

#### Tópicos Abordados

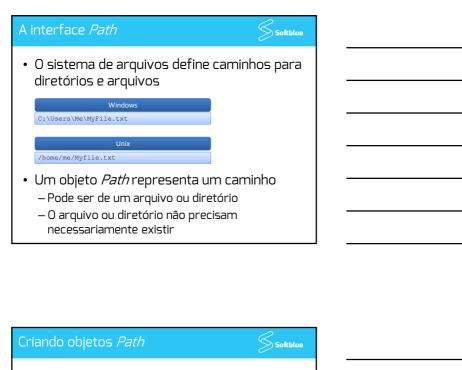


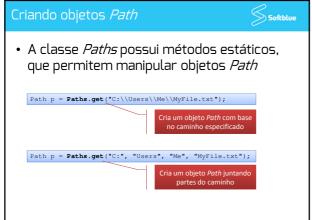
- NIO.2 API x Legacy I/O API
- A interface Path
  - Manipulando objetos *Path*
  - Métodos úteis da classe *Paths*
- A classe *Files*
- Operações em arquivos e diretórios
  - Criar, excluir, copiar, mover, etc.
- Lendo e escrevendo dados
- Iterando sobre elementos de um diretório
  - Filtros glob
  - Filtros customizados
  - A interface *FileVisitor*
- Watch Service API
- Integração entre NIO.2 e Legacy I/O APIs

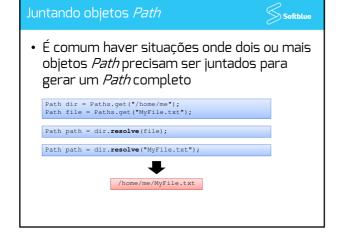
#### NIO.2 API x Legacy I/O API

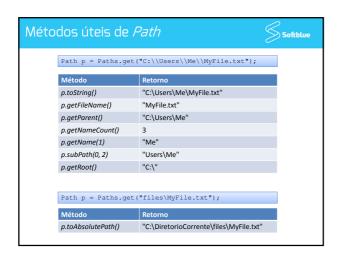


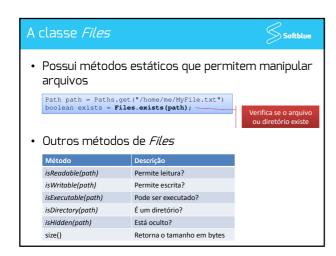
- O Java conta com uma API de I/O desde a sua primeira versão
  - Esta API tem algumas limitações
- A New I/O (NIO.2) API surgiu para resolver os problemas que a API legada possui
- A NIO.2 API surgiu com o Java 7
- A tendência é que ela substitua a API de I/O legada gradualmente







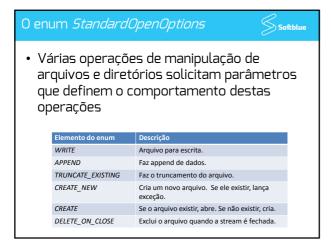






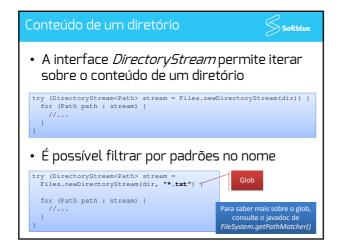












# 

## Iterando sobre a árvore de diretórios



- A interface *FileVisitor* possui 4 métodos que devem ser implementados
  - preVisitDirectory()
  - postVisitDirectory()
  - visitFile()
  - visitFileFailed()
- É possível também estender a classe *SimpleFileVisitor*
- O método *Files.walkFileTree()* é chamado para iniciar a iteração

### Watch Service API



 Permite que a aplicação seja notificada quando um diretório ou arquivo é criado, excluído ou modificado

• Deve existir um loop na aplicação que fica recebendo os eventos

# Integrando NIO.2 com a Legacy I/O API Softblue • Aplicações antigas (anteriores ao Java 7)

- não tiveram acesso à NIO.2 API
- Para que estas aplicações não precisem ser reescritas, existe uma forma de integrar a API legada de I/O com a NIO.2 API



