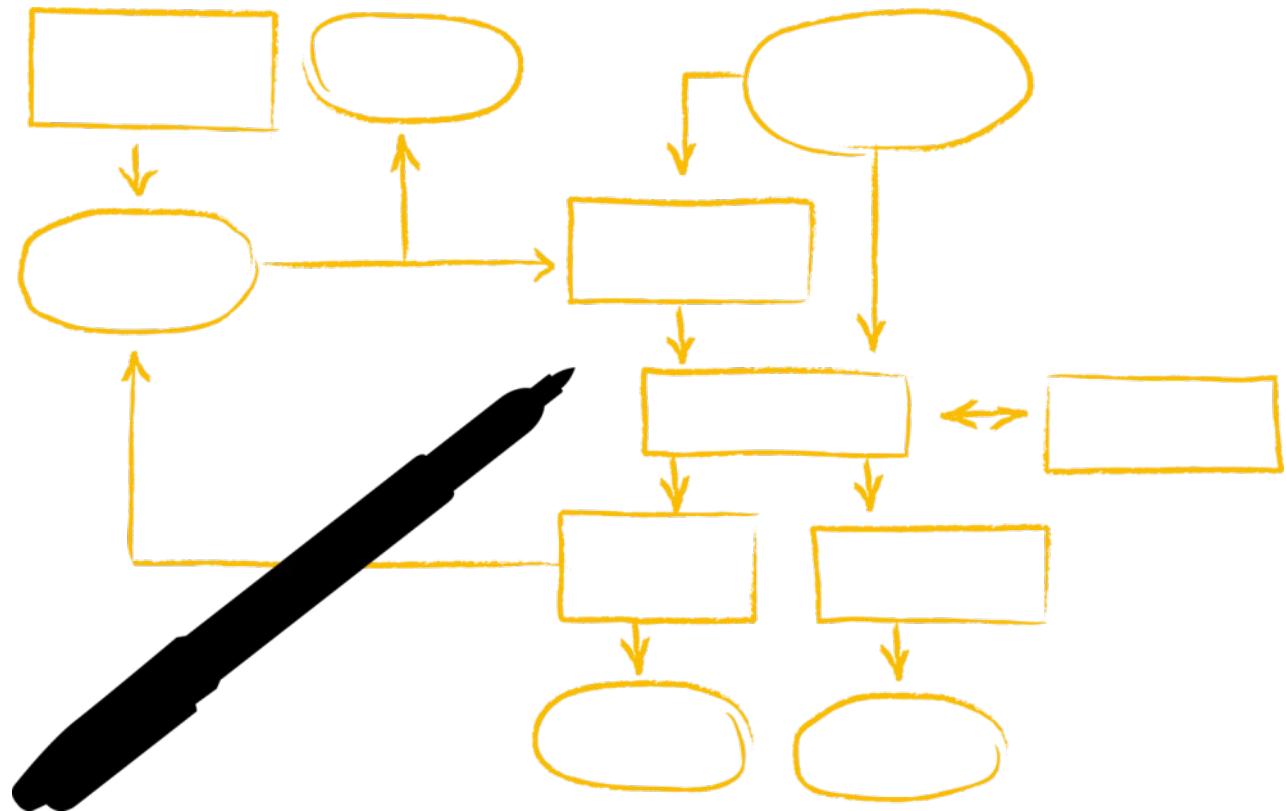


# Файлов вход/изход в Java 7

Владимир Панов  
Ноември, 2014

Public



# Съдържание

---

**Операции с файловите метаданни**

**Операции с файл като едно цяло**

**Произволен (непоследователен) достъп до файловите данни**

# Пътища

---

## Файлова система

- Йерархична структура
- Състои се от директории и файлове
- Главна директория (корен)
- Текуща (работна) директория

## Пътища

- Относителни
- Абсолютни

## Символни връзки

# java.nio.file.Path

## Създаване

---

Класът **java.nio.file.Path** служи за представяне на път.

- Може да сочи към файл или директория.
- Може да сочи към съществуващ или несъществуващ обект.

Създаване – чрез **java.nio.file.Paths**:

```
Path absolutePath = Paths.get("/home", "joe", "file.txt");  
  
Path absolutePath = Paths.get("c:\\Users\\joe", "file.txt");  
  
Path relativePath = Paths.get("documents", "FileIO.odp");  
  
Path relativePath = Paths.get("documents/FileIO.odp");
```

# java.nio.file.Path

## Извличане на информация

Метод	... връща под UNIX	... връща под Windows
<code>toString()</code>	/home/joe/foo	C:\home\joe\foo
<code>getFileName()</code>	foo	foo
<code>getName(0)</code>	home	home
<code>getNameCount()</code>	3	3
<code>subpath(0, 2)</code>	home/joe	home\joe
<code>getParent()</code>	/home/joe	\home\joe
<code>getRoot</code>	/	C:\

# java.nio.file.Path

## Преобразувания

**Path toAbsolutePath()** - преобразува пътя до абсолютен (започващ от главната директория).

**Path toRealPath()** - преобразува пътя до реален:

- абсолютен
- с премахнати символни връзки (заменени от своите цели)
- с премахнати . и ..

**Path resolve(Path anotherPath)** – добавя относителен път:

```
Path p1 = Paths.get("/home/joe/foo");
Path p2 = p1.resolve("bar"); // резултатът е /home/joe/foo/bar
```

```
Path p3 = Paths.get("whatever");
Path p4 = p3.resolve("/home/joe"); // резултатът е /home/joe
```

# **java.nio.file.Path**

## Сравняване

---

**boolean equals(Path anotherPath)**

**boolean startsWith(Path anotherPath)**

**boolean endsWith(Path anotherPath)**

# java.nio.file.Files

## Метаданни

---

`long size(Path path)`

`boolean isRegularFile(Path path)`

`boolean isDirectory(Path path)`

`boolean isSymbolicLink(Path path)`

`boolean isHidden(Path path)`

`FileTime getLastModifiedTime(Path path)`

`FileTime setLastModifiedTime(Path path, FileTime time)`

`UserPrincipal getOwner(Path path)`

`Object getAttribute(Path path, String attribute)`

`Path setAttribute(Path path, String attribute, Object value)`

# java.nio.file.Files

## Други проверки

---

`boolean exists(Path path)`

`boolean notExists(Path path)`

`boolean isReadable(Path path)`

`boolean isWritable(Path path)`

`boolean isExecutable(Path path)`

`boolean isSameFile(Path path1, Path path2)`

# java.nio.file.Files

## Изтриване

---

`void delete(Path path)`

- ако `path` не съществува, изхвърля `NoSuchFileException`
- ако `path` е непразна директория, изхвърля `DirectoryNotEmptyException`

`boolean deleteIfExists(Path path)`

- ако `path` не съществува, просто връща `false`
- ако `path` е непразна директория, изхвърля `DirectoryNotEmptyException`

# java.nio.file.Files

## Копиране

---

`Path copy(Path source, Path target, CopyOption... options)`

- Съдържанието на директория не се копира
- `options` може да бъде една или повече от:
  - `StandardCopyOption.COPY_ATTRIBUTES`: копират се и атрибутите на файла/директорията, напр. време на последна промяна
  - `StandardCopyOption.REPLACE_EXISTING`: ако `target` е съществуващ файл, то той се надписва вместо да се хвърля изключение

# java.nio.file.Files

## Преместване (преименуване)

```
Path move(Path source, Path target, CopyOption... options)
```

- директория може да се премести само на същата файлова система
- `options` може да бъде една или повече от:
  - `StandardCopyOption.ATOMIC_MOVE`: преместването е атомарно
  - `StandardCopyOption.REPLACE_EXISTING`: ако `target` е съществуващ файл, то той се надписва вместо да се хвърля изключение

# **java.nio.file.Files**

## Създаване на директория

---

**Path createDirectory (Path dir)** – създава директория с път **dir**, ако родителската директория на този път вече съществува

**Path createDirectories (Path dir)** - създава директория с път **dir**, създавайки и родителските директории, ако има нужда

# java.nio.file.Files

## Обхождане на директория

`DirectoryStream<Path> newDirectoryStream(Path dir)` – създава поток, от който могат да бъдат прочетени файловете и поддиректориите на директорията с път `dir` като инстанции на `Path`

- `DirectoryStream` прилика повече на колекция (защото е наследник на `Iterable`), отколкото на входно-изходен поток.

```
Path dir = ...;
try (DirectoryStream<Path> stream =
      Files.newDirectoryStream(dir)) {
    for (Path fileOrSubDir: stream) {
        // употреба на fileOrSubDir
    }
} catch (IOException | DirectoryIteratorException x) {
    // обработка на изключенията
}
```

# java.nio.file.Files

## Обхождане на директория с глоб (жокери, wild cards)

---

`DirectoryStream<Path> newDirectoryStream(Path dir, String glob)` – създава поток, който съдържа само файловете и поддиректориите, които отговарят на глоба `glob`

# java.nio.file.Files

## Глоб

---

\* - замества произволен брой символи (вкл. николко) в рамките на едно име от пътя (т.е. не може да замести пътния разделител - / или \)

\*\* - като \*, но пресича границите на имената в пътя

? - замества точно един символ

{ subGlob1, ..., subGlobN} – замества някой от подглобовете

[c<sub>1</sub>...c<sub>N</sub>] – замества някой от символите

[c<sub>1</sub>-c<sub>2</sub>] – замества някой от символите в диапазона от c<sub>1</sub> до c<sub>2</sub>

\ - използва се за исkeleyпване на специалните символи, вкл. \

всеки друг символ - замества себе си

# **java.nio.file.Files**

## Примерни глобове

---

`/home/*/*file` - замества `/home/joe/file` (и други), но не и  
`/home/joe/work/file`

`/home/**/*file` - замества `/home/joe/work/file` (и други)

`{temp*, tmp*}` - замества всички имена, започващи с `temp` или `tmp`

`[fmi]` – замества една от буквите `f`, `m`, и `i`

`[a-z]` – замества една от малките латински букви

`[fF]` – замества малко или голямо `F`

`\ \` - замества обратно наклонена черта

`alabala` – замества `alabala`

`[[]]` - замества `[; ; ]` всички специални символи с изключение на `,`, `-` и `\`  
може да не се исkeleyват

# java.nio.file.Files

## Временни файлове и директории

---

`Path createTempFile(Path dir, String prefix, String suffix)` – създава нов файл в директорията `dir` и с име, започващо с `prefix` и завършващо със `suffix`

`Path createTempFile(String prefix, String suffix)` – създава нов файл в системната временна директория и с име, започващо с `prefix` и завършващо със `suffix`

`Path createTempDirectory(Path dir, String prefix)` – създава нова поддиректория в директорията `dir` и с име, започващо с `prefix`

`Path createTempDirectory(String prefix)` – създава нова поддиректория в системната временна директория и с име, започващо с `prefix`

# java.nio.file.Files

## Файлове с произволен достъп - отваряне

Позволяват непоследователен (произволен) достъп до файловото съдържание (за разлика от потоците).

```
SeekableByteChannel newByteChannel(Path file, OpenOption...  
options)
```

- `options` може да бъде една или повече от:
  - `StandardOpenOption.WRITE`
  - `StandardOpenOption.APPEND`
  - `StandardOpenOption.TRUNCATE_EXISTING`
  - `StandardOpenOption.CREATE_NEW`
  - `StandardOpenOption.CREATE`
  - `StandardOpenOption.DELETE_ON_CLOSE`
  - `StandardOpenOption.SYNC`

# **java.nio.channels.SeekableByteChannel**

## **Файлове с произволен достъп**

---

**long position()** - връща текущата позиция на канала

**SeekableByteChannel position(long newPosition)** - променя текущата позиция на канала на **newPosition**

**int read(ByteBuffer dst)** - прочита данни от текущата позиция на канала в **dst**

**int write(ByteBuffer src)** - записва данни от **src** на текущата позиция в канал

**SeekableByteChannel truncate(long size)** - отрязва канала до размер **size**

# java.nio.file.Files

## Обработване на съдържанието на файл наведнъж

---

Следните методи са подходящи за малки файлове:

```
byte[] readAllBytes(Path file)
```

```
List<String> readAllLines(Path file, Charset cs)
```

```
Path write(Path file, byte[] bytes, OpenOption... options)
```

```
Path write(Path path, Iterable<? extends CharSequence> lines,  
Charset cs, OpenOption... options)
```

# Допълнителни материали

---

Java tutorial за файлов вход/изход:

<https://docs.oracle.com/javase/tutorial/essential/io/fileio.html>

... или просто потърсете следните ключови думи:

Java tutorial file io



# Благодаря за вниманието!