

File, Files, Path

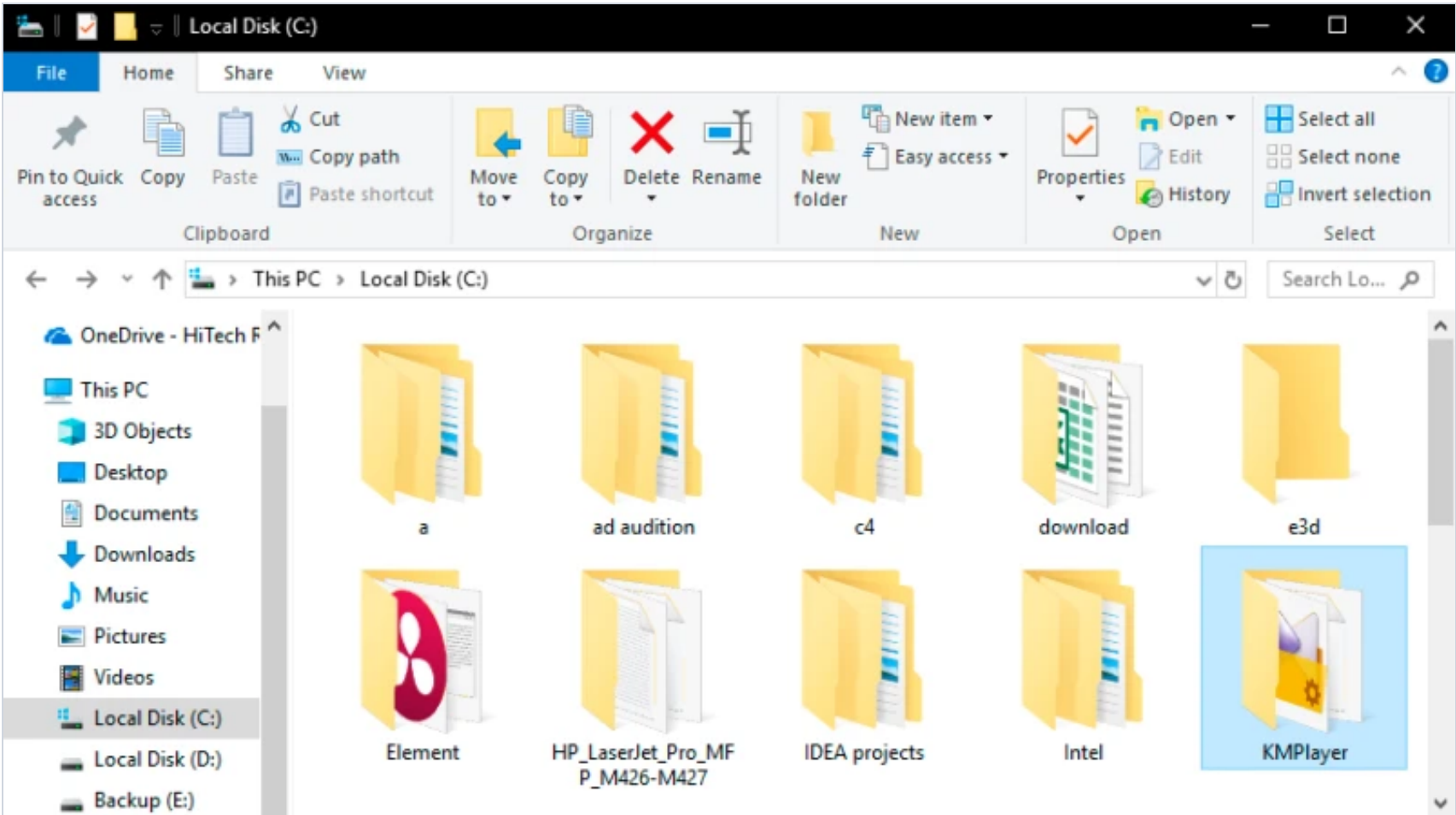
Java Collections
1 уровень, 1 лекция

ОТКРЫТА

— Привет, Амиго. Давно не виделись.

— Привет, Билаабо. О чем будешь рассказывать?

— **Сегодня я расскажу о работе с файлами.** В Java есть специальный класс (File), с помощью которого можно управлять файлами на диске компьютера. Для того чтобы управлять содержимым файлов, есть другие классы: FileInputStream, FileOutputStream,...



— Интересно. А когда ты говоришь «управлять файлами», что ты имеешь в виду?

— Как раз сейчас и расскажу. Файлы можно создавать, удалять, переименовывать и еще много чего. Практически во все классы, которые работают (читают, пишут, изменяют) с содержимым файла, можно передавать объект класса File. Пример:

Можно сразу передавать имя файла в FileInputStream	
1	FileInputStream input = new FileInputStream("c:/path/a.txt");

А можно отдельно создать объект-файл, а потом передать его в FileInputStream	
1	File file = new File("c:/path/a.txt");
2	FileInputStream input = new FileInputStream(file);

— Но во втором случае длиннее же получается. Так и не понял – зачем эти файлы нужны.

— Для этого конкретного случая – да. Это не пример, как надо делать, а скорее – как можно. Но вот представь, что тебе нужно вывести на экран список всех файлов, которые находятся в определенной директории (папке). Вот как это можно сделать с помощью файлов:

```
1 File folder = new File("c:/path/");
2 for (File file : folder.listFiles())
3 {
4     System.out.println(file.getName());
5 }
```

— **listFiles()** – это метод, который возвращает список файлов в папке с именем «с:/path/»?

— Да. Хотя программисты обычно говорят «директория» или «каталог». Название «папка» стало употребляться совсем недавно, но, в принципе, они все верные, и ты можешь говорить, как тебе удобнее.

— Ок. А **getName()** что делает? Выдает имя файла? И какое именно имя? Полное вместе с путем или только имя самого файла?

— Только имя самого файла. Для полного есть file.getAbsolutePath()

— А какие еще методы есть у класса File?

— Вот, смотри:

Метод	Описание
<code>boolean isDirectory()</code>	Является ли «объект файла» директорией
<code>boolean isFile()</code>	Является ли объект файлом
<code>long length()</code>	Возвращает размер/длину файла в байтах.
<code>boolean createNewFile()</code>	Создает файл. Если такой файл уже был, возвращает false.
<code>boolean mkdir()</code>	Создает директорию. Название mkdir происходит от «make directory».
<code>boolean mkdirs()</code>	Создает директорию и все поддиректории.
<code>boolean delete()</code>	Удаляет файл объекта на диске. Если объект – директория, то только, если в ней нет файлов.
<code>void deleteOnExit()</code>	Добавляет файл в специальный список файлов, которые будут автоматически удалены при закрытии программы.
<code>File createTempFile(</code> <code>String prefix,</code> <code>String suffix,</code> <code>File directory)</code>	Создает «временный файл» — файл с случайно сгенерированным уникальным именем – что-типа «dasd4d53sd». Дополнительные параметры – префикс к имени, суффикс (окончание). Если директория не указана, то файл создается в специальной директории ОС для временных файлов
<code>boolean exists()</code>	Возвращает true, если файл с таким именем существует на диске компьютера.
<code>String getAbsolutePath()</code>	Возвращает полный путь файла со всеми поддиректориями.
<code>String getCanonicalPath()</code>	Возвращает канонический путь файла. Например преобразовывает путь «с:/dir/dir2/ /a txt» к пути

<code>String[] list()</code>	Возвращает массив имен файлов, которые содержатся в директории, которой является текущий объект-файл.
<code>File[] listFiles()</code>	Возвращает массив файлов, которые содержатся в директории, которой является текущий объект-файл.
<code>long getTotalSpace()</code>	Возвращает размер диска (количество байт) на котором расположен файл.
<code>long getFreeSpace()</code>	Возвращает количество свободного места (количество байт) на диске, на котором расположен файл.
<code>boolean renameTo(File)</code>	Переименовывает файл – содержимое файла фактически получает новое имя. Т.е. можно переименовать файл «c:/dir/a.txt» в «d:/out/text/b.doc».
<code>String getName()</code>	Возвращает только имя файла, без пути.
<code>String getParent()</code>	Возвращает только путь (директорию) к текущему файлу, без самого имени.
<code>Path toPath()</code>	Возвращает объект Path, который соответствует текущему объекту File.

— Ничего себе! А немаленький такой список получился. Да и вроде, можно довольно много сделать с помощью него: создавать, удалять файлы, переименовывать,...

А чтобы получить директорию текущего файла, надо вызвать `getParent()`?

— Ага, но он вернет не объект-файл, а строку – путь к файлу. Фактически у класса File почти все методы дублированы: одна версия возвращает String, вторая File. Вот смотри:

```
1 File file = new File("c:/path/a.txt");
2 String directory = file.getParent();
```

```
1 File file = new File("c:/path/a.txt");
2 File directory = file.getParentFile();
```

Если у тебя есть строка с путем к файлу, а надо объект **File**, то воспользуйся конструктором. Если же наоборот – есть объект **File**, а нужна строка – используй `getAbsolutePath()`. Пример:

```
1 String path = "c:/a.txt";
2 File file = new File(path);
```

```
1 File file = new File("c:/a.txt");
2 String path = file.getAbsolutePath();
```

— Понял.

— Отлично. Тогда вот тебе маленькое задание – выведи имена всех файлов, которые есть в той же директории, что и текущий файл.

— Нет ничего проще, вот, смотри:

```
1 //какой-то текущий файл
2 File originalFile = new File("c:/path/dir2/a.txt");
3
4 //объект-директория
5 File folder = originalFile.getParentFile();
6
7 //печать списка файлов на экран
8 for (File file : folder.listFiles())
9 {
10     System.out.println(file.getName());
11 }
```

— Гм. Правильно.

— А то! Немного путает, что и для файла и для директории используется один и тот же класс – File. Как то это не очень логично, мне кажется.

— Так исторически сложилось. Раньше директория была специальным «пустым» файлом на диске. Сейчас уже, конечно, многое изменилось, но не везде. У меня на сегодня все.

— Спасибо, за отличную лекцию, Билаабо.

 +206 

Комментарии (86)

популярныеновыестарые

JavaCoder

Введите текст комментария

Сkarlett Уровень 32, Vilnius, Литва 11 сентября 2021, 12:13 

Насколько сложно было добавить лекцию про Files.walkFileTree? Объяснить русским языком то, на что ориентированы все задачи в уровне?

Ответить  +18 


Artur Уровень 31, Canada 10 марта, 05:30 

Бред просто... Каким боком эти задачи вообще к материалу

Ответить  0 

Дмитрий Б. Уровень 29, Благовещенск, Россия 2 сентября 2021, 10:39 

Ох, после 2 недель решения алгоритмических задачек, чтение теории прям как глоток свежего воздуха.

Ответить  +9 

Юрий Уровень 31, Калининград, Россия 21 июня 2021, 16:04 

Как не хватало этой лекции в Java Multithreading!!!!

Ответить  +8 

Александр Уровень 41, Екатеринбург 24 июля 2021, 12:54 

Да. В задачке про архиватор по сути все это уже пришлось выучить, но самому, с помощью гугла и такой то матери))))

Ответить  +2 

Anonymous #2498733 Уровень 34 25 апреля 2021, 15:55 

теперь класс File считается устаревшим, и использовать его не рекомендуется. Вы, конечно, можете еще встретить его в коде, параметрах методов или конструкторах классов.

Мы будем начинать изучение работы с файлами сразу с класса Path. Path — это класс, который пришел на смену File. Работа с ним безопаснее и эффективнее."



Ответить

+12

Anonymous #2539565 Уровень 47, Минск, Беларусь

25 ноября 2021, 19:37

а я вот читаю лекцию и в голове крутился этот материал) оказывается точно такое было)

Ответить

+2

Viacheslav Уровень 26, Киев, Украина

24 марта 2021, 15:09

Пришёл сюда, после CORE. Кто сделал также, ставьте лайк! Спасибо. Кто пошёл в мультитрид, тоже лайк :)

Ответить

+18

LuneFox инженер по сопровождению в BIFIT EXPERT

14 декабря 2021, 04:04

А в каком случае можно не ставить лайк?

Ответить

+15

Антон Уровень 23, Санкт-Петербург, Россия

20 мая, 16:03

В случае если ты пришел сюда после Syntax)))
ну или случайно забрел

Ответить

+3

LuneFox инженер по сопровождению в BIFIT EXPERT

20 мая, 17:25

Эти случаи исключены :)

Ответить

0

Евгений режиссер в киностудия Лисоцвет

около 3 часов назад

Я вот пришел сюда именно после Syntax, в поисках знаний для решения задачи 16 уровня.
Постоял у закрытых ворот, и пошёл себе дальше...()

Ответить

0

SolomonVP 🌞 🍵 в 🧑💻

1 марта 2021, 21:29

Пришлось повторять всё начиная с reader\writer(inputStream\OutputStream, FileInputStream\FileOutputStream).)
Чтобы освежить.

Ответить

+1

ivan tay Уровень 30, Киев, Украина

10 марта 2021, 16:21

Когда ты, проходишь материал несколько раз, начинаешь понимать вещи, которые по началу были непонятны!

Ответить

+7

Игорь Уровень 22, Могилёв, Беларусь

15 февраля 2021, 18:59

Кто пришел сюда после CORE? Не скупимся на лайк)) этож не драгоценную тёмную материю растрачивать))))

Ответить

+31

Pig Man Главная свинья в Свинарнике

29 января 2021, 14:29

File:

Класс для манипуляции над самим файлом (объект класса можно передавать в качестве пути к файлу там, где он требуется). У класса File был ряд недостатков. Например, в нем не было метода cору(), который позволил бы скопировать файл из одного места в другое. Кроме того, в классе File было достаточно много методов, которые возвращали boolean. При ошибке такой метод возвращает false, а не выбрасывает исключение, что делает диагностику ошибок и установление их причин очень непростым делом

Вместо единого класса File появились целых 3 класса: Paths, Path и Files

[Дальнейшая лекция](#)

(Добавлено)
А, дальше будет про это, но можно почитать и сейчас

Ответить

+10

Никита Навмов Уровень 33. Санкт-Петербург

19 января 2021. 07:45

НАЧАТЬ ОБУЧЕНИЕ

Ответить

Елизавета Андрюшина

Уровень 26, Москва

4 января, 19:26

...

Огромное спасибо!

Ответить

−

+4

+

0

+

Future Man

Уровень 25

18 января 2021, 18:43

...

Блин кто прошел курс может подскажут какой лучше проходить? Этот или мультипоточность? Или де параллельно идти?

Ответить

−

+2

+

Slaby

Уровень 35, Краснодар, Россия

25 февраля 2021, 22:20

...

Многопоточность.

Ответить

−

+2

+

Александр

Уровень 41, Екатеринбург

24 июля 2021, 12:56

...

Сначала многопоточность, потом этот. Так фундаментальнее знания в голове отложатся.

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

