования текста статьи. При таком режиме работы автоматически сгенерированные тексты по качеству и по объему представленной информации уже приближаются к лучшим публикациям в отечественной периодике. Единственным недостатком данного способа работы является повышенный Интернет-трафик и огромный расход машинного времени. Конечно, автор не может раскрыть все тайны «ноу-хау», так же как не может сделать свою программу Freeware (надеюсь, не надо объяснять читателям, почему автор этого сделать не может), поэтому вниманию читателей может быть представлен только алгоритм создания статей, да и то в сильно упрощенном виде. Однако заинтересованный читатель всегда сможет выполнить все действия над текстами в «ручном режиме», то есть, используя данную методику и любой текстовый редактор, например Ворд.

Как это делает программа?

Формирование служебной информация для статьи.

Программа создает образ статьи, который можно описать следующим образом. Кстати, здесь будет сделано специальное замечание для рецензентов и прочих критиков, которые будут ругать эту статью. В тексты программ, которые приведены здесь, заведомо внесены ошибки, для исключения несанкционированного копирования. Итак, текст первой автоматически сгенерированной статьи представлен читателям на рисунке 1.

```
* Copyright 2003 Trepator. All rights reserved.
* $Revision: 1.1 $
*****/
#include <iostream>
#include "article.h"

int main(void)
{
    // Create article objects.
    article my_article1;
    // выводим на печать
    cout << my_article1();
}
Остальной текст не приводится!
```

Рис.1.Текст программы для формирования статьи.

Смотрим на рис.1. Все предельно ясно. А почему? Да потому, что при описании программ используется язык описания, очень сильно структурированный. Чем это хорошо? Когда жестко заданы конструкции языка и порядок их следования, то пользователь языка получает возможность легко воспринимать то, что написано в тексте программы. Кроме того, любое отклонение от жестко заданной структуры легко определяется и трактуется как ошибка. А, следовательно, любая ошибка может быть обнаружена и устранена. Для сравнения приведем всем известную фразу: «казнить нельзя помиловать». В приведенном примере показано, что в случае неструктурированного текста отделить правильные фразы от неправильных – невозможно!

Теперь начнем рассматривать приведенный выше пример еще раз, но теперь уже как пример структурированного текста. В самом начале следует служебная информация. Запоминаем: нам надо поместить в начало статьи свою служебную информацию и копирайт. Выглядит это следующим образом. Вместо имени файла в ручном режиме

«трепации» проекта надо написать большими буквами название статьи. Ну, например так, как показано на рис.2.

Наша первая статья.
Автор_который_написал_статью.
Почтовый_адрес_автора_который_написал_статью.

Рис.2.Текст заголовка статьи.

Теперь приступаем к следующему этапу. Опять смотрим на рис.1. Следующее, что необходимо выполнить, это сформировать инклюдные файлы. Как известно инклюдные файлы – это обычно библиотечные файлы, в которых описано то, что является известным и необходимым для данного проекта. Однако инклюдные файлы для статьи должны находиться не в начале статьи, как это принято во всех нормальных языках программирования, а в конце. Тут, господа читатели, сделать совершенно ничего невозможно, поскольку статьи возникли в докомпьютерную эру и сломить эту привычку читателей на данном этапе совершенно невозможно. Пример подключения инклюдных файлов к статье приведен на рис.3.

Присоединение инклюдных файлов начинается со служебной фразы «Литература». Формат присоединения инклюдного файла к статье начинается не со знака «#» а с цифры, так как указано на рис.3. Особенностью инклюдных файлов в статье является то, что в силу сложившихся традиций практикуется непосредственная расстановка ссылок на инклюдные файлы в тексте статьи. Например, если мы хотим показать, что написал автор данной статьи в 2003 году, то мы должны написать: «контроллеры Fast Ethernet для встроенных применений, см. Л[3]»

Литература.

1. KS8001-ds.pdf. KS8001 1.8V, 3.3V 10/100BASETX/FX Physical Layer Transceiver DATASHEET V 1.01. micrel.com
2. Application Note 120. Capacitive Coupling Ethernet Transceivers without Using Transformers. Micrel 10/100 Ethernet Products
3. Иосиф Каршенбойм. Контроллеры Fast Ethernet для встроенных применений. Компоненты и технологии, № 5'2003.

И так далее...

Рис.3.Текст инклюдных файлов статьи.

Формирование основной части статьи.

Теперь приступаем к основному этапу. Смотрим на рис.1 и ищем слово «main». Следующее, что необходимо выполнить, это нам надо понять, что же будет представлять понятие «main» в новой статье. Когда автор этой статьи был студентом, то институтский куратор просто и доступно объяснил то, как студентами пишутся дипломы. «Представьте себе олимпийский рекорд. Как делается новый рекорд – да добавляется 1,5 см к старому и спортсмен прыгает. Вот так и дипломники – к старому диплому добавляют 2 страницы и ... Только разница в том, что дипломники ставят совой «рекорд» с первого раза». Поступим аналогичным образом.

Четко представим себе то, о чем мы хотим сообщить читателям, что будет «изюминкой» новой статьи. Из всего объема информации выделим то, что уже известно. Это будет называться аналог. Если аналогов несколько, то выделим наиболее близкий аналог. Такой аналог, по методикам ВНИИГПЭ называется «прототип». То, о чем мы хотим писать назовем «наш объект». Чтобы правильно описать в статье аналоги и

прототип, программа формирует класс и несколько экземпляров этого класса – аналоги, прототип и «наш объект».

Как устроен этот класс? Сначала идет инициализация объекта. То есть сначала описывается, что представляет собой член класса: где он находится, как выглядит, на что похож. Здесь автор новой статьи должен показать читателю то, что он хочет описать, к примеру, что есть программа ХХ, на ее основном окне расположены кнопки управления, перечислить их и предварительно объяснить, для чего они нужны. Например, «группа кнопок для редактирования».

Следом за этим надо описать соединение описываемых частей. То есть как одни меню соединены с другими, как из одного окна попасть в другое.

И только потом, как и положено, описывается, что и где делается с членами класса, чтобы выполнялись требуемые функции. Вот только в этом месте статьи автор должен показать читателю, к примеру, как надо нажимать на кнопку программы (например той, что мы бы хотели описать в генерируемой нами статье) и что при этом произойдет.

И если при описании аналога и прототипов все уже известно, то при описании объекта - «наш объект», необходимо подробно выделить то самое новое, что будет являться существенным отличительным признаком, отделяющим «наш объект» от прототипа. И эти отличия необходимо описать так, чтобы у читателя не возникало вопросов о том, что именно хотел сказать в этой части статьи автор. И еще немного по технике создания статьи. При описании любого объекта из описываемого здесь класса, необходимо упомянуть, что объекты эти сами состоят из более мелких объектов, называемых «абзац текста» и «рисунок». Что касается понятия «рисунок», то здесь вроде все понятно, поскольку рисунок не «дробится» на более мелкие составляющие. А вот «абзац текста» - это довольно сложное «сооружение». Основная идея каждого такого фрагмента – это то, что он должен содержать одну законченную идею. (Автор данной статьи не хочет оперировать в терминах начальной школы, т.к. это, по его мнению, приведет только к отторжению данной методики от широких читательских масс). В заключение высказываний о мелких объектах, называемых «абзац текста» и «рисунок» хочется напомнить, что каждый такой мелкий объект должен быть как можно более запоминающимся и полностью доводить мысль автора до утомленного читателя. Для примера, автор данной статьи позволит себе привести образцы таких мелких объектов, вычитанных в журнале «Крокодил» более 30-ти лет назад (и до сих пор они в памяти). Вот пример того, что здесь называется «абзац текста»: «Председатель колхоза сумел ухватить доярок за живое, в результате чего надои молока в колхозе резко увеличились. Из протокола общего собрания колхоза». А вот пример того, что здесь называется «рисунок». К сожалению, сам рисунок за давностью лет автор привести не может. Но может его описать. Итак, на рисунке был изображен здоровый детина с татуировкой на груди – «Нет счастья в жизни с 8-ми до 5-ти». Вот так коротко и ясно! Господа, начинающие писатели, учтите, что картинки должны занимать столько же места, сколько и текст, который было бы необходимо написать, чтобы объяснить читателям суть того, что изображено на картинке.

Подводя итог, можно сделать следующую блок-схему, см. рис.4.

Тип объекта	Объект	Часть объекта	Пример
Аналог 1	<i>Что такое аналог 1</i>	инициализация	
	<i>Как связаны его составляющие и как он работает</i>	описание	
Аналог 2	<i>Что такое аналог 2</i>	инициализация	
	<i>Как связаны его составляющие и как он</i>	описание	

	<i>работает</i>		
Прототип	<i>Что такое прототип</i>	инициализация	
	<i>Как связаны его составляющие и как он работает</i>	описание	
«наш объект»	<i>Что такое «наш объект»</i>	инициализация	
	<i>Как связаны его составляющие и как он работает</i>	описание	

Рис.4. Структура основной части статьи.

И не забудьте о разделах. Это такие небольшие разделители текста, которые позволяют читателю вспомнить, в каком месте статьи его оторвали от чтения статьи. Раздел должен сообщать читателю, что его ждет в следующем далее разделе. Разделы можно нумеровать цифрами, можно давать им названия. Можно так же делать подразделы для более детального разбиения текста на части. Внешне это должно выглядеть так, как показано на рис.5

Раздел_Название_Раздела.

Здесь далее следует текст данного раздела.

Рис.5. Так должен выглядеть один раздел статьи.

Если в сейчас еще раз внимательно рассмотреть рис.1, то мы заметим строчки начинающиеся со знаков «//». Нет нужды объяснять, что это комментарии. Программа «трепатор» переводит эти комментарии как два фрагмента статьи – введение и заключение. Раздел, заключающий статью называется «Выводы». Внешний вид этих фрагментов представлен на рис.6.

Введение.

Здесь далее следует текст данного фрагмента.

Рис.6а. Так должен выглядеть раздел - «Введение».

Выводы.

Здесь далее следует текст данного фрагмента.

Рис.6б. Так должен выглядеть раздел - «Выводы».

Для чего пишутся два эти фрагмента, спросит нетерпеливый читатель, если в статье и так все подробно расписано? Ах, уважаемый читатель! Увы и ах! Далекое не все хотят читать Ваш труд. Но многие захотят показать, что они его читали и даже поняли, то о чем Вы написали в своей статье. Поэтому они читают только эти две части статьи, в которых в очень сокращенной форме говорится о том, какой Вы «крутой» писатель, какую чудесную статью Вы представили читателям, и какие молодцы читатели, если они ее прочли. Вот эту то часть статьи и используют для режима «быстрого ознакомления со статьей». Поэтому, следуя вековым традициям, приходится очень аккуратно генерировать эти два фрагмента.

Ну и заканчивая все эти рассуждения о том, как сделать написание статей легким и прибыльным делом при использовании режима ручной «трепации», надо свести все элементы вместе. Тогда любая статья должна выглядеть так, как показано на рис.7.

Наша первая статья.

Автор_который_написал_статью. Почтовый_адрес_автора_который_написал_статью.			
Введение. <i>Здесь далее следует текст данного фрагмента.</i>			
Тип объекта	Объект	Часть объекта	Пример
Аналог 1	<i>Что такое аналог 1</i>	инициализация	
	<i>Как связаны его составляющие и как он работает</i>	описание	
Аналог 2	<i>Что такое аналог 2</i>	инициализация	
	<i>Как связаны его составляющие и как он работает</i>	описание	
Прототип	<i>Что такое прототип</i>	инициализация	Раздел <i>Текст раздела.</i>
	<i>Как связаны его составляющие и как он работает</i>	описание	Раздел <i>Текст раздела.</i>
«наш объект»	<i>Что такое «наш объект»</i>	инициализация	Раздел <i>Текст раздела.</i>
	<i>Как связаны его составляющие и как он работает</i>	описание	Раздел <i>Текст раздела.</i> Раздел <i>Текст раздела.</i>
Выводы. <i>Здесь далее следует текст данного фрагмента.</i>			
Литература. 4. KS8001-ds.pdf. KS8001 1.8V, 3.3V 10/100BASETX/FX Physical Layer Transceiver DATASHEET V 1.01. micrel.com 5. Application Note 120. Capacitive Coupling Ethernet Transceivers without Using Transformers. Micrel 10/100 Ethernet Products 6. Иосиф Каршенбойм. Контроллеры Fast Ethernet для встроенных применений. Компоненты и технологии, № 5'2003. <i>И так далее...</i>			

Рис.7. Структура всей статьи в целом.

Выводы.

Ну, вот и все, уважаемый читатель. Заканчивается этот длинный день (см. дату в начале статьи), и завтра все будет по-другому. Завтра уже никто и не вспомнит о режиме «ручной трепации», а жаль, ей богу. Дело то хорошее и поучительное. До свиданья, и ровно через год, день в день, возможно, будет написано продолжение.

Copyright (э) 2005 Иосиф Каршенбойм iosifk@narod.ru <http://www.iosifk.narod.ru>

Перепечатка в Интернете разрешается только с сохранением копирайта и со ссылкой на www.iosifk.narod.ru.

Публикация в офлайновых изданиях разрешается только после согласования с Каршенбоймом - iosifk@narod.ru