Anterior:Manipulando Erros, Acima:Erros [Conteúdo][Índice]

11.7.3.4 Símbolos de erro e nomes de condição

Ao sinalizar um erro, você especifica um *símbolo de erro* para especificar o tipo de erro que você tem em mente. Cada erro tem um e apenas um símbolo de erro para categorizá-lo. Esta é a melhor classificação de erros definida pela linguagem Emacs Lisp.

Essas classificações restritas são agrupadas em uma hierarquia de classes mais amplas chamadas *condições de erro* , identificadas por *nomes de condição* . As classes mais restritas pertencem aos próprios símbolos de erro: cada símbolo de erro também é um nome de condição. Existem também nomes de condição para classes mais extensas, até o nome da condição `error`que aceita todos os tipos de erros (mas não `quit`). Assim, cada erro tem um ou mais nomes de condição: `error`, o símbolo de erro se for distinto de `error`, e talvez algumas classificações intermediárias.

Função: mensagem de nome de erro de definição *e pai opcional*

> Para que um símbolo seja um símbolo de erro, ele deve ser definido com o `define-error`qual recebe uma condição pai (o padrão é `error`). Esse pai define as condições às quais esse tipo de erro pertence. O conjunto transitivo de pais sempre inclui o próprio símbolo de erro e o símbolo `error`. Como desistir não é considerado um erro, o conjunto de pais de `quit`é apenas `(quit)`.

Além de seus pais, o símbolo de erro possui uma *mensagem* que é uma string a ser impressa quando esse erro é sinalizado, mas não tratado. Se essa mensagem não for válida, a mensagem de erro '`erro peculiar`' é usado. Consulte Definição de sinal .

Internamente, o conjunto de pais é armazenado na `error-conditions` propriedade do símbolo de erro e a mensagem é armazenada na `error-message`propriedade do símbolo de erro.

Aqui está como definimos um novo símbolo de erro, `new-error`:

```
(define-error 'new-error "Um novo erro" 'my-own-errors)
```

Esse erro tem vários nomes de condição: `new-error`, a classificação mais restrita; `my-own-errors`, que imaginamos ser uma classificação mais ampla; e todas as condições das `my-own-errors`quais devem incluir `error`, que é a mais ampla de todas.

A string de erro deve começar com uma letra maiúscula, mas não deve terminar com um ponto. Isso é para consistência com o resto do Emacs.

Naturalmente, o Emacs nunca sinalizará `new-error`sozinho; apenas uma chamada explícita para `signal`(consulte Definição de sinal) em seu código pode fazer isso:

```
(sinal 'novo-erro' (xy))
     erro→ Um novo erro: x, y
```

Esse erro pode ser tratado por meio de qualquer um de seus nomes de condição. Este exemplo trata `new-error`e quaisquer outros erros na classe `my-own-errors`:

```
(condição-caso foo
    (barra zero t)
  (meu-próprio-erros nil))
```

A maneira significativa pela qual os erros são classificados é por seus nomes de condição - os nomes usados para corresponder erros com manipuladores. Um símbolo de erro serve apenas como uma maneira conveniente de especificar a mensagem de erro pretendida e a lista de nomes de condições. Seria complicado fornecer `signal`uma lista de nomes de condições em vez de um símbolo de erro.

Por outro lado, usar apenas símbolos de erro sem nomes de condição diminuiria seriamente o poder de `condition-case`. Os nomes de condição possibilitam categorizar erros em vários níveis de generalidade quando você escreve um manipulador de erros. O uso de símbolos de erro por si só eliminaria tudo, exceto o nível mais estreito de classificação.

Consulte Erros padrão , para obter uma lista dos principais símbolos de erro e suas condições.

Anterior:Manipulando Erros, Acima:Erros [Conteúdo][Índice]