

# NIO.2 уже в JDK7!

Александр Последов  
Sun Microsystems

# Всё это и даже больше уже есть в JDK7, настало время NIO.2!

- Работа с расширенными атрибутами файлов
- Уведомления об изменениях файлов
- Копирование файлов средствами файловой системы
- Асинхронный ввод/вывод
- Поддержка MIME информации
- Работа с символьными ссылками
- Поддержка новых файловых систем

# Всё это и даже больше уже есть в JDK7, настало время NIO.2!

- Работа с расширенными атрибутами файлов
- Уведомления об изменениях файлов
- Копирование файлов средствами файловой системы
- Асинхронный ввод/вывод
- Поддержка MIME информации
- Работа с символьными ссылками
- Поддержка новых файловых систем

# Что нового в работе с файлами?

- Знакомьтесь - `java.nio.file.Path`

# Что нового в работе с файлами?

- Знакомьтесь - `java.nio.file.Path`
- Старое по-новому: `java.io.File`

# Что нового в работе с файлами?

- Знакомьтесь - `java.nio.file.Path`
- Старое по-новому: `java.io.File`
- Директории – это удобно!

# Что нового в работе с файлами?

- Знакомьтесь - `java.nio.file.Path`
- Старое по-новому: `java.io.File`
- Директории – это удобно!
- Рекурсивные операции

# Что нового в работе с файлами?

- Знакомьтесь - `java.nio.file.Path`
- Старое по-новому: `java.io.File`
- Директории – это удобно!
- Рекурсивные операции
- Уведомления об изменениях в файлах



# Что нового в работе с файлами?

- Знакомьтесь - `java.nio.file.Path`
- Старое по-новому: `java.io.File`
- Директории – это удобно!
- Рекурсивные операции
- Уведомления об изменениях в файлах
- Атрибуты файлов теперь доступны!

# Основа новой концепции

- **FileRef**

- простейший дескриптор файла

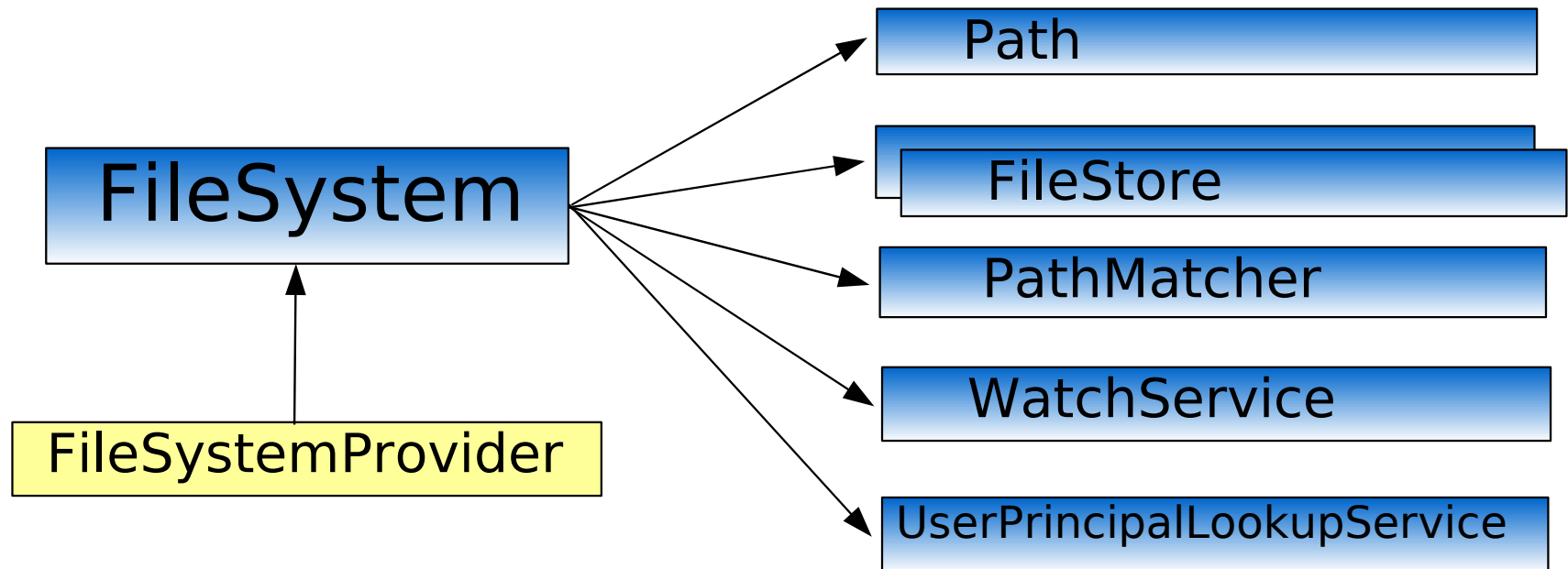
- **Path**

- содержит данные, зависящие от конкретной файловой системы

- **FileSystem**

- интерфейс к файловой системе платформы
  - файловая система по умолчанию для локальной платформы

# Структура FileSystem API



# Возможности класса Path

- **Доступ к компонентам пути**
  - *getName, getParent, getRoot, subpath*

# Возможности класса Path

- **Доступ к компонентам пути**
  - *getName, getParent, getRoot, subpath*
- **Сравнение и проверка путей**
  - *startsWith, endsWith, equals, compareTo*

# Возможности класса Path

- **Доступ к компонентам пути**
  - *getName, getParent, getRoot, subpath*
- **Сравнение и проверка путей**
  - *startsWith, endsWith, equals, compareTo*
- **Преобразование путей**
  - *resolve* (и наоборот: *relativize*)

# File: НОВЫЕ ВОЗМОЖНОСТИ

- **Операции I/O:**

- *newInputStream/newOutputStream* для stream I/O
- *newSeekableByteChannel* для channel I/O
- приведение к *FileChannel* сделало возможным блокировку файлов, отображение файлов в памяти и т.д..

- **Другие операции:**

- *createFile, createDirectory, delete, copyTo, moveTo, isSameFile, toRealPath, checkAccess,...*

# Работа с директориями

- **DirectoryStream для рекурсивного обхода**
  - Работает с очень большими директориями
  - Требуется меньше ресурсов
  - Лучше работает с удалёнными файловыми системами
- **Фильтрация в процессе обхода**



# Поддержка атрибутов

- Мета-данные ассоциированные с файлом
- Чего не хватало приложениям?...

# Поддержка атрибутов

- Мета-данные ассоциированные с файлом
- Чего не хватало приложениям?...
  - работа с владельцами/группами
  - работа с правами доступа
  - временные метки
  - DOS атрибуты
  - расширенные атрибуты
  - ...

# Поддержка атрибутов

- Мета-данные ассоциированные с файлом
- Чего не хватало приложениям?...
- Зависимость от файловой системы платформы

# Поддержка атрибутов

- Новый пакет: *java.nio.file.attribute*

# Поддержка атрибутов

- Новый пакет: *java.nio.file.attribute*
- Основные атрибуты: *BasicFileAttributeView*

# Поддержка атрибутов

- Новый пакет: *java.nio.file.attribute*
- Основные атрибуты: *BasicFileAttributeView*
- POSIX атрибуты: *PosixFileAttributeView*

# Поддержка атрибутов

- Новый пакет: *java.nio.file.attribute*
- Основные атрибуты: *BasicFileAttributeView*
- POSIX атрибуты: *PosixFileAttributeView*
- ACL : *AclFileAttributeView*
  - Основан на NFSv4 ACL модели

# Поддержка атрибутов

- Новый пакет: *java.nio.file.attribute*
- Основные атрибуты: *BasicFileAttributeView*
- POSIX атрибуты: *PosixFileAttributeView*
- ACL : *AclFileAttributeView*
  - Основан на NFSv4 ACL модели
- *FileOwnerAttributeView*



# Поддержка атрибутов

- Новый пакет: *java.nio.file.attribute*
- Основные атрибуты: *BasicFileAttributeView*
- POSIX атрибуты: *PosixFileAttributeView*
- ACL : *AclFileAttributeView*
  - Основан на NFSv4 ACL модели
- *FileOwnerAttributeView*
- Ваш собственный  
*UserDefinedFileAttributeView!*

# WatchService API

- Регистрация объекта файловой системы используя WatchKey
- В случае изменений в файле, WatchKey сигнализирует об этом в WatchService
- WatchService содержит методы принудительного опроса об изменениях
- Реакция на событие
- Сброс в начальное состояние (ready)

# Ссылки

- Проект New I/O на сайте OpenJDK
  - <http://openjdk.java.net/projects/nio>
- Блог Алана Бэйтмана
  - <http://blogs.sun.com/alanb>

# Вопросы и ответы