

Files, Path

Java Collections

1 уровень, 3 лекция

ОТКРЫТА

— О, Риша, здорово!

— Привет, Амиго! Как жизнь?

— Отлично. Только что Билабо рассказывал кучу интересного про File, и как с ним работать.

— Думаю, у меня как раз есть что добавить по этой теме.

— Да? Тогда с удовольствием послушаю.

— Тогда слушай. Java постоянно развивается, постоянно ищутся новые способы делать различные вещи эффективнее. Еще в Java 7 была добавлена альтернатива классу File.



— Альтернатива?

— Ага. Взяли за основу класс **File**, добавили в него немного нового, переименовывали методы, а в конце еще и разделили на два. Так что теперь есть два новых класса – **Path** и **Files**. **Path** – это, фактически новый аналог класса **File**, а **Files** – это утилитный класс (по аналогии с классами **Arrays & Collections**), в него вынести все статические методы класса **File**. Так «правильнее» с точки зрения ООП.

— Ну, раз с точки зрения ООП, то – ок. А что поменялось-то?

— Во-первых, отказались от дублирования методов, которые возвращали **String** и **File**. В классе **Path** все методы возвращают **Path**.

Во-вторых, убрали многие статические утилитные методы в класс **Files**.

В третьих, удобнее стало работать с относительными путями.

Вот список методов:

НАЧАТЬ ОБУЧЕНИЕ

<code>boolean isAbsolute()</code>	Возвращает true, если путь – абсолютный.
<code>Path getRoot()</code>	Возвращает корень текущего пути – директорию самого верхнего уровня.
<code>Path getFileName()</code>	Возвращает имя файла из текущего пути.
<code>Path getParent()</code>	Возвращает директорию из текущего пути.
<code>boolean startsWith(Path other)</code>	Проверяет, что текущий путь начинается с переданного пути.
<code>boolean endsWith(Path other)</code>	Проверяет, что текущий путь заканчивается на переданный путь.
<code>Path normalize()</code>	Нормализует текущий путь. Например, приводит путь «c:/dir/dir2/../a.txt» к пути «c:/dir/a.txt»
<code>Path relativize(Path other)</code>	Вычисляет относительный путь двух путей – «разницу путей»
<code>Path resolve(String other)</code>	Восстанавливает абсолютный путь по текущему и относительному.
<code>URI toUri()</code>	Возвращает URI текущего пути/файла.
<code>Path toAbsolutePath()</code>	Приводит путь к абсолютному, если был относительный.
<code>File toFile()</code>	Возвращает объект File, который соответствует текущему объекту Path.

— А текущий путь – это что?

— Это тот путь, который был передан в конструктор объекта Path, методы которого вызываются. Path – это «путь» по-английски.

— Ок. А какие методы есть у класса **Files**?

— Ну и куда же ты так спешишь-то? Сейчас все расскажу. Вот основные методы:

Методы класса Files	Описание
<code>Path createFile(...)</code>	Создает файл на диске.
<code>Path createDirectory(...)</code>	Создает директорию.
<code>Path createDirectories(...)</code>	Создает директорию и поддиректории.
<code>Path createTempFile(...)</code>	Создает «временный файл»
<code>Path createTempDirectory(...)</code>	Создает «временную директорию»
<code>void delete(Path path)</code>	Удаляет файл/директорию.
<code>Path copy(Path source, Path target,...)</code>	Копирует файл.
<code>Path move(Path source, Path target, ...)</code>	Перемещает файл

НАЧАТЬ ОБУЧЕНИЕ

<code>boolean isDirectory(Path)</code>	Путь — это директория?
<code>boolean isRegularFile(Path)</code>	Путь – это файл?
<code>long size(Path)</code>	Возвращает размер файла.
<code>boolean exists(Path)</code>	Объект с таким именем существует?
<code>boolean notExists(Path)</code>	Объект с таким именем не существует?
<code>long copy(InputStream, OutputStream)</code>	Копирует байты из InputStream в OutputStream.
<code>long copy(Path, OutputStream)</code>	Копирует все байты из Path в OutputStream.
<code>long copy(InputStream, Path)</code>	Копирует все байты из InputStream в Path.
<code>byte[] read(InputStream, int initialSize)</code>	Читает массив байт из InputStream.
<code>byte[] readAllBytes(Path path)</code>	Читает все байты из InputStream.
<code>List<String> readAllLines(Path path,...)</code>	Читает текстовый файл, возвращает список строк.
<code>Path write(Path path, byte[] bytes,...)</code>	Пишет массив байт в файл.

— Как интересно, столько крутых функций, и все в одном месте.

А зачем нужны временные файлы?

— Ну, смотри. Допустим, ты хочешь скачать файл из интернета, а потом отправить его кому-то. Для этого очень удобно создать временный файл на диске и сохранять считанные данные в него.

— А сложно скачать файл из интернета?

— Очень просто. Смотри пример:

Код	
1	<code>URL url = new URL("https://www.google.com.ua/images/srpr/logo11w.png");</code>
2	<code>InputStream inputStream = url.openStream();</code>
3	
4	<code>Path tempFile = Files.createTempFile("temp-", ".tmp");</code>
5	<code>Files.copy(inputStream, tempFile);</code>

— И это все?

— Да, а что ты ожидал тут увидеть? Тут всего 4 строчки.

Строка номер 1. Создается объект URL, куда передается ссылка в интернете на файл с картинкой.

Строка номер 2. У объекта url открываться поток на чтение файла – InputStream.

Строка номер 4. С помощью метода `createTempFile` создается временный файл.

Строка номер 5. Метод Files.copy копирует данные из `inputStream` в `tempFile`. Все.

— Гениально!

— Отлично, рад, что тебе понравилось. Думаю, с остальными методами ты разберёшься сам. А я попрошу Диего дать тебе

НАЧАТЬ ОБУЧЕНИЕ

Вот тебе, кстати, хорошая ссылка на этот материал:

[Ссылка на материал](#)

[← Предыдущая лекция](#)

 ×21


 +130 

Комментарии (104)

популярные новые старые

JavaCoder

Введите текст комментария


Just me Уровень 41, Гомель 1 апреля, 14:31 

Долго не понимал, почему для файла и папки используется один класс - File или Path. Сейчас понял: мы, люди, привыкли, что файлы и папки - это разные сущности. Как банка сгущенки(файл) и коробка(папка). Но по сути, что банка сгущенки, что коробка - это емкости. А вот сама сгущенка - это и есть то, что мы привыкли называть файлом - набор данных.

Память устройства - это огромное количество ячеек, где в каждую ячейку "записывается" 0 либо 1. Но наши данные - это огромные последовательности таких 1 и 0. Например, число типа int займет 4 байта, или 32 бита. И чтобы записать число типа int, например, 5, нам потребуется целых 32 ячейки памяти! Но нам же еще надо прочитать это число, а именно эту последовательность единиц и нулей. Для этого нам надо знать место на диске, где располагается эта последовательность. Так вот это место на диске - это и есть тот самый путь, который мы обычно указываем. Т.е. папка - это ящик, а название файла - это банка. Но это все по сути - просто ссылка на некую область памяти.

Но здесь есть один момент. Нам надо понять, что это за последовательность единиц и нулей лежит перед нами - музыка, фильм, картинка чтобы знать, какой программой её прочитать. Для этого мы помечаем каждую такую последовательность путем добавления дополнительных маркерных единиц и нулей в её начало. Этот маркер и есть наше расширение файла, которой мы привыкли видеть.

Поэтому любой файл на компьютере - это ссылка на область памяти, в которой лежат данные, а не сами данные.

Ответить  +7 

KOTNinja Уровень 32, Санкт-Петербург, Россия 10 марта, 16:23 

Я один дико бомблю с контраста между лекциями и задачами? Лекция: **"Это вот цифры, дружок, с помощью них ведут подсчет! Вот так-то ;)"** Задачи: **"Ты уже знаешь что такое цифры, так что вот тебе, дружок, задачка - найди ка все числа Армстронга и быстренько давай."**

Ответить  +19 

comrade_b Уровень 32, Амстердам, Нидерланды 13 июля, 13:23 

ты не один!

Ответить  0 

Orion Уровень 31, Санкт-Петербург, Россия 22 июня 2021, 19:17 

Вот я чего не понимаю- Path ведь не класс, а интерфейс. Тогда объект чего создаётся тут Path newPath = Paths.get("bla/bla/bla");?
Тут где-то анонимный класс или как?
У меня после короны вообще соображать не получается...

Ответить  0 



Евгений Ведущий инженер в ПАО Сбербанк EXPERT 26 июня 2021, 22:00 

Глянь в исходники.
Paths.get вызывает вот такой метод:

```
1 public static Path of(String first, String... more) {
2     return FileSystems.getDefault().getPath(first, more);
3 }
```

Там где-то реализация интерфейса Path возвращается.

Ответить  0 

НАЧАТЬ ОБУЧЕНИЕ

Это и есть инкапсуляция и удобство интерфейсов

Ответить

− +3 +

BogarD Уровень 35, Kherson, Украина

23 ноября 2021, 13:23 ...

Если говорить кратко, то создаётся не совсем объект Path, а объект класса, который его реализует. Метод Paths.get() на самом деле делегирует вызов метода Path.of(). Этот метод вызывает FileSystems.getDefault().getPath(first, more). Метод getDefault() возвращает одну из реализаций FileSystem, которая зависит от ОС (например в Linux вы получите объект LinuxFileSystem , который расширяет класс UnixFileSystem), и у возвращённого объекта вызывается getPath(), который вернёт окончательный объект класса, который реализует Path (у той же Linux это будет UnixPath).

Исходный код этих классов Osgle не предоставляет, но в этом и вся прелесть инкапсуляции, зачем лишний раз лезть под капот.))

Ответить

− +3 +

Hamlet simonyan Уровень 32

4 мая, 00:32 ...

ага сложное скрыть простое показать

Ответить

− 0 +

vladimir Уровень 24, Москва

25 апреля 2021, 15:14 ...

Думаю логичнее было бы поменять местами эту тему и прошлый блок задача. Я уже всё изучил, что бы решить задачи те и уже всё знаю. Верните мою материю

Ответить

− 0 +

Maks Panteleev Java Developer в Bell Integrator

30 апреля 2021, 15:22 ...

это фишка жабараша - сначала заставить тебя два дня гуглить, а потом дать лекцию по этой теме)))

Ответить

− +6 +

Flexo Bending Unit #3370318

10 марта 2021, 21:11 ...

Кстати, шо за ограбление на ровном месте? Я уже потратил 21 единицу материи, когда тыкал на предыдущей страничке "Следующая лекция"

Ответить

− 0 +

Vladrip Уровень 41, Ивано-Франковск, Украина

14 марта 2021, 23:27 ...

Та какая разница госпаде

Ответить

− +5 +

vladimir Уровень 24, Москва

25 апреля 2021, 15:13 ...

Каждая материя на вес золота !

Ответить

− +1 +

Александр Горохов Уровень 24, Дятьково, Россия

1 февраля, 12:36 ...

Ну если задачи скипать, то да

Ответить

− 0 +

Pavel Kurchavov Support Engineer в Ozon.ru

3 марта 2021, 16:54 ...

Зато задачи были по Paths, Files, Джавараш не перестает рофлить над нами)

Ответить

− +1 +

Flexo Bending Unit #3370318

10 марта 2021, 21:09 ...

Предполагалось, шо мы пролистнём сложные задачки, видимо. А потом, где-нибудь на 38-м уровне, когда энергия закончится, вернёмся к решению ♂

Ответить

− +1 +

Dmitry Vasilyev Уровень 26, Саратов, Россия

11 марта 2021, 07:20 ...

Но мы не такие и поэтому должны страдать!

Особенно обидно, что для решения 3101 задачи я использовал такой код:

```
1  for (Map.Entry<String, File> entry: fileTreeMap.entrySet()) { // итерируемся
2      File file = entry.getValue();
3      List<String> text = Files.readAllLines(file.toPath());
4      FileOutputStream output = new FileOutputStream(newFile, true);
5      for (String s : text) {
6          output.write(s.getBytes());
7          output.write("\n".getBytes());
8      }
9      output.close();
```

НАЧАТЬ ОБУЧЕНИЕ

а ведь можно было сделать намного элегантнее:

```
1  FileOutputStream output = new FileOutputStream(newFile, true);
2  for (Map.Entry<String, File> entry: fileTreeMap.entrySet()) {
3      Files.copy(entry.getValue().toPath(), output);
4      output.write("\n".getBytes());
5  }
6  output.close();
```

И не спрашивайте, зачем я клал в лист объект File, а не Path. Возможно, хотел сделать приятное валидатору в каком-то необычном месте, а может и просто проморгал эту вспышку)

Ответить

+2

Алексей Дервяго

Уровень 23, Минск, Беларусь

13 июля 2021, 04:25

```
1  fileTreeMap.values().forEach(file -> { // ...
2  });
```

Ответить

0

Anton Solovev

Уровень 26, Turkey

6 сентября 2021, 19:56

Бро, как я тебя понимаю!

Ответить

0

I_Gubenko

Уровень 41

5 февраля 2021, 15:21

long copy(InputStream, OutputStream) *Копирует байты из InputStream в OutputStream*
нет такого метода....

Ответить

+1

Pig Man

Главная свинья в Свиарнике

29 января 2021, 14:59

Создание временных файлов в Java:

Иногда нам нужно создать временный файл, который будет использоваться только в нашем приложении

```
1  Path tempFile = Files.createTempFile(String prefix, String suffix, File directory)
2
```

Этот метод создает файл с определенным суффиксом и префиксом в указанной папке. Аргумент File directory должен уже существовать и должен быть директорией, иначе будет брошено исключение IOException. Имя файла создается по формуле prefix + random_long_no + suffix. Это сделано для безопасности приложения, так как никто не будет знать имя файла и ваша программа одна будет работать временным файлом. prefix должен быть минимум три символа. Если suffix является null, то по умолчанию используется .tmp. Если в параметр directory передан null, то временный файл создается в стандартном каталоге для временных файлом операционной системы

Это самый простой способ создать временный файл в стандартной папке для временных файлом операционной системы

Files.createTempDirectory(Path) — создает временную директорию

Ответить

+10

Valua Sinicyn

Уровень 41, Харьков, Украина

20 января 2021, 14:18

Так что теперь есть два новых класса – Path и Files

java.nio.file.Path это интерфейс.

Ответить

+2

Dwarf

Ворген

3 февраля 2021, 23:52

Тоже обратила внимание. Видимо имели ввиду Paths и Files.

Ответить

0

VDT

Java Developer в USETECH

10 января 2021, 20:30

Почему-то не пишут, что строка

вы пропишите ТЕМП файлу. Хотя как его назовите. Вы поймаете эксцепшен.

Что только вот эта форма, будет корректно отрабатывать.

```
1 Files.copy(stream, tempFile, StandardCopyOption.REPLACE_EXISTING);
```

Если раньше в лекциях просто было модно не договорить об 90% функционала, то теперь походу настали времена, немного даже вводить в заблуждение.

Поправьте меня, кто там на 40х левелах, если я не прав.

Ответить

 +22 

hidden #2547125 Уровень 29 14 марта 2021, 17:12

...

Спасибо, Зашел в комменты ради этого.

Ответить

 +2 


Михаил Уровень 22, Санкт-Петербург, Россия 31 октября 2021, 14:11

...

Самое смешное, что приведенная вами конструкция, хоть и рабочая, но не правильная с т.з. логики.
Т.е. мы создаем временный файл, получаем его имя, а потом перезаписываем его, зачем и создавать?
Тут по хорошему нужно просто сгенерировать вручную видимо временное имя файла, либо записать в уже созданный и существующий временный файл.

Ответить

 0 



 Показать еще комментарии

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА



 Русский

▼

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

