

Программная инженерия

Андрей Дмитриев

andrei-dmitriev@yandex.ru

<http://improve-it.org/>

©2008-2010



Оформление программного кода

План

- Введение
 - Причины появления
 - Основные аспекты оформления
 - Выводы
-

Соглашения об оформлении

- *coding standards, coding convention, programming style*
 - набор правил и соглашений, используемых при написании исходного кода на *некотором* языке программирования
-

Причины появления

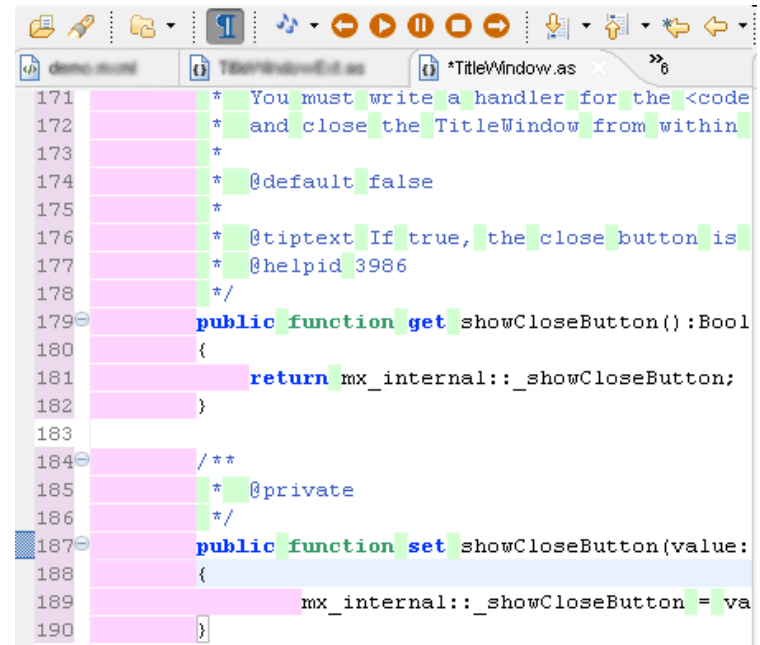
- Принимается группой или сообществом разработчиков с целью единообразного оформления кода проекта(ов)
 - Единство позволяет визуально выделять значимые блоки
 - Стил ь оформления может защищать от ошибок программирования
 - Проще разобраться в коде

Что описывают соглашения

- ??

Что описывают соглашения

- Имена переменных и т.д.
- Отступы
- Комментарии
- Оформление блоков (скобки, переводы строк)
- Использование пробелов
- Длины строк, пустые строки
- Бинарные операторы
- ...



```
171 * You must write a handler for the <code>  
172 * and close the TitleWindow from within  
173 *  
174 * @default false  
175 *  
176 * @tipText If true, the close button is  
177 * @helpid 3986  
178 */  
179 public function get showCloseButton():Bool  
180 {  
181     return mx_internal::_showCloseButton;  
182 }  
183  
184 /**  
185 * @private  
186 */  
187 public function set showCloseButton(value:  
188 {  
189     mx_internal::_showCloseButton = va  
190 }
```

Общие правила (1/2)

- Код чаще читается, чем пишется
 - Делайте идентификаторы в меру длинными
 - Избегайте потенциальных конфликтов имен в публичном API

Общие правила (2/2)

- Делать НЕ стоит:
 - ❑ Использовать малопонятные префиксы и суффиксы
 - ❑ Использовать символ подчеркивания
 - ❑ Давать имена, отличающихся регистром букв

Стили оформления

- **Паскаль** – первая буква заглавная и все последующие первые буквы слов тоже заглавные. Например, **BackColor**, **LastModified**, **DateTime**
- **Кэмел** – первая буква строчная, а остальные первые буквы слов заглавные. Например, **borderColor**, **accessTime**, **templateName**
- Какой стиль принят в C#? В Java?

Пространства имен

- Можно использовать имя компании или назначение проекта:
 - `namespace MyCompany.Controls {}`
 - `namespace MyCompany.Collections {}`
 - `package mycompany.controls;`
 - `package mycompany.collections;`

Имена

- Имена классов обычно должны быть существительными (File)
- Имена интерфейсов – прилагательные или существительные (Readable)
- Имена методов – глаголы (remove)

Параметры

- Делайте имена значимыми
- При совпадении имени параметра с именем поля, разделяйте их с помощью префикса `this`.
- `Type GetType(string typeName);`
- `string Format(string format, params object[] args);`

Комментарии

- Многострочные
или однострочные?
- Пишите, что код
делает, а не какая
операция
выполняется

```
##### World Cup Soccer Data 'soccer.txt' #####

##### Read datafile into R #####
soccer<-read.table("soccer2002.txt", header=TRUE)

#### Sample size ####
sampleN<-sum(soccer$Freq)

#### Sample mean ####
smean<-(1/sampleN)*(sum(soccer$Goals*soccer$Freq))

#### Poisson probability for X=1, P(X=1) #####
dpois(1, lambda=smean)

#### Poisson probabilities for X=0, 1, ..., 8 with LAMBDA=smean ####
prob<-dpois(0:8, lambda=smean)
prob

#### Compute expected frequencies #####
efreq<-sampleN*prob
efreq
```

Волшебные числа

- Не используйте литеральные константы
 - магические числа
 - зашитые в код размеры буферов
 - времена ожидания
 - Определите константу или переменную только для чтения
-

ВЫВОДЫ

ССЫЛКИ

- «Совершенный код» С. Макконнелл
- RSDN стиль
 - <http://www.rsdn.ru/article/mag/200401/codestyle.XML>
- Java code conventions
 - <http://java.sun.com/docs/codeconv/>
- Стандарт кодирования GNU (Столлман)
 - http://www.opennet.ru/docs/RUS/coding_standard
- C++
 - <http://geosoft.no/development/cppstyle.html>

Q&A

Спасибо!

Андрей Дмитриев
andrei-dmitriev@yandex.ru
<http://improve-it.org/>
©2008-2010
