

...

Além do *identificador*, *tipo de dado* e *valor contido*, as variáveis e constantes possuem também **escopo**, que quer dizer “visibilidade” ou “abrangência” dentro de um módulo do código-fonte. O escopo define em que partes do programa um elemento pode ser acessado ou referenciado, e pode ser: **local** ou **global**. Elementos com escopo **local** só podem ser acessados dentro do módulo que o definiu, ao passo que os com escopo **global** podem ser acessados de qualquer parte do programa; e para ilustrar estas situações vamos criar um programa denominado “PrEscopos”. Para os programas compostos de módulos, como nas linguagens orientadas a procedimentos, cada um desses módulos (aqui tratados como *métodos*) pode ter seus próprios elementos *locais*, acessados somente dentro de cada um desses módulos; e no paradigma da linguagem C# também vale para os objetos instanciados de alguma classe, como será visto.

A **figura 1** mostra os escopos das variáveis: **contador**, **r**, **l**, **area** e **perimetro**, além da constante **PI**. As variáveis **r** e **l** são *locais* ao método **Main()**; a variável **area** é local ao método **CalcAreaCirculo()** e **perimetro** é local ao método **CalcPerimetro()**. Por outro lado, é importante frisar que **raio** e **lado** são parâmetros (não variáveis) em seus respectivos métodos. Mas, apesar do termo “global” ser universal em programação, no caso da linguagem C# ele deve ser entendido de um modo bem relativo; e algumas literaturas chegam até a afirmar que o escopo *global* não existe nesta linguagem. Na verdade, o termo “variável global” se aplica à determinada área de codificação; e no caso do C#, sendo uma linguagem orientada a objetos, os módulos podem ser entendidos como *classes*. Deste modo, uma variável acessível a toda essa classe pode ser considerada uma “variável *global*” visível em todos métodos dessa classe embora seja, na verdade, uma variável local; e a mesma conclusão o se aplica às constantes.

No esquema da **figura 1** a variável **contador** e a constante **PI** são visíveis em toda a classe **Program**; assim, todos os métodos dessa classe (**Main**, **CalcAreaCirculo** e **CalcPerimetro**) podem acessar esses dois elementos. Então, de certa forma, não está errado dizer que elas são *globais*. Por outro lado, quando se trata de criar elementos que devem ser acessados em todo o projeto (em todas os módulos e métodos) a boa prática de programação sugere que se crie uma classe independente onde esses elementos são definidos como propriedades; e nesses casos, um arquivo com extensão **.cs** é criado para conter essa classe adicionada ao projeto.

No esquema da **figura 2** pode ser observado que ao tentar exibir o conteúdo da variável **area** no método **Main()** ocorre um erro, pois esta variável é local ao método **CalcAeaCirculo()** e não acessível em nenhum outro módulo: só onde ela foi criada. A **figura 3** é uma saída do programa “PrEscopos” onde a área do círculo de raio **5** e o perímetro do quadrado de lado **6** são exibidos normalmente. Por outro lado, é importante observar os valores da variável **contador**: **1** e **2**, respectivamente: isto aconteceu devido ao fato desta variável ser *global* (acessível em toda a área da classe **Program**). Assim, sendo iniciada com **0** (zero) ela teve seu valor incrementado de uma unidade quando o método **CalcAreaCirculo()** foi executado e incrementada novamente quando foi executado o método **CalcPerimetro()**. Por isto é muito perigoso definir variáveis com escopo *global*, pois elas podem sofrer alterações indesejáveis em várias ocasiões. O programador só deve fazer uso deste tipo de escopo para variáveis quanto for muito necessário; por exemplo, em situações em que o usuário deva entrar com uma senha para acessar o sistema; nestes casos é importante ter uma variável estática com escopo global de modo que seja contado o número de tentativas para comparar com o máximo permitido; mas só em situações semelhantes, pois o ideal é SEMPRE utilizar variáveis *locais*. Por outro lado, para constantes o escopo *global* é o mais indicado, pois seu valor não pode ser alterado através de atribuições.

Nota: Postagem tirada do livro: “Linguagem C#: Com acessos a Bancos de Dados”
<https://www.amazon.com.br/Linguagem-Com-Acesso-Bancos-Dados-ebook/dp/B0BRBKNV4N>

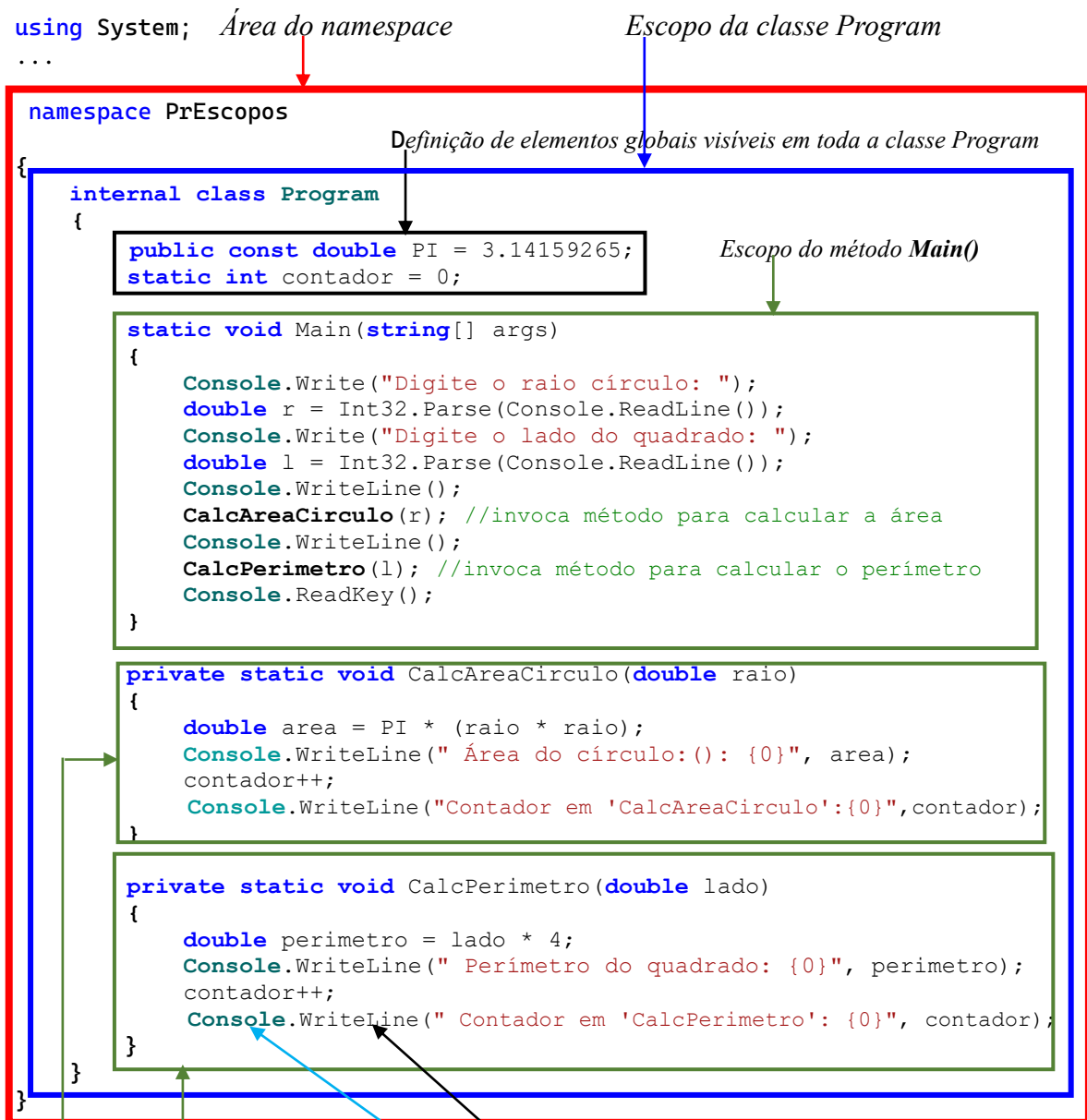


Figura 1 - Escopos dos módulos no programa “PrEscopos”

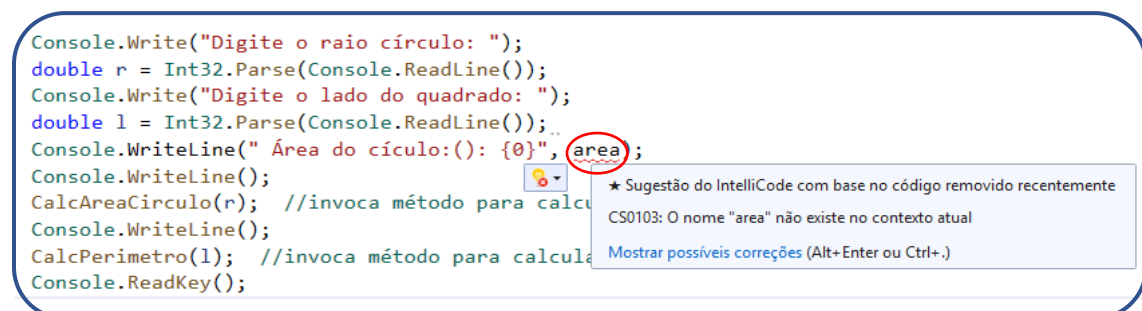
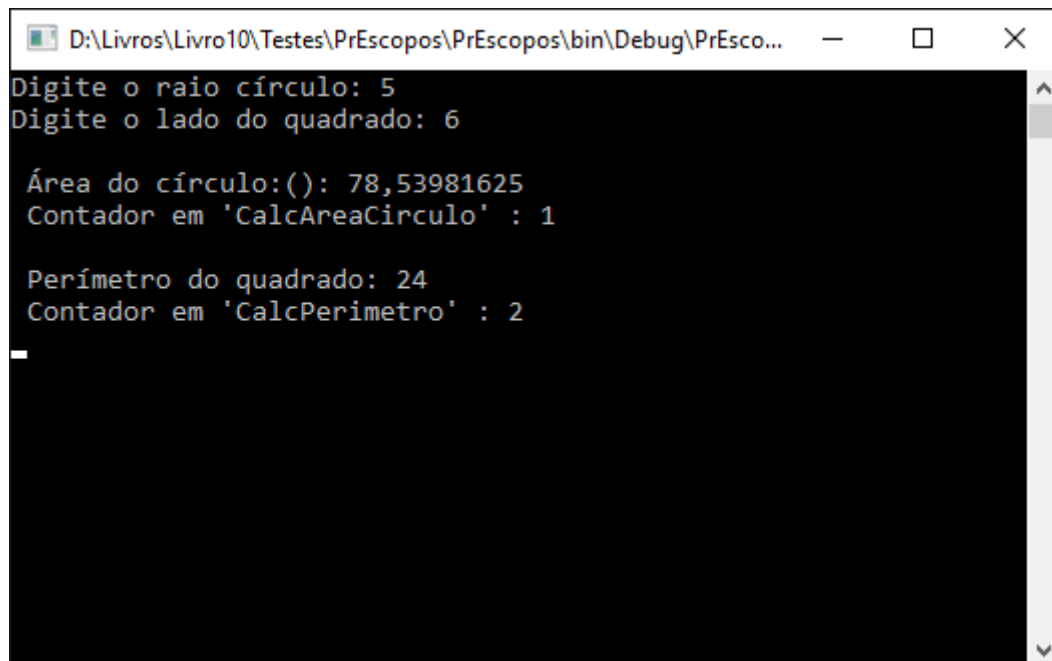


Figura 2 - Erro de acesso à uma variável



The image shows a Windows command prompt window with a black background and white text. The title bar at the top reads "D:\Livros\Livro10\Testes\PrEscopos\PrEscopos\bin\Debug\PrEsco...". The text inside the window is as follows:

```
Digite o raio círculo: 5
Digite o lado do quadrado: 6

Área do círculo(): 78,53981625
Contador em 'CalcAreaCirculo' : 1

Perímetro do quadrado: 24
Contador em 'CalcPerimetro' : 2
```

A small white cursor is visible on the line following the last output.

Figura 3 - Saída do programa “PrEscopos”