



Стандартные интерфейсы: InputStream, OutputStream

Java Core (/quests/QUEST_JAVA_CORE)
3 уровень (/quests/lectures/?quest=QUEST_JAVA_CORE&level=3), 8 лекция (/quests/lectures/questcore.level03.lecture08)

ОТКРЫТА

— Привет, Амиго! Сейчас я тебе расскажу про два интерфейса – InputStream и OutputStream. Объявлены они как абстрактные классы, но если начать разбираться, то можно увидеть, что по своей сути – это интерфейсы. Почти все их методы абстрактные, кроме нескольких незначущих методов. Очень похожи на нашего «телохранителя», которого мы рассматривали.

Это очень интересные интерфейсы. Пока что я специально буду называть их интерфейсы, чтобы ты понял, зачем они нужны. А потом мы поговорим, почему же их все-таки сделали абстрактными классами.

— Хорошо. Так что это за интерфейсы?

— Сейчас расскажу.

Есть такая интересная вещь в Java как «поток». Поток – это очень простая сущность. И его простота есть залог очень мощного механизма обмена данными. Потоки бывают двух видов: поток для чтения и поток для записи.

В поток для записи, как ты уже, наверное, догадался, можно записывать данные. Для этого у него есть метод write(). Из потока для чтения можно данные читать. Для этого у него есть метод read().

InputStream – это интерфейс потока чтения, описывающий такую способность: «из меня можно читать байты».

A OutputStream– это, соответственно, интерфейс потока записи, описывающий способность: «в меня можно записывать байты».

— И это все?

— Фактически да. Но все дело в том, что в Java есть очень много классов, которые умеют работать с интерфейсами InputStream и OutputStream. Например, ты хочешь прочитать файл с диска и вывести его содержимое на экран. Нет ничего проще.

Для того, чтобы прочитать данные из файла на диске, есть специальный класс FileInputStream, который реализует интерфейс InputStream. Хочешь записать прочитанные данные в другой файл? Для этого есть класс FileOutputStream, который реализует интерфейс OutputStream. Вот как выглядит код копирования [данных одного] файла в другой.

Код

```
1 public static void main(String[] args) throws IOException
2 {
3     InputStream inStream = new FileInputStream("c:/source.txt");
4     OutputStream outStream = new FileOutputStream("c:/result.txt");
5
6     while (inStream.available() > 0)
7     {
8         int data = inStream.read(); //читаем один байт из потока для чтения
9         outStream.write(data); //записываем прочитанный байт в другой поток.
10    }
11
12    inStream.close(); //закрываем потоки
13    outStream.close();
14 }
```

Представь, что мы написали класс, и добавили ему способности InputStream и OutputStream.

Если мы корректно реализовали поддержку этих интерфейсов, то объекты нашего класса теперь можно сохранить в файл на диске. Просто вычитав их содержимое через метод read. Или загрузить из файла, создав объект и записав в него содержимое файла через метод write.

— А можно пример?

— Можно.

Код**Описание**

```
1 class MyClass
2 {
3     private ArrayList<Integer> list;
4 }
```

Для простоты представим, что наш класс содержит в себе один объект – ArrayList типа Integer.

Теперь добавим в него методы read и write

Код**Описание**

```
1 class MyClass
2 {
3     private ArrayList<Integer> list;
4     public void write(int data)
5     {
6         list.add(data);
7     }
8     public int read()
9     {
10        int first = list.get(0);
11        list.remove(0);
12        return first;
13    }
14
15    public int available()
16    {
17        return list.size();
18    }
19 }
```

Теперь у нас в классе реализован метод read, который позволяет последовательно вычитать все содержимое нашего списка list.

И метод write, который позволяет записывать в наш list значения.

Это, конечно, не реализация интерфейсов InputStream и OutputStream, но очень похоже.

— Да, это понятно. А как все-таки сохранить содержимое такого объекта в файл?

— Давай я напишу тебе пример:

Запись объекта MyClass в файл

```
1 public static void main(String[] args)
2 {
3     MyClass myObject = new MyClass();
4     OutputStream outStream = new FileOutputStream ("c:/my-object-data.txt");
5
6     while (myObject.available() > 0)
7     {
8         int data = myObject.read(); //читаем один int из потока для чтения
9         outStream.write(data); //записываем прочитанный int в другой поток.
10    }
11
12    outStream.close();
13 }
```

ЗАДАЧА **T** Java Core, 3 уровень, 8 лекция

ДОСТУПНА

★★★★☆

Набираем код

×14

Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. Вот почему во время обучения в секретном центре JavaRush вы иногда встречаете задания на набор кода. Набирая код, вы привыкаете к синтаксису и зарабатываете немного материи. А ещё — боретесь с ленью.

[Открыть](#)

Чтение объекта MyClass из файла

```
1 public static void main(String[] args)
2 {
3     InputStream inStream = new FileInputStream("c:/my-object-data.txt");
4     MyClass myObject = new MyClass();
5
6     while (inStream.available() > 0)
7     {
8         int data = inStream.read(); //читаем один int из потока для чтения
9         myObject.write(data); //записываем прочитанный int в другой поток.
10    }
11
12    inStream.close(); //закрываем потоки
13 }
```

— Ух ты! Действительно, очень похоже на работу с InputStream/OutputStream. Потоки – это крутая вещь!

— А то!

[< \(/quests/lectures/questcore.level03.lecture07\)](/quests/lectures/questcore.level03.lecture07)[> \(/quests/lectures/questcore.level03.lecture09\)](/quests/lectures/questcore.level03.lecture09)

Комментарии (30)

популярные

новые

старые

Никита

Введите текст комментария

Nyuruko 18 уровень, Москва

12 февраля, 10:00

Зачем нужно использовать полиморфизм такого вида?

```
1  InputStream inStream = new FileInputStream("путь_к_файлу");
```

Почему нельзя сделать тип переменной FileInputStream? Нигде не смог найти ответ...
Они так и не объяснили это с того момента, как дали пример с коровой и китом!
Еще не понятно, как они хотят применить к myObject метод из другого класса:

```
1  myObject.available()
```

И если взять код из первого примера, то он не только копирует содержимое одного файла в другой, но и удаляет ВЕСЬ старый текст из файла, в который копируют. У кого-нибудь такое было?

Ответить

+2

Дмитрий Довганюк 19 уровень

21 марта, 17:22

Ибо с помощью одного и того же интерфейса можно обращаться к AudioInputStream, ByteArrayInputStream, FileInputStream, FilterInputStream, ObjectInputStream, PipedInputStream, SequenceInputStream, StringBufferInputStream. Вернись на пару лекций, где было понятие полиморфизма и ООП.

Ответить

+1

voffka 17 уровень, Киев

29 декабря 2017, 13:04

```
1  public int read()
2  {
3      int first = list.get(0);
4      list.remove(0);
5      return first;
6  }
```

что это ? почему ремув ?

Ответить

0

Daniel Slangman 25 уровень, Алматы

2 января, 10:14

при чтении данного объекта метод read вызывается в цикле while.
при такой реализации метода с каждым повтором цикла из массива (который является единственным объектом нашего класса) считывается только один элемент с индексом 0.
для того чтобы обеспечить последовательное чтение элементов из массива, необходимо каждый раз после чтения нулевого элемента удалять его.

Ответить

+3

Аслан 30 уровень, Москва

10 октября 2017, 20:53

у кого сколько материи?
у меня 666)

Ответить

0

elfima 22 уровень

24 октября 2017, 14:23

713

Ответить

0

Никита Васильченко 35 уровень, Москва

23 ноября 2017, 10:55

687

Ответить

0

Alex 29 уровень

31 декабря 2017, 08:25

2

Ответить

0

Юрий Кузнецов 17 уровень

25 марта, 18:02

640

УЧУ

Ответить

0

Лайт 21 уровень, Санкт-Петербург

22 сентября 2017, 03:13

...

На 13-м тролли вовсю присутствуют. Интересно, на каком уровне им уже не написать будет, надоели.

Ответить

+7

Денис ден 19 уровень

16 августа 2017, 21:20

...

Это жесть просто...перемещаю объявление массива и сразу на всех условиях провал показывает, хотя до этого было сделано 3 из 5. Чтение и запись нужно объявлять в отдельных try/catch обязательно, этот валидатор раздражает уже...

Ответить

0

Марк 20 уровень, Санкт-Петербург

6 августа 2017, 16:58

...

Ну, как мы узнали, Диего вполне себе пьет пиво, и сексист. Если бы он ещё матерился и оскорблял, то был бы вылитый Бендер)

Ответить

+9

Vikentsi 22 уровень, Минск

23 июля 2017, 14:45

...

Мне нравится когда Диего читает лекции. Вообще мне нравится Диего, он как Бендор только Диего)

Ответить

+2

Alex 29 уровень

27 декабря 2017, 11:51

...

ахах, я ненавижу Бендера
особенно был эпизод где он вырезал у Фрая почку во сне и продал ее)
а потом тип да все нормально братан)

Ответить

0

Андрій 17 уровень, Киев

26 июня 2017, 12:34

...

а какая разница между двумя примерами? если в обоих примерах - сначала идет чтение, потом запись?

Ответить

+2

Akira Rokudo 20 уровень, Москва

27 июня 2017, 13:40

...

Если речь о использовании или не использовании класса MyClass, то разница в том, что мы создаем специальный объект для хранения данных-myObject. В примере этого нету, но мы можем выполнять с этими данными определенные действия. к примеру после считывания переводить все символы из ввода в нижний регистр.

Ответить

0

Сергей Зимовец 31 уровень

8 июня 2017, 18:28

...

все просто !!!

FileInputStream - берет данные по адресу и куда то передает (в нашем случае как поток байтов в переменную int)

FileOutputStream - передает данные на нужный адрес , данные передаются как параметр (в нашем случае это переменная int)

Ответить

+2

Anvarito 18 уровень, Уфа

19 апреля 2017, 19:09

...

вот если потом из массива брать то что туда попало, то выясняется что там ахинея из кучи непонятных цифр. как с этим справиться?

Ответить

0

Akira Rokudo 20 уровень, Москва

27 июня 2017, 13:36

...

потому что это числовое представление 1 байта. а нам нужно перевести из числового, в визуальное. Для этого есть тип char)

Ответить

+2

[↺ Загрузить еще](#)



Программистами не рождаются
© 2018