Code Style Guidelines

##### 遵循的原则

经过组内成员的讨论,我们小组准备遵循的代码风格主要需具备以下特点:

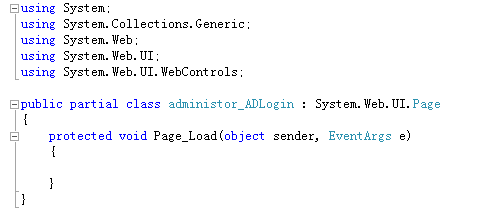
1. 命名符合主流的编程规范,这里的规范包括了变量名,函数名,类名等名字的前缀限制,开头字母大小写限制等;
2. 代码的布局符合主流的编程规范,这个规范主要包括大括号的布局,行与行的分隔,空格的使用等;
3. 命名应该是有意义的.这点要求组员在编写代码时,对于变量等的命名,不能是一时毫无根据的空想,也不应是晦涩难懂的词语,而是选择那些意思简单明了的词.

至于其他关于代码风格的经典规范(文档,注释等方面),考虑到以下几个原因:

1. 成员水平和精力时间的限制;
2. 自动检查工具对于这一方面的严格;
3. 有一部分代码是vs自动生成的,这部分代码并不需要注释;

我们小组内暂时不做硬性要求,但会要求组员对于自己编写的类和方法书写注释,虽然这部分不会被检查工具检查.

代码示例——如下vs自动生成的代码是符合我们的原则的：

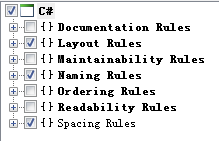


##### 执行的方法

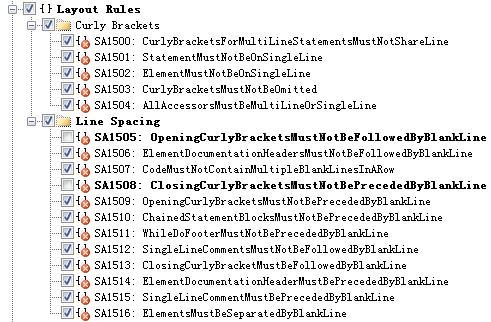
对于如何强制不同的组员都遵循统一的原则,前人已经为我们提供了解决方案.那就是使用代码风格检查工具来强制遵循指定的规则集合.

由于我们是使用asp.net框架,大部分代码是c#代码,所以我们使用工具StyleCop. StyleCop的安装和使用非常简单,安装之后就相当于vs的一个插件,在要检查的源文件上右击,便可以看到检查的选项.

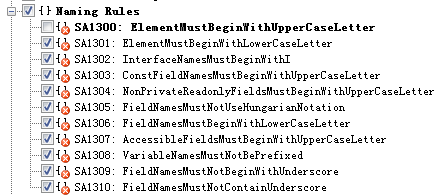
StyleCop提供了一系列规则来供用户选择适合自己的规则集合.按照我们小组的风格原则,我们选择了如下的规则:



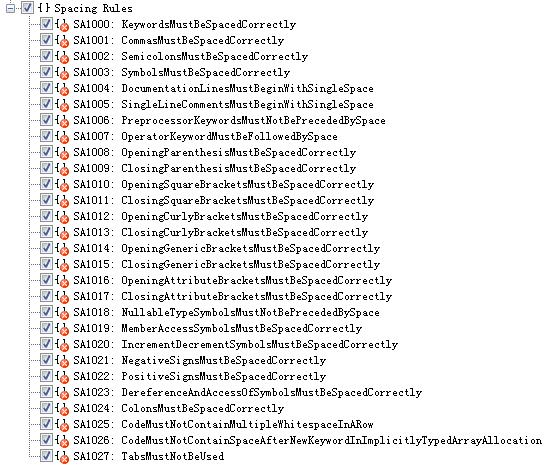
——所选的规则集合概览



——代码布局规则集合



——命名规则集合



——空格的使用规则

##### 关于审查

对于审查代码风格,我们认为每个组员应当在编写代码时自觉遵循我们设定的原则,避免在之后又去审核代码风格,甚至更改代码.

对于可以直接使用工具检查的部分,在编写时使用工具自动检查;对于注释等部分采取组员自觉遵守和互相监督结合的方法.

组员间互相监督可以实时进行,在浏览他人编写的代码时,发现不符合风格的问题直接提出,并在小会议上和相应的组员进行沟通,而不是专门的去检查代码风格,为检查而检查.