## Оформление смысловых блоков

Каждая команда псевдокода преобразуется в однострочный комментарий. После этого комментария пишется исходный код на выбранном языке программирования (в нашем случае это C#). Комментарий и последующий исходный код образуют единый блок «комментарий+код», который далее будем называть смысловым блоком.

Схема оформления смысловых блоков:

// комментарий к смысловому блоку 1

команда

команда

…

команда

// комментарий к смысловому блоку 1

команда

команда

…

команда

…

Смысловые блоки отделяются друг от друга пустыми строками. Это позволяет программисту быстро ориентироваться в исходном коде. Прочитав комментарий в начале смыслового блока, программист может определить, нужно ли ему погружаться в чтение исходного кода этого блока или же следует перейти к следующему смысловому блоку. Найти следующий смысловой блок не составит труда, т.к. он имеет аналогичный уровень отступов, отделен от текущего блока пустой строкой и имеет однострочный комментарий в начале. Учитывая, что смысловой блок обычно не делается громоздким (если он превышает пару десятков строк, то имеет смысл задуматься о его выделении в отдельный метод), поиск начала следующего блока для опытного программиста не превышает секунды.

Со следующим смысловым блоком происходит то же самое. В результате программист, читая только текст комментариев на родном для него языке и не читая ни одной команды языка программирования, может очень быстро перемещаться по исходному коду и, соответственно, быстро найти интересующий его участок. Это главное достоинство такого оформления исходного кода.

## Оформление методов

Метод, как и смысловой блок, должен иметь в начале однострочный комментарий. Как показывает практика, одной строки вполне достаточно, чтобы кратко описать функционал/назначение метода. Особенно если учесть, что при написании исходного кода следует придерживаться принципа единственной ответственности как по отношению к классу, так и по отношению к его методам.

Если одной строки комментария не хватает, то это значит, что вы либо пытаетесь перечислять выполняемые внутри метода действия (тем самым перегружаете ненужной информацией комментарий), либо взваливаете на метод сразу несколько ответственностей (тем самым создаете плохо сопровождаемый исходный код).

Рекомендуемая схема оформления методов:

// однострочный комментарий к методу

ключевые\_слова возвращаемый\_тип ИмяМетода (параметры)

/\*

Параметры:

параметр1 – описание параметра 1

параметр2 – описание параметра 2

…

параметрN – (вых) описание параметра N

Возвращаемое значение:

Описание, что возвращается, либо перечисление возвращаемых

значений и условий, при которых они возвращаются.

\*/

{

// комментарий к смысловому блоку 1

команда

команда

…

команда

// комментарий к смысловому блоку 2

команда

команда

…

команда

…

}

Если параметров у метода больше двух, то читать их вытянутыми в одну строку становится неудобно. В этом случае удобнее перечислить их друг под другом:

// однострочный комментарий к методу

ключевые\_слова возвращаемый\_тип ИмяМетода

(

параметр1,

параметр2,

…

параметрN

)

Если параметров несколько, то они перечисляются в следующем порядке:

* входные;
* входные-выходные;
* выходные.

## Оформление команды switch

Каждая ветвь case является, по сути, отдельным смысловым блоком и логичнее отделить ее от других ветвей case пустой строкой. Только комментарий лучше поставить не перед первой строкой блока, а в конце строки case.

Выравнивание команд ветви внутри каждого case лучше сделать по началу значения, указанного после case – так удобнее читать и ориентироваться в исходном коде.

switch (...)

{

case "1": // <Комментарий>

<код для данного case>

case "2": // <Комментарий>

<код для данного case>

case "3": // <Комментарий>

<код для данного case>

default : // <Комментарий>

<код для default >

}