Инструкция по стилю кодирования (v.4.0)

# Стили наименования символов

Требуется визуально разделять символы, относящиеся к различным категориям. В префиксных стилях (название стиля начинается со слова prefix) использование префикса, указывающего на тип обязательно, в непрефиксных – недопустимо. Для непрефиксных типов допускается использование специальных префиксов, чья роль создание дополнительных подстилей.

Возможные стили именования:

|  |  |
| --- | --- |
| Стиль | Пример |
| PREFIX\_UPPER\_CASE\_STYLE | STR\_ITEM\_NAME, NODE |
| prefix\_underscore\_style | str\_item\_name, node |
| camelStyle | itemName, node |
| prefixCamelStyle | strItemName, node |
| DelphiStyle | ItemName, Node |
| PrefixDelphiStyle | StrItemName, Node |
| Underscore\_Delphi\_Style | Item\_Name, Node |
| Prefix\_Underscore\_Delphi\_Style | Str\_Item\_Name, Node |
| prefix\_Underscore\_Delphi\_Style | str\_Item\_Name, Node |

В проекте для именование различных видов символов должны использоваться следующие стили:

|  |  |  |
| --- | --- | --- |
| Категория символа | Отличать от | Стиль |
| Константы | символов, которым можно присваивать значение | PREFIX\_UPPER\_CASE\_STYLE |
| Методы | от свойств и глобальных полей | camelStyle |
| Имя класса |  | DelphiStyle |
| Свойства | локальных переменных | DelphiStyle |
| Публичные поля | локальных переменных | DelphiStyle |
| События | свойств, методов | OnDelphiStyle |
| Локальные переменные | полей, связанных со свойствами  полей, не связанных со свойствами | prefix\_underscore\_style - **допускается использование одного префикса** |
| Непубличные поля не связанные со свойствами | локальных переменных полей, связанных со свойствами | prefixCamelStyle – **не допускается использование одного префикса!** |
| Статические непубличные поля не связанные со свойствами | Нестатических полей, не связанных со свойствами | prefixCamelStyle – **не допускается использование одного префикса!** |
| Поля связанные со свойствами | локальных переменных полей, не связанных со свойствами | FDelphiStyle |
| Параметры метода | локальных переменных, полей, свойств | ADelphiStyle или AnDelphiStyle если первый звук корня – гласный. Для булевых параметров допускаются стили ToDelphiStyle и IsDelphiStyle |

# Используемые префиксы

Для именования символов, для которых установлен префиксный стиль, требуется использовать следующие префиксы:

* для базовых типов:

|  |  |
| --- | --- |
| Базовый тип | Префикс |
| int | n |
| Number | dbl |
| Boolean | is, to |
| String | str |

* для массивов, коллекций, множеств, векторов используется составной префикс:

a+тип элемента

пример для вектора строк (Vector.<String>): astr

* для других типов используются префиксы производные от имени типа

пример для типа Point: pnt

# Выделение операций

Аддитивные операции, операции присвоения, сравнения, а также логические булевы операции выделяются с обоих сторон пробелами. Примеры:

a += c; b = (c > a) && b; c -= a\*c - a;

# Шаблон модуля

Устанавливается следующий шаблон модуля:

package {

//import <#Package#>;

public class <#DelphiStyle#>

{

//public fields & consts

//public consts

/\*public const <#PREFIX\_UPPER\_CASE\_STYLE#>: <#Type#>;\*/

//public fields

/\*public var <#DelphiStyle#>: <#Type#>;\*/

//events

/\*<#OnDelphiStyle#>: CustomDispatcher = new CustomDispatcher();\*/

//protected fields & consts

//protected consts

/\*protected const <#PREFIX\_UPPER\_CASE\_STYLE#>: <#Type#>;\*/

//protected fields

/\*protected var <#prefixCamelStyle#>: <#Type#>;\*/

//private fields & consts

//private consts

/\*private const <#PREFIX\_UPPER\_CASE\_STYLE#>: <#Type#>;\*/

//property fields

/\*private var <#FDelphiStyle#>: <#Type#>,\*/

//other fields

/\*private var <#prefixCamelStyle#>: <#Type#>;\*/

//properties

/\*public function get <#DelphiStyle#> ():<#Type#>

{

return ...;

}\*/

//methods

//constructor

/\*public function <#DelphiStyle#> (<#ADelphiStyle#>: <#Type#>)

{

var <#prefix\_underscore\_style#>: <#Type#>;

}\*/

//public methods

/\*public function <#camelStyle#>(<#ADelphiStyle#>: <#Type#>)

{

}\*/

//protected methods

/\* protected function <#camelStyle#>(<#ADelphiStyle#>: <#Type#>)

{

}\*/

//private methods

/\* private function <#camelStyle#>(<#ADelphiStyle#>: <#Type#>)

{

}\*/

//event handlers

/\* private function <#object\_OnDelphiStyle#>(<#AnDelphiStyle#>: Event): void

{

}\*/

}

}

# Применение стиля кодирования

С момента введения стиль кодирования применяется для нового и изменяемого кода, а также во время рефакторинга.