

exceção, essa exceção substitui qualquer exceção que estivesse no processo de ser jogado. Se uma cláusula finalmente emitir uma declaração de devolução, o método retorna normalmente, mesmo que uma exceção tenha sido lançada e tenha ainda não foi tratado.

Tente e finalmente pode ser usado juntos sem uma cláusula de captura. Em Este caso, o bloco finalmente é simplesmente o código de limpeza que é garantido para ser executado, independentemente do que acontecer na tentativa bloquear. Lembre -se de que não podemos simular completamente um loop para um com um enquanto loop porque a declaração continua se comporta de maneira diferente para os dois loops. Se adicionarmos uma declaração de tentativa/finalmente, podemos escrever um enquanto loop que funciona como um loop e que as alças continuam declarações corretamente:

```
// simular para (inicializar; teste; incremento) corpo;
inicializar;
while (teste) {
  tente {body;}
  finalmente {incremento;}
}
```

Observe, no entanto, que um corpo que contém uma declaração de quebra se comporta um pouco diferente (causando um incremento extra antes de sair) no enquanto o loop do que faz no loop for, mesmo com o finalmente cláusula, não é possível simular completamente o loop for com enquanto.

Cláusulas de captura nua

Ocasionalmente, você pode se encontrar usando uma cláusula de captura apenas para detectar e parar a propagação de

Uma exceção, mesmo que você não se importe com o tipo ou o valor da exceção. Em ES2019

E mais tarde, você pode omitir os parênteses e o identificador e usar a palavra -chave de captura nua neste caso. Aqui está um exemplo: