6

O Pronome MAIN

Neste capítulo, é explorado o pronome MAIN. Na Seção 6.1, o conceito de objeto principal de sistema é explicado. Na Seção 6.2, são mostradas as transformações de código necessárias para que o pronome MAIN seja corretamente executado. Na Seção 6.3, é apresentado o nível Meta necessário para a obtenção destas transformações. Finalmente, na Seção 6.4, a implementação do padrão *Singleton* [G+95], descrito na Subseção 2.4.6, utilizando o pronome MAIN é comparada com a implementação tradicional.

6.1

O Objeto Principal do Sistema

Aplicações orientadas a objetos são essencialmente mediadas [BRM+96] e um programa executável deve conter uma classe, indicada em tempo de ligação, com um método onde a execução se inicia. Uma instância desta classe é criada em tempo de inicialização do sistema e a este objeto denominamos objeto principal do sistema.

Por ser a raiz da árvore de agregação, o objeto principal consegue se comunicar com todos os objetos do sistema. Nenhum destes, no entanto, conseguem enviar mensagens ao objeto principal, a menos que referências a este sejam exportadas aos outros objetos do sistema. Esta exportação, no entanto, obriga que algum procedimento de segurança seja definido para evitar que a referência ao objeto mais importante do sistema seja vítima de algum mau uso.

A utilização do pronome MAIN substitui a referência ao objeto principal no contexto dos outros objetos do sistema.

Por existir durante toda a execução do sistema, o objeto principal é o lugar ideal para se declarar objetos *singletons* [G+95], em linguagens orientadas a objetos onde não são permitidos objetos globais. Esta estratégia só pode ser

adotada, no entanto, se algum acesso a estes objetos for disponibilizado já que estes precisam ser conhecidos por todo o sistema. Como o pronome MAIN disponibiliza o acesso ao objeto principal do sistema, basta que este disponibilize acesso aos *singletons* a ele agregados.

A Figura 6.1 mostra o cenário no nível base de duas classes onde uma utiliza o pronome MAIN. A classe *A* é declarada como principal pelo especificador <<main>> introduzido na linguagem. Já a classe *B*, envia as mensagens *m* e *n* para o objeto principal e declara que o faz através do especificador <<message>> também introduzido na linguagem.

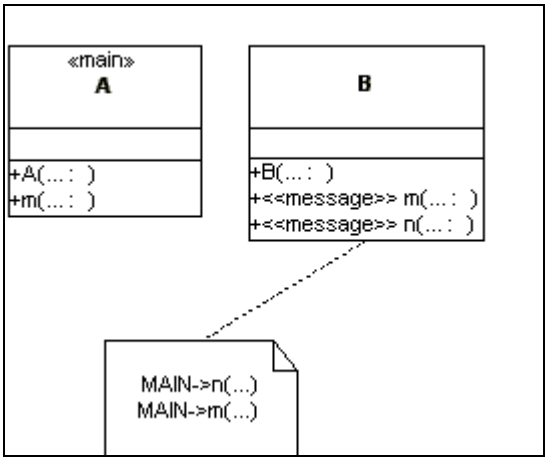

Figura 6.1 – Cenário de utilização do pronome MAIN em nível base

6.2

Transformações de Código

Este cenário deve ser transformado de modo que:

1. as classes *A* e *B* possuam uma superclasse que implemente de forma vazia um tratador para *m()* e para *n()*; e
2. o principal seja conhecido no momento em que uma mensagem a este seja enviada.

Para a primeira transformação, pode ser adotada a mesma estratégia dos pronomes descritos anteriormente.

Para a segunda transformação, é necessário mapear quando o objeto principal é determinado e torná-lo acessível a todos os objetos do sistema.

A execução do sistema se inicia em um método, determinado estaticamente como inicial, do objeto principal. A criação deste, portanto, deve se dar em um momento anterior ao início da execução do sistema, mais precisamente no momento de sua inicialização. Ou seja, quando o sistema inicia sua execução propriamente dita, o objeto principal já está determinado. Uma referência global, introduzida no sistema como suporte de tempo de execução, disponibiliza-o para os outros objetos.

Como a classe do objeto principal é conhecida, graças ao especificador <<main>>, a referência global pode ser atualizada por um comando introduzido no construtor desta classe.

A Figura 6.2 mostra o cenário transformado.

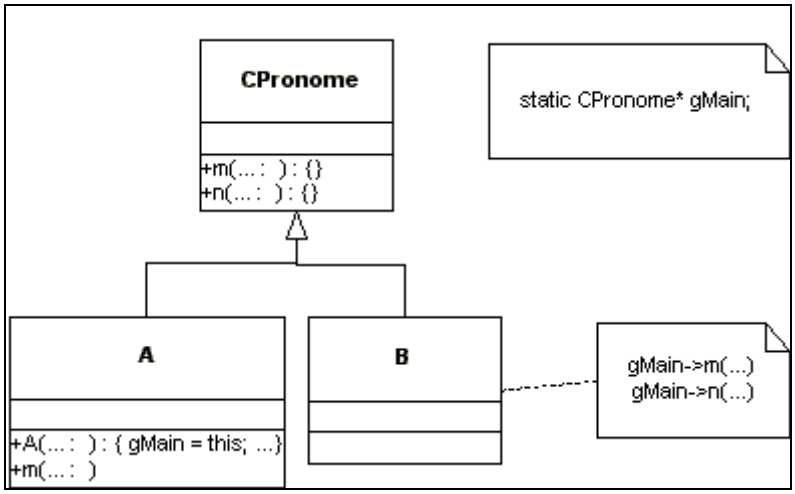

Figura 6.2 – Cenário de utilização do pronome MAIN transformado

6.3

O Nível Meta

Neste processo, existem dois tipos básicos de transformações:

- As transformações que devem ser feitas em todas as classes do sistema, que transformam as referências ao pronome MAIN em referências à referência global, *gMain*;
- As transformações que devem ser feitas apenas na classe especificada como principal que introduz em seus construtores a inicialização da referência global.

Para obter estas transformações, duas metaclasses são utilizadas. A primeira, denominada *MCReferenciaPrincipal*, é associada a todas as classes de nível base e efetiva o primeiro tipo de transformação. A segunda, denominada *MCPrincipal*, é associada à classe declarada como principal e efetua o segundo tipo de transformação. Como o primeiro tipo de transformação também precisa ser efetuado nesta classe, *MCPrincipal* herda de *MCReferenciaPrincipal*.

A classe *CPronome*, por sua vez, é associada uma metaclasse especial denominada *MCRaizPronome*. O nível meta e sua associação com o nível base mostrado anteriormente são detalhados na Figura 6.3.

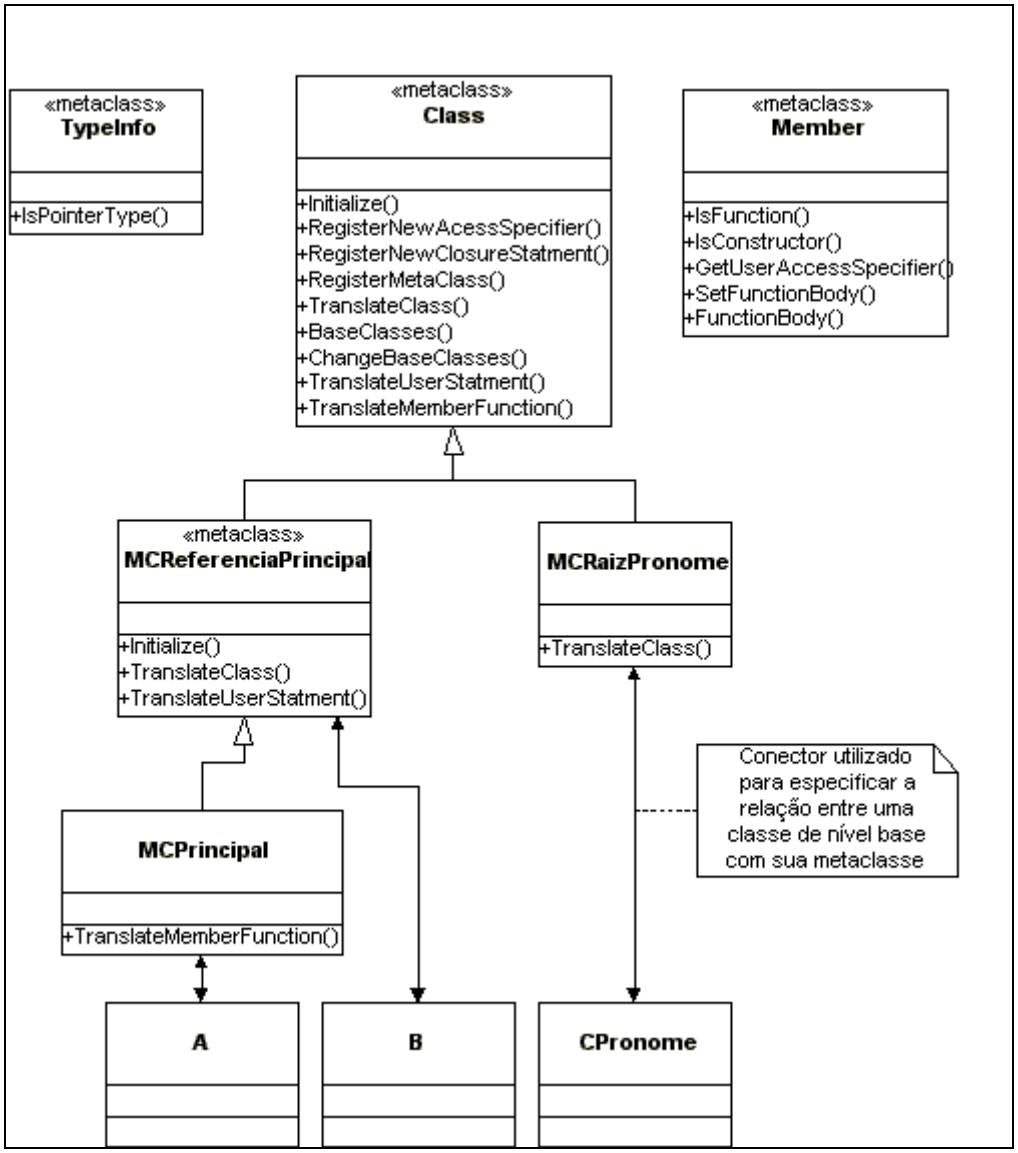

Figura 6.3 – Nível Meta para implementação do pronome MAIN

Quase todos os métodos que aparecem na Figura 6.3 já foram descritos em capítulos anteriores e não o serão novamente. Apenas um está sendo apresentado nesta transformação:

- *static void Class::RegisterMetaclass(char* keyword, char* metaclass)*: registra *keyword* como um novo modificador e associa-o a metaclasse *metaclass*;

Para que os novos especificadores (message e main) e o novo comando (envio de mensagem para o MAIN) sejam identificados, a metaclasse *MCReferenciaPrincipal* redefine o método *Initialize()* e faz os registros:

- RegisterNewAcessSpecifier(“message”)
- RegisterNewClosureStatment(“MAIN”)
- RegisterMetaclass(“MAIN”, MCPrincipal);

As inserções da nova herança e o armazenamento das mensagens enviadas através do pronome são feitos na redefinição do método *TranslateClass()*.

Para o armazenamento das mensagens enviadas através do pronome, foi utilizado o objeto *Singleton* [G+95] *RepositorioMensagens* que armazena as mensagens declaradas com o especificador <<message>> e as disponibiliza para ser introduzidas na classe *CPronome*.

A transformação do envio efetivo da mensagem através do pronome MAIN é feita redefinindo-se o método *TranslateUserStatment()*. Este substitui no comando original a palavra-chave MAIN pela referência global *gMain*.

Para que esta referência esteja se referenciando realmente ao objeto cuja classe foi declarada como principal, a metaclasse *MCPrincipal* redefine o método *TranslateClass()* e, neste, transforma o código dos construtores da classe para atualizar a referência com o objeto que está sendo criado.

Ao término da transformação da última classe associada a *MCReferenciaPrincipal*, todas as mensagens enviadas através do pronome MAIN estão armazenadas. A metaclasse *MCRaizPronome* redefine o método *TranslateClass()* e neste, insere na classe *CPronome* uma função membro para cada mensagem armazenada no repositório, sempre com implementações vazias.

6.4

O Padrão *Singleton* Utilizando MAIN

O padrão *Singleton* [G+95], descrito na Subseção 2.4.6, pode ser implementado utilizando-se o pronome MAIN. Neste caso, a estrutura do padrão propriamente dita não é modificada. A diferença se dá na forma de acesso ao objeto *singleton*. Prega o padrão que este deve ter acesso fácil e mantenha-se disponível para todo o sistema. A sugestão dada, ainda por este, é que o acesso seja através de variáveis globais. Como pode ser verificado na Figura 6.4, com o pronome MAIN, o objeto principal torna-se facilmente acessível por todo o sistema, o que o torna um forte candidato a servir de mediador dos objetos *singletons* para o resto do sistema. Assim, facilitamos o acesso a *singletons* sem que seja necessária a disponibilização de globais para isto.

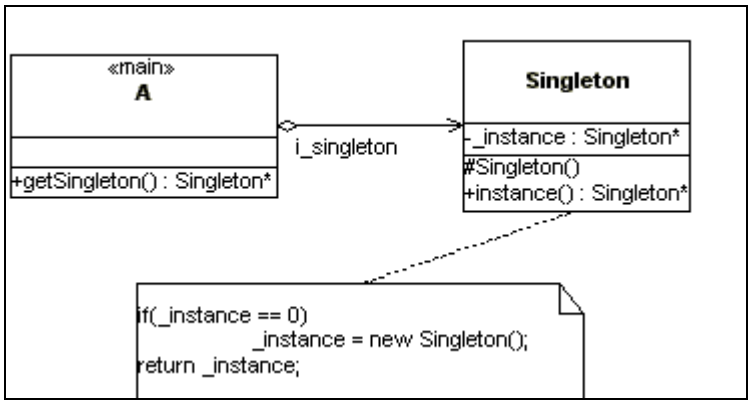

Figura 6.4 – Esquema da implementação do padrão *Singleton* utilizando MAIN

A implementação obtida após a transformação do pronome se diferencia da proposta pelo pronome pela atualização da referência global quando o *singleton* é criado e pela introdução de mais uma indireção na obtenção da referência ao *singleton*. A primeira diferença, pelo *singleton* ser criado apenas uma vez, não influi no tempo final da execução. A segunda, por sua vez, influi pois o *singleton* pode ser acessado um número indeterminado de vezes. É esta variação que foi explorada na tomada de tempos das implementações tradicional e com o pronome MAIN do padrão mostrada na Tabela 6.1 e no gráfico da Figura 6.5. Isto é, a variação se deu no número de acessos ao singleton.

Da mesma forma que nos pronomes descritos anteriormente, cada configuração do exemplo foi executada 1000 vezes, sendo os tempos mostrados

na Tabela 6.1 a média destas execuções. Os desvios padrões das amostras consideradas são apresentados na Tabela 6.2. O ambiente de execução destes experimentos foi o mesmo utilizado para os pronomes anteriores. Isto é, um computador Pentium4, com 256 MB de RAM, utilizando-se Linux Mandrake 8.1.

Número de acessos ao *Singleton*	Implementação em [G+95] (μs)	Implementação com MAIN (μs)	Diferença (μs)	Aumento Percentual
0	346	358	12	3,4682
1	355	368	13	3,6619
10	365	376	11	3,0136
100	984	1027	43	4,3699
1000	1951	2041	90	4,6130
10000	2730	2802	72	2,6373
100000	3306	3406	100	3,0248
1000000	6512	6703	191	2,9330
10000000	12036	12299	263	2,1851
100000000	14211	14737	526	3,7013
1000000000	29585	30832	1247	4,2149
10000000000	60467	63129	2662	4,4024

Tabela 6.1 – Tempos de execução do padrão *Singleton*

Número de acessos ao *Singleton*	Desvios Padrões Implementação Padrão	Desvios Padrões Implementação com MAIN
0	0,2309	0,2172
1	0,2850	0,2374
10	0,2173	0,3106
100	0,3297	0,3153
1000	0,3107	0,2139
10000	0,2853	0,1830
100000	0,1983	0,2182
1000000	0,3083	0,2309
10000000	0,4290	0,2180
100000000	0,1294	0,3183
1000000000	0,2409	0,2951
10000000000	0,2273	0,2906

Tabela 6.2 – Desvios Padrões das amostras de tempo utilizadas para análise da execução do padrão *Singleton*

Os dados apresentados na Tabela 6.1 são melhor visualizados no gráfico da Figura 6.5.

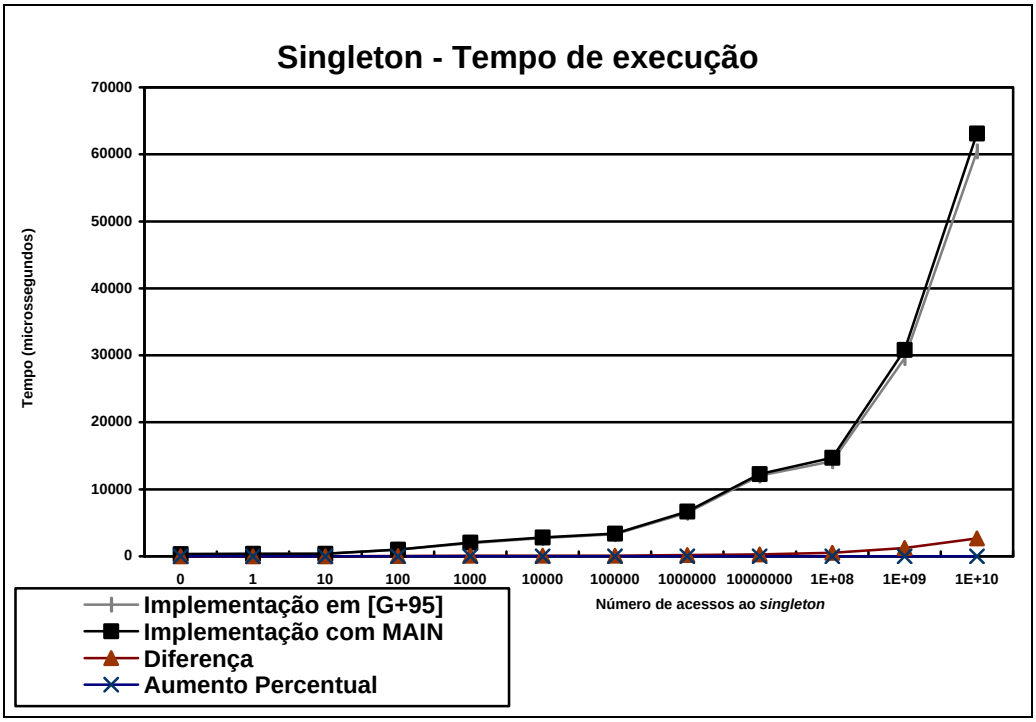

Figura 6.5 – Gráfico de comparação entre os tempos de execução do padrão *Singleton*

Na Tabela 6.1 e no gráfico mostrado na Figura 6.5 pode-se verificar que o tempo de execução na implementação do padrão utilizando-se o pronome MAIN é sempre um pouco maior, o que já era esperado. O aumento percentual máximo obtido, no entanto, foi de 4%, evidenciando a não degradação do sistema com o uso do pronome.