

# Задачи | 1 уровень | 6 лекция

Java Collections  
1 уровень, 6 лекция

ОТКРЫТА

— Привет, Амиго! Я придумал тебе пару интересных задач.

Решать их можно только в IntelliJ IDEA. Вот, смотри, какие интересные условия...

Задача \* Java Collections, 1 уровень, 6 лекция

РЕШЕНА

HARD

## Разархивируем файл

★★★★☆

×28

Открыть

Задача 🏠 Java Collections, 1 уровень, 6 лекция

РЕШЕНА

HARD

## Добавление файла в архив

★★★★☆

×28

Открыть

<    Предыдущая лекция

×21

− +49 +

Комментарии (132)

популярные    новые    старые

JavaCoder

Введите текст комментария

Дмитрий Щебрюк    Уровень 23, Москва, Russian Federation

14 июня, 19:21    ⋮

Вся сложность задач в валидаторе.

Ответить

− +2 +

Евгений Т.    Уровень 34, Москва, Россия

6 апреля, 07:38    ⋮

Часто вспоминаю, слова под одной из задач Java Syntax квеста, когда кто-то хаял там условия задачи, что мол непонятно написано, один знающий человек написал, радуйтесь до 20 уровня все хорошо и

НАЧАТЬ ОБУЧЕНИЕ

Сергей Смирнов

Уровень 35

29 апреля, 12:02

...

Точно-точно, было такое

Ответить

−

0

+

Артем Горин

Software Developer в noname

3 марта, 09:21

...

try (ZipInputStream is = new ZipInputStream(new  
SequenceInputStream(Collections.enumeration(fileInputStreams))))

очень красивое решение, жаль что не объясняют

Ответить

−

+2

+

Александр Горохов

Уровень 24, Дятьково, Россия

6 февраля, 16:25

...

Кто сюда только пришёл - удачи вам, крепитесь! Знаний с предыдущих лекций явно не хватит, так что, если хотите сами решить эти 2 задачи, ищите инфу сами. Я пошёл этим путём, но потратит неприлично много времени. Хотя, как знать, может буду работать архивариусом.

Ответить

−

+2

+

Виталий

Работает в поте лица

20 января, 00:02

...

Жесть какая-то.

Ответить

−

+3

+

Greatsky

future developer в future developer

8 января, 13:38

...

Ребят подскажите во второй задаче использовал HashMap в которую как ключ записал имя файла а в значение поток ByteArrayOutputStream - которые потом закрыл когда писал файл (ByteArrayOutputStream через метод toByteArray() ) записал в zip.write(); валидар не принял из-за слишком долго тестирования - почему в решении используется byte[] buffer = new byte[1024]; именно 1024 если если шликом длинные файлы будет подписать? аля фильм на 30 гигов?

Ответить

−

0

+

Павел

Уровень 35, Минск, Беларусь

19 декабря 2021, 15:59

...

Ребята, не страдайте фигнёй, и не пытайтесь угодить валидатору.  
Полностью готовое и рабочее решение может не приниматься по всем пунктам, хотя проблем с ним никаких нет. Я такое вообще первый раз вижу.  
Сделали задачу, протестировали на своих данных, скормили "эталонное" решение, и поехали дальше.  
Не теряйте время.

Ответить

−

+6

+

Максим Дудин

Уровень 38, Калининград

13 декабря 2021, 00:17

...

Прикольно судя по коментам первая была когда-то easy =))) а теперь hard, причём как по мне такой полновесный... часов пять /шесть у меня на задачу уходило...

Ответить

−

0

+

fFamous

Уровень 51, Санкт-Петербург

4 декабря 2021, 01:06

...

Если кому интересно, тут такое дело. Тестил свой код на чтение zip'a и разархивацию всего добра в папку и по кд ловил ошибки, мол, не существует пути по которому нужно записать файл. Окей, пошел смотреть, дебагером доходил до проверки на то, является ли ZipEntry папкой и если да, то тогда она создается, но этот момент почему-то сразу пропускался, хотя должна была идти папка, а ZipEntry оказывался файлом. Искал, искал, в чем может быть косяк и наткнулся на сообщение какого-то товарища из Китая:

ZipInputStream only returns entry a/b.txt first before entry "a" (if there is one) when entry "a/b.txt" was placed before the directory entry "a" in the zip file when the zip file was created. ZIP spec does not require the ordered "placement" of the file and its parent(s) directory in the zip file, and in fact the parent directory entries can be totally absent, such as the deployed rt.jar, in which all directory entries might have been totally striped for performance reason. So the "zip extracting" implementation needs to take this one in count. The attached implementation obviously assumes that there MUST be the parent directory entry before the corresponding file entry present, which is WRONG. The correct way to do that is to get the parent path of the file entry, check if it exists, if not, File.mkdirs() before trying to extract the file entry into it, like what the jar tool does (and probably all other zip/unzip tool does).

В общем, я создал архив выделив все папки в которых были и файлы и создал через всплывающее меню винды. Видимо, там нет нормального ордеринга в таком случае.  
Потом, я создал архив отдельно и добавлял в него папки по очереди и ВСЕ, УРА, РАБОТАЕТ

Ответить

−

0

+

Ars

Уровень 41

15 ноября 2021, 16:47

...

Инфа по последней задаче:  
Среднее количество попыток для этой задачи 5.04. Всего эту задачу решили 6177 учеников.

Ответить

−

0

+

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

▼

