**Типы путей**

Рассмотрим эти имена файлов:

C:\temp\file.txt - это путь, абсолютный путь ( *getAbsolutePath()* ) и канонический путь ( *getCanonicalPath()* ).

.\file.txt - это путь ( *getPath()* ). Это не абсолютный путь и не канонический путь.

C:\temp\myapp\bin\..\\..\file.txt - это путь и абсолютный путь. Это не канонический путь.

Канонический путь-это всегда абсолютный путь.

*getContextPath()* от слова Context, это путь контекста, то есть путь к проекту.

*getRealPath()* получает реальный путь, который является абсолютным путем, с буквенным именем диска.

*getAbosolutePath()* совпадает с *getRealPath()*, но объект, вызывающий метод, отличается.

Преобразование из пути в канонический путь делает его абсолютным (обычно привязка к текущему рабочему каталогу, например, ./file.txt становится c:/temp/file.txt ). Канонический путь файла просто "purifies" путь, удаляющий и разрешающий такие вещи, как ..\ и разрешающие символьные ссылки (на unixes).

Также обратите внимание на следующий пример с nio.Paths:

String canonical\_path\_string = "C:\\Windows\\System32\\";

String absolute\_path\_string = "C:\\Windows\\System32\\drivers\\..\\";

System.out.println(Paths.get(canonical\_path\_string).getParent());

System.out.println(Paths.get(absolute\_path\_string).getParent());

Хотя оба пути относятся к одному и тому же местоположению, выходные данные будут совершенно разными:

C:\Windows

C:\Windows\System32\drivers

Поскольку getCanonicalPath () читает файловую систему, это приводит к снижению производительности. Если мы уверены, что нет избыточных имен или символических ссылок, а регистр букв диска стандартизирован (если используется ОС Windows), то мы должны предпочесть использование getAbsoultePath ()