

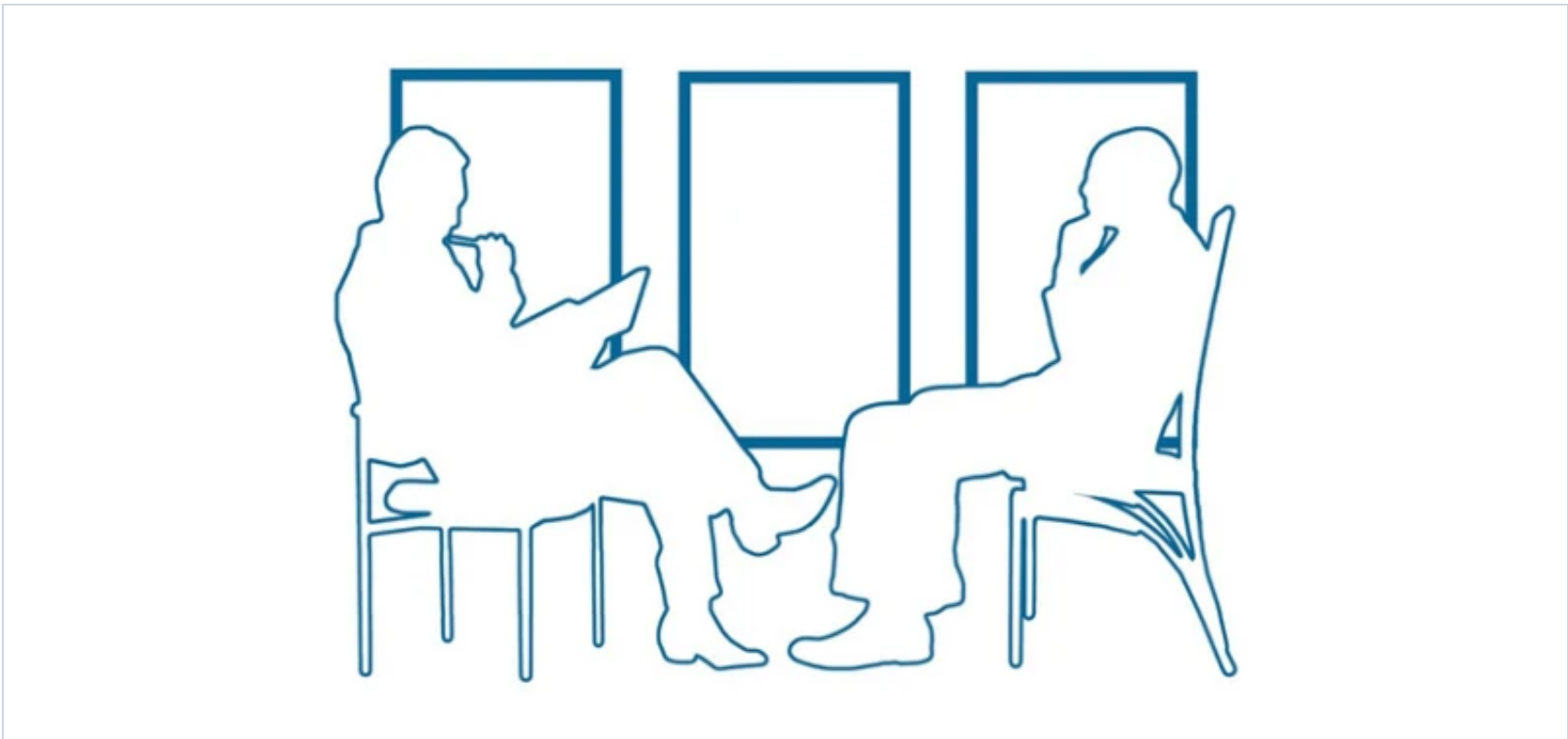
DefNeo
36 уровень

15.09.2016 7649 13

Уровень 32. Ответы на вопросы к собеседованию по теме уровня

Статья из группы Архив info.javarush.ru
15197 участников

Присоединиться



1. Зачем нужен `RandomAccessFile` ?

`RandomAccessFile` — класс пакета Java IO API, он позволяет перемещаться по файлу, читать из него или писать в него, как вам будет угодно. Вы также сможете заменить существующие части файла, речь идет о обновлении содержимого файла, а точнее о обновлении фрагмента файла. Это невозможно сделать с помощью `FileInputStream` или `FileOutputStream`, но `RandomAccessFile` даст вам эту возможность.

Ссылка: [RandomAccessFile и его возможности](#)

2. Что будет если файл, откуда читает `RandomAccessFile`, не существует?

Будет `FileNotFoundException`

3. Что будет если файл, куда пишет `RandomAccessFile`, не существует?

Создаст новый и запишет в него.

4. Зачем нужен класс `StringReader` ?

The `Java.io.StringReader` class is a character stream whose source is a string.

Представляет собой поток символов, чей источник называется строкой

НАЧАТЬ ОБУЧЕНИЕ

```
public class StringWriter  
extends Writer
```

A character stream that collects its output in a string buffer, which can then be used to construct a string.

Closing a `StringWriter` has no effect. The methods in this class can be called after the stream has been closed without generating an `IOException`.

Поток символов, собирающий свой поток в буфер строк, которые затем могут быть использованы для создания строки.

6. Зачем нужен класс `ByteArrayStream` ?

Итак, `ByteArrayInputStream` и `ByteArrayOutputStream`.

Эти классы по сути чем-то похожи на `StringReader` и `StringWriter`. Только `StringReader` читал символы (`char`) из строки (`String`), а `ByteArrayInputStream` читает байты из массива байт (`ByteArray`).

`StringWriter` писал символы (`char`) в строку, а `ByteArrayOutputStream` пишет байты в массив байт у него внутри. При записи в `StringWriter` строка внутри него удлинялась, а при записи в `ByteArrayOutputStream` его внутренний массив байт тоже динамически расширяется.

7. Зачем нужен класс `PrintStream` ? Назовите места, где он используется?

Класс `PrintStream` был придуман для читабельного вывода информации. Он практически весь состоит из методов `print` и `println`.

8. Зачем нужен `DynamicProxy` ?

В Java есть специальный класс (`java.lang.reflect.Proxy`), с помощью которого фактически можно сконструировать объект во время исполнения программы (динамически), не создавая для него отдельного класса.

9. Как работает RMI?

RMI расшифровывается Remote Method Invokation – удаленный вызов методов. Или другими словами RMI – это механизм, который позволяет объекту в одной Java-машине вызывать методы объекта в другой Java-машине, даже если они находятся на разных компьютерах, в разных странах, на разных сторонах земного шара.

Традиционный подход к выполнению кода на других машинах, разнесенных по сети может смутить из-за своей нудной и склонной к ошибкам реализации. Лучший способ рассмотреть эту проблему состоит в предположении, что некоторые объекты располагаются на другой машине, и что вы можете посылать сообщения этим удаленным объектам и получать результат, как будто они располагаются на вашей локальной машине. Это упрощение в точности является тем, что позволяет делать Удаленный Вызов Методов (RMI) в Java.

Вот статья по созданию собственной реализации RMI: [Удаленный вызов методов \(RMI\)](#)

10. Объекты каких типов можно передавать по RMI?

Объекты должны имплементировать интерфейс `Serializable`

Remote method Invocation — механизм, который позволяет вызывать метод удалённого объекта. Согласно ему, все операции по подготовке и передаче данных инкапсулируются в вызываемом методе клиентского объекта-заглушки (stub). Сам же вызов метода ничем не отличается от вызова метода обычного локального объекта, за небольшим исключением:

- локальные объекты передаются по значению (копии);
- при передаче удалённого (`Remote`) объекта, если он экспортирован, передаётся stub этого объекта;
- передаваемые объекты должны быть `Serializable`;
- кроме всех прочих исключительных ситуаций, при вызове удалённого метода может возбуждаться исключение `RemoteException` (ошибки маршализации/демаршализации, передачи данных и другие возможные ошибки протокола);

Так же нужно отметить, что при вызове метода мы работаем с удалённым интерфейсом, а не с удалённым классом.

Комментарии (13)

популярные

новые

старые

JavaCoder

Введите текст комментария

misha_lazarev

Уровень 34, Ростов на Дону, Russian Federation

7 июня, 19:43

...

Спасибо всем за ответы.

Ответить

−

0

+

Евгений

Уровень 35

31 марта, 20:31

...

Немного поумничаю) Пункт 4

The Java.io.StringReader class is a character stream whose source is a string.

Представляет собой поток символов, чей источник называется строкой

Если дословно, то должно быть так:

The Java.io.StringReader class есть символьный поток, чей источник есть строка.

Ответить

−

0

+

LuneFox

инженер по сопровождению в BIFIT

EXPERT

24 декабря 2021, 15:00

...

7. Зачем нужен класс *PrintStream*? **Назовите места, где он используется?**

Не увидел ответа на вопрос. (Самый простой пример использования - переменная `System.out`, являющаяся экземпляром этого класса, которую мы используем для печати информации в консоль.)

10. Объекты каких типов можно передавать по RMI?

Я здесь ответил по-другому, судя из лекций, которые я прочитал. По RMI можно передавать (публиковать) объекты, поддерживающие интерфейс Remote. При этом аргументы вызываемых методов у этих объектов должны поддерживать интерфейс Serializable, чтобы их можно было упаковать и распаковать при передаче по сети.

Ответить

−

+3

+

Pavel Soros

Уровень 34, Санкт-Петербург, Россия

27 июля 2021, 14:41

...

По 8 вопросу Dynamic Proxu малова-то написано. Всё таки там тема пошире будет.

Ответить

−

+1

+

barracuda

Уровень 41, Санкт-Петербург, Россия

EXPERT

29 января 2021, 00:56

...

Спасибо.

Ответить

−

0

+

Кех

Уровень 38, Тольятти

EXPERT

30 июня 2020, 16:47

...

Ну на вопросы зачем тут ответили словами из лекции, которые в свою очередь вообще ни как не ответили на этот вопрос...(

Ответить

−

0

+

MURS

Уровень 35, Киев

15 июня 2017, 01:14

...

Remote Method Invocation — поправьте, пожалуйста. Думаю, для многих людей, чей уровень английского даже ниже среднего, такие строки читаются с душераздирающей болью

Ответить

−

+4

+

Павел X.

Team Lead в Иннотех

EXPERT

14 декабря 2018, 02:24

...

Думаю, для многих людей, чей уровень английского даже ниже среднего, нет будущего в программировании.

Ответить

−

+9

+

PaIMai in IF

Grand Master в Eagles' Claw

19 октября 2021, 22:24

...

НАЧАТЬ ОБУЧЕНИЕ

как бы, достаточно хороший и плотный фундамент, т.е. когда я еще будучи школьником гонял в Англию по обмену на 2 недели и жил в чисто аутентичной англиканской семье, то этого уровня с лихвой хватало на общение + на посещение и активное прослушивание всех топчик экскурсий Лондиниума, конечно читать Дикенса в оригинале будет трудновато, но думаю для разраба pre-intermediate тоже будет достаточно хорошо.

Ответить

+2

Appolinaryy

Уровень 35, Россия

5 апреля 2017, 14:55

во всей литературе(как и в данном случае, и в лекции курса по RMI) написано, что при работе с RMI передаваемые объекты должны быть Serializable, но нигде в коде я этого факта не наблюдал? Remote в свою очередь тоже не наследует от Serializable. Как так?

Ответить

+1

IichMax

Уровень 40, Санкт-Петербург, Россия

3 июля 2017, 22:36

не сам объект, а параметры удалённых методов и их возвращаемые значения. Именно так везде написано (по крайней мере — в лекции).

Ответить

+4

Sanakan

Уровень 39, Маунтин-Вью, США

5 декабря 2016, 16:15

По вопросу №2, если в конструкторе «gw», то исключение не кинется, а так же создастся файл.

Ответить

+1

IichMax

Уровень 40, Санкт-Петербург, Россия

3 июля 2017, 22:33

Вообще-то это уже третий вопрос (чтение и запись). И если этот файл создать не удастся, то тоже будет выброшено исключение FileNotFoundException.

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



