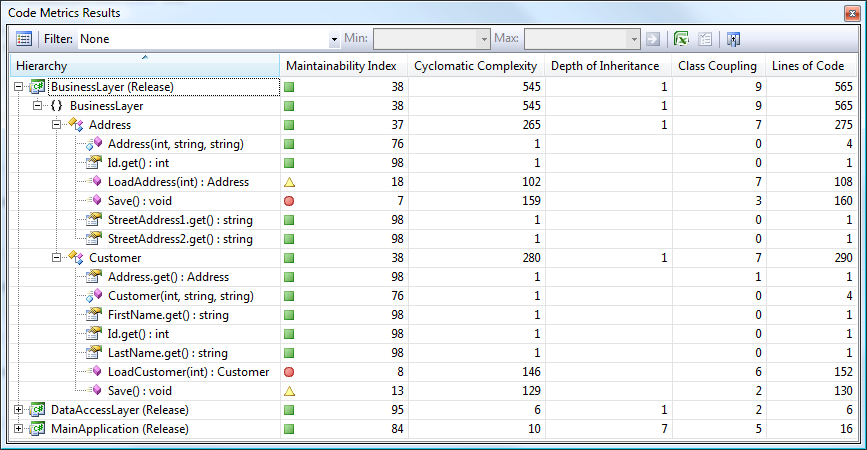
Novo para o Visual Studio 2008 - Code Metrics

Embora nós já falado brevemente sobre métricas de código [, quando anunciou que](http://translate.googleusercontent.com/translate_c?hl=pt-BR&prev=/search%3Fq%3Dvisual%2Bstudio%2B%2522Calculate%2Bcode%2Bmetrics%2522%26hl%3Dpt-BR%26lr%3D%26sa%3DX%26biw%3D1440%26bih%3D809%26prmd%3Divns&rurl=translate.google.com.br&sl=en&twu=1&u=http://blogs.msdn.com/fxcop/archive/2007/02/28/announcing-visual-studio-code-metrics.aspx&usg=ALkJrhilPcIYXd7wKcuBrf20BWfdoj3f8g) , eu pensei que iria iscutir o assunto de uma forma mais pequena profundidade, em especial as métricas que ele proporciona.

Como mencionado anteriormente, métricas de código é uma janela nova ferramenta que ajuda os usuários a encontrar e agir de acordo com áreas complexas e insustentável dentro de um aplicativo.

A seguir mostra os resultados da execução métricas de código em um aplicativo de negócios de ficção:

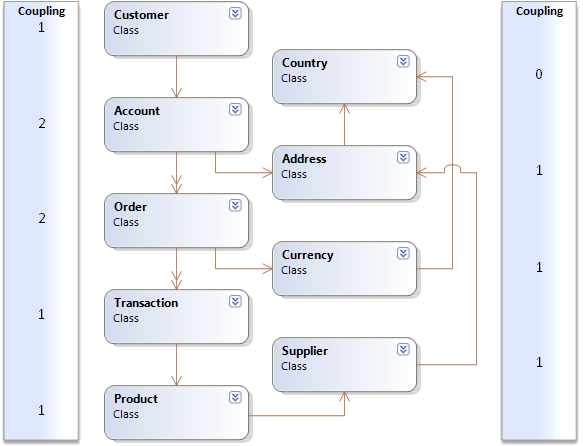


Como você pode ver de cima, para o Visual Studio 2008, estamos fornecendo cinco métricas. Estes são detalhados abaixo.

**Classe de acoplamento**

Em cada nível, que indica o número total de dependências que o item tem em outros tipos. Esse número exclui os primitivos e tipos internos como Int32, String e Object. Quanto maior este número, ondulação as mudanças mais prováveis ​​em outros tipos embora este item. Um valor mais baixo no nível do tipo podem indicar candidatos para possível reutilização.

O seguinte mostra como o acoplamento é calculado:

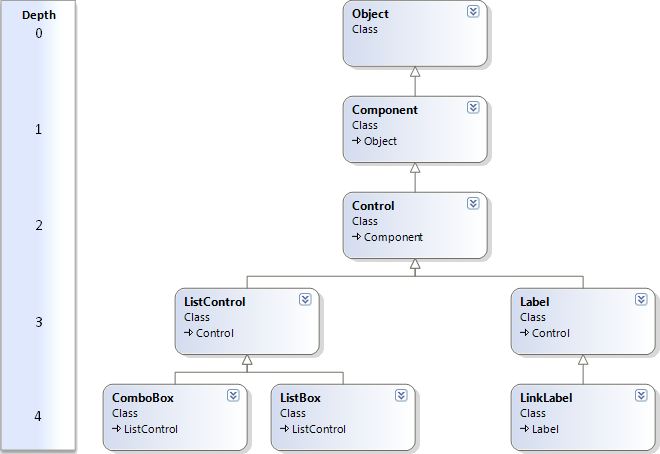


Por exemplo, como você pode ver acima, a **conta** é acoplado a dois outros tipos, **endereço** e da **Ordem,** enquanto o **País** não depende de qualquer outro tipo.

**Profundidade da Herança**

Ao nível do tipo, profundidade de herança indica o número de tipos que estão acima do tipo na árvore de herança.Por exemplo, um tipo que deriva diretamente de Object teria uma profundidade de herança de 1. Ao nível do espaço e do projeto, isto indica que a maior profundidade da herança de todos os tipos contidos dentro dele. Esse número não leva em consideração a profundidade de todas as interfaces implementadas. árvores de herança profunda pode indicar um excesso de engenharia de um problema e podem aumentar a complexidade dos testes e manutenção de uma aplicação.

O seguinte mostra como a profundidade é calculada:

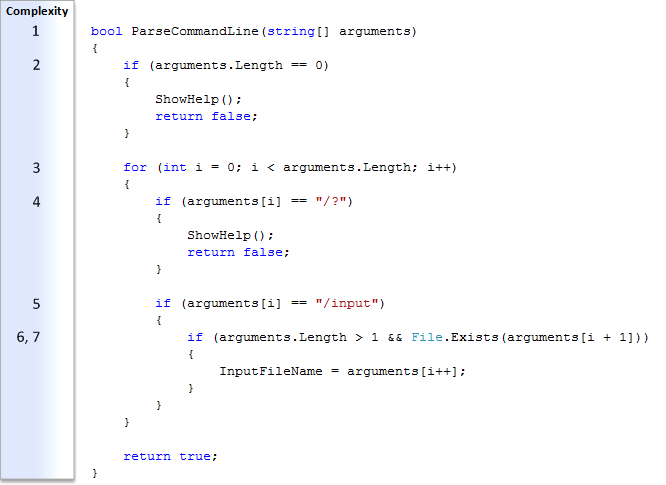


Por exemplo, na hierarquia de herança acima, **ListControl** e **Label** tem uma profundidade de herança, de 3, enquanto **componente** tem uma profundidade de herança de 1.

**Complexidade ciclomática**

Em cada nível, este mede o número total de caminhos individuais através do código. Este é basicamente calculado pela contagem do número de pontos de decisão (como **se os blocos,** alterne casos, e **fazer, ao mesmo tempo, foreach**  e **para** loops) e adicionando um. Esse número também é uma boa indicação sobre o número de testes de unidade que será necessário para atingir a cobertura completa linha. Menor é normalmente melhor.

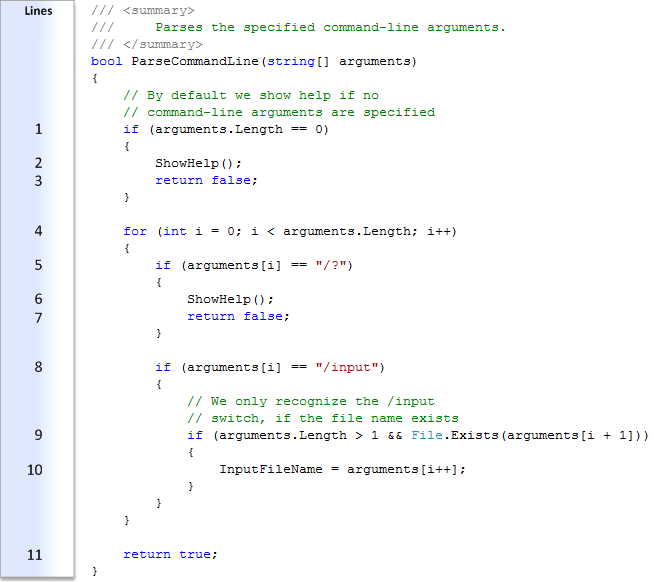
O seguinte mostra como a complexidade é calculado:



Linhas de Código

Em cada nível, esta é uma medida do número total de linhas de código executável. Isto exclui espaço em branco, comentários, suspensórios e as declarações de membros, tipos e espaços próprios. Menor é normalmente melhor.

O seguinte mostra como as linhas são calculadas:



Índice de Mantenabilidade

Ao nível dos Estados e do tipo, este é um índice 0-100, indicando a manutenção global do membro ou tipo. Ao nível do espaço e montagem, esta é uma média do índice de manutenção de todos os tipos contidos nela. Este índice é baseado em vários outros indicadores, incluindo *Halstead Volume* (quais os fatores do número e da utilização de operandos e operadores), Cyclomatic Complexidade e linhas de código. Um número baixo indica que o código é complexo e difícil de manter.

O Índice de Mantenabilidade coluna também inclui um ícone que dá uma rápida indicação quanto à manutenção geral e complexidade de um determinado item. A tabela a seguir mostra a escala em que um ícone é mostrado:

|  |  |  |
| --- | --- | --- |
| Ícone | Nível | Faixa |
| Verde | Alta Manutenibilidade | Entre 20 e 100, inclusive |
| Amarelo | Moderado Manutenibilidade | Entre 10 e 19, inclusive |
| Vermelho | Baixa Manutenibilidade | Entre 0 e 9, inclusive |

Estes ícones permitem-lhe ver rapidamente todos os pontos de problema que você deve começar a se centrar nos ou preenchendo um bug contra.

Conclusão

Agora que você sabe que essas medidas métricas, e como eles são calculados, a próxima coisa que você vai querer saber é o que você deve fazer quando você começa a ver algum vermelho em sua aplicação. Em um post futuro, vou cobrir por que deveria deveria se preocupar com cada métrica, e como proceder para corrigir alguns dos problemas comuns que causam as métricas especial para alto (ou baixo no caso do Índice de Mantenabilidade).