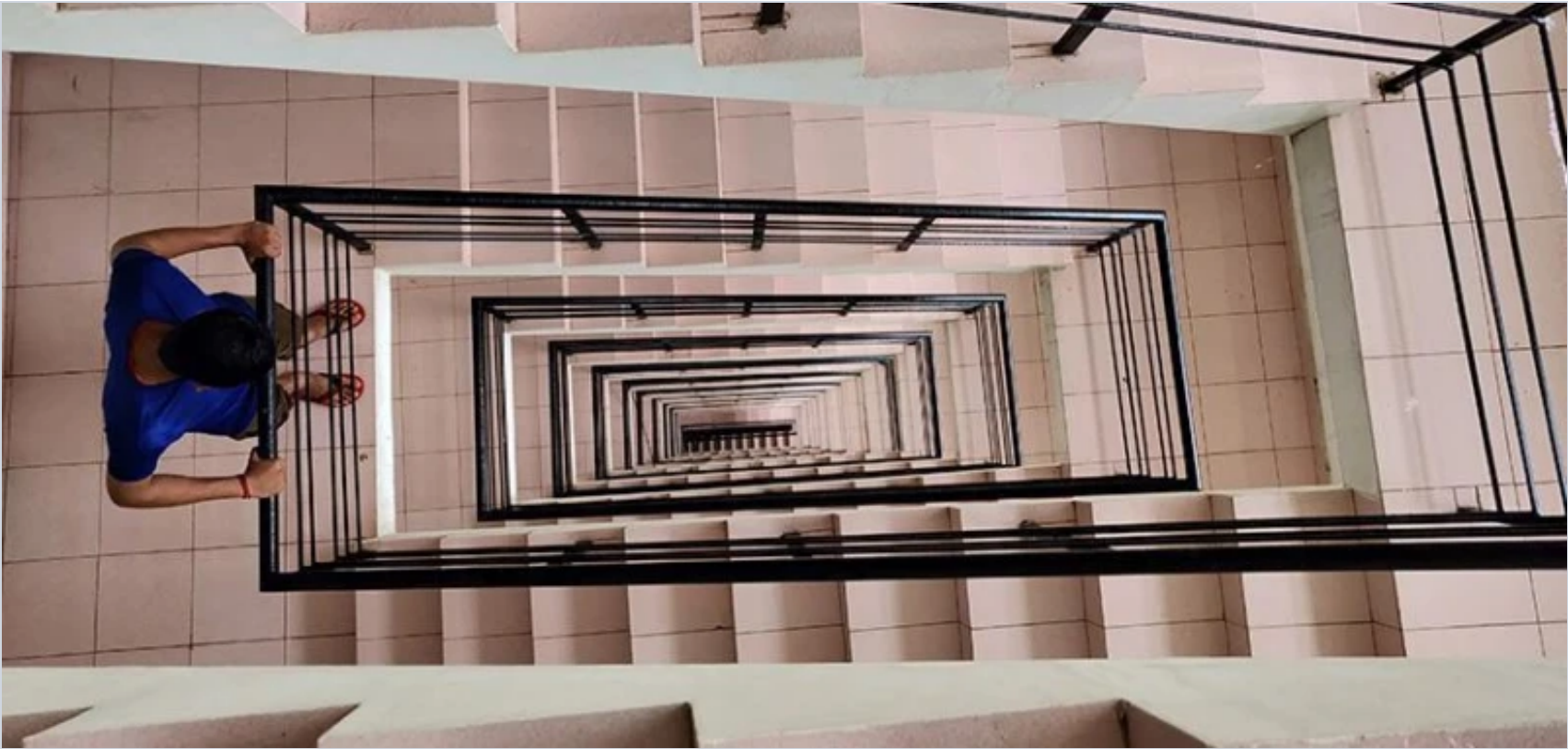


Рекурсия

Java Collections
4 уровень, 1 лекция

ОТКРЫТА

— Привет, Амиго. Сегодня Билаабо расскажет тебе, что такое рекурсия.



Как ты знаешь, в Java одни методы вызывают другие методы. При этом, при вызове метода, в него передаются конкретные значения аргументов, а во время его работы локальные переменные метода принимают некоторые значения.

— Ага.

— И как ты знаешь, внутренние переменные разных методов независимы друг от друга.

— Ага.

— Так вот, представь ситуацию, когда метод может вызвать сам себя. Именно она называется рекурсией. Пример:

Пример

```
1 public static void main(String[] args)
2 {
3     countdown(10);
4 }
5
6 public static void countdown(int x)
7 {
8     if (x <=0)
9         System.out.println("Boom!");
10    else
11    {
12        System.out.println(x);
13        countdown(x - 1);
14    }
15 }
```

НАЧАТЬ ОБУЧЕНИЕ

1	10
2	9
3	8
4	7
5	6
6	5
7	4
8	3
9	2
10	1
11	Boom!

— Честно говоря, вижу, что метод в коде сам себя вызывает, но не понимаю, что именно при этом происходит.

— Да примерно то же, что и при вызове другого метода.

— Нет, я спрашиваю, что происходит с переменными? С их значениями? И как мы выходим из метода? Или мы выходим из всех сразу?

— Господи. Да все гораздо проще. Представь что метод, который вызывает сам себя, размножили много раз. Тогда будет аналогичная ситуация:

Рекурсивный вызов метода	Что происходит «на самом деле»
<pre>1 public static void main(String[] args) 2 { 3 countDown(3); 4 } 5 6 public static void countDown(int x) 7 { 8 if (x <=0) 9 System.out.println("Boom!"); 10 else 11 { 12 System.out.println(x); 13 countDown(x - 1); 14 } 15 }</pre>	<pre>1 public static void main(String[] args) 2 { 3 countDown1(3); 4 } 5 6 public static void countDown1(int x) 7 { 8 if (x <=0) 9 System.out.println("Boom!"); 10 else 11 { 12 System.out.println(x); 13 countDown2(x - 1); 14 } 15 } 16 public static void countDown2(int x) 17 { 18 if (x <=0) 19 System.out.println("Boom!"); 20 else 21 { 22 System.out.println(x); 23 countDown3(x - 1); 24 } 25 } 26 public static void countDown3(int x) 27 { 28 if (x <=0) 29 System.out.println("Boom!"); 30 else</pre>

	<pre>33 countDown4(x - 1); 34 } 35 } 36 37 public static void countDown4(int x) 38 { 39 if (x <=0) 40 System.out.println("Boom!"); 41 else 42 { 43 System.out.println(x); 44 countDown5(x - 1); 45 } 46 }</pre>
--	--

Вывод на экран:	Вывод на экран:																
<table><tr><td>1</td><td>3</td></tr><tr><td>2</td><td>2</td></tr><tr><td>3</td><td>1</td></tr><tr><td>4</td><td>Boom!</td></tr></table>	1	3	2	2	3	1	4	Boom!	<table><tr><td>1</td><td>3</td></tr><tr><td>2</td><td>2</td></tr><tr><td>3</td><td>1</td></tr><tr><td>4</td><td>Boom!</td></tr></table>	1	3	2	2	3	1	4	Boom!
1	3																
2	2																
3	1																
4	Boom!																
1	3																
2	2																
3	1																
4	Boom!																

Т.е. **каждый раз при вызове метода (даже самого себя) создаются новые переменные, которые хранят данные для этого метода. Никаких общих переменных нет.**

При каждом вызове в памяти появляется еще одна копия аргументов метода, но уже с новыми значениями. При возвращении в старый метод, там используются его переменные. Т.е. при рекурсии **фактически мы вызываем другой метод, но с таким же кодом как наш!**

— Ясно. А как работает выход из этих методов? Можно пример?

— Ладно. Один пример стоит тысячи слов. Вот тебе пример:

Рекурсивный вызов метода	Что происходит «на самом деле»
<pre>1 public static void main(String[] args) 2 { 3 print(3); 4 } 5 6 public static void print(int x) 7 { 8 if (x <=0) 9 System.out.println("Boom!"); 10 else 11 { 12 System.out.println(x); 13 print(x - 1); 14 System.out.println(x); 15 } 16 }</pre>	<pre>1 public static void main(String[] args) 2 { 3 print1(3); 4 } 5 6 public static void print1(int x) 7 { 8 if (x <=0) 9 System.out.println("Boom!"); 10 else 11 { 12 System.out.println(x); 13 print2(x - 1); 14 System.out.println(x); 15 } 16 } 17 </pre>

	20	if (x <=0)
	21	System.out.println("Boom!");
	22	else
	23	{
	24	System.out.println(x);
	25	print3(x - 1);
	26	System.out.println(x);
	27	}
	28	}
	29	
	30	...

Вывод на экран:		Вывод на экран:	
1	3	1	3
2	2	2	2
3	1	3	1
4	Boom!	4	Boom!
5	1	5	1
6	2	6	2
7	3	7	3

— Ок. Вроде понял. А зачем нужна рекурсия?

— Есть очень много задач, которые разбиваются на отдельные подзадачи, идентичные первоначальной задаче. Например, надо обойти все элементы XML-дерева. У каждого элемента может быть несколько дочерних элементов, а у них — свои.

Или тебе нужно вывести список файлов, которые есть в директории и все ее поддиректориях. Тогда ты пишешь метод, который выводит файлы текущей директории, а потом для получения файлов всех поддиректорий вызываешь его же, но с другим параметром – поддиректорией.

Пример:

Вывод всех файлов на экран из директории и её поддиректорий	
1	public static void main(String[] args)
2	{
3	printAllFiles(new File("c:/windows/"));
4	}
5	
6	public static void printAllFiles(File dir)
7	{
8	for (File file : dir.listFiles())
9	{
10	if (file.isDirectory())
11	printAllFiles(file);
12	else
13	System.out.println(file.getAbsolutePath());
14	}
15	}

Строка 8 – получаем список всех файлов (и директорий) в директории dir.

— Ок. Вроде понял. Спасибо, Билаабо.

< Предыдущая лекция

 ×24

 +65 


Комментарии (55)

популярные новые старые

JavaCoder

Введите текст комментария



LuneFox инженер по сопровождению в **BIFIT** EXPERT 17 января, 03:30 

Если не до конца поняли, что такое рекурсия, прочитайте мой комментарий от 17 января 2022 года вот на [этой](#) странице.

Ответить  +40 

FreyGreen Уровень 41, Нижний Новгород, Россия 24 января, 23:03 

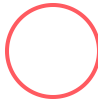
Гений


Ответить  +3 

hidden #2595317 Уровень 45 24 февраля, 13:19 


ты плохой

Ответить  +2 



Роман Кончалов Уровень 28, Россия EXPERT 30 марта, 13:19 

Понимание рекурсии – условие выхода из рекурсии, гениально

Ответить  +1 


Александр Горохов Уровень 25, Дятьково, Россия 26 июля, 15:40 

Пхаххаха!)))


Ответить  +1 

ArturZ Уровень 37, Russian Federation 2 августа, 11:50 

Переходил по ссылкам два часа кряду и преисполнился

Ответить  +2 



LuneFox инженер по сопровождению в **BIFIT** EXPERT 2 августа, 23:43 

Вкладк оверфлоу?)

Ответить  0 

ArturZ Уровень 37, Russian Federation 4 августа, 17:05 

чтобы прийти в себя пришлось идти за всяким, случилось zakladkOverflow


Ответить  0 

Maks Panteleev Java Developer в **Bell Integrator**

7 июля 2021, 10:02 

Самое время начать изучение рекурсии

Ответить  +10 

it0088 Уровень 40, Худжанд, Таджикистан 7 июля 2021, 13:05 

нет, думаю еще рано))

Ответить  +7 

Max Zap Уровень 41 26 октября 2021. 11:57 

НАЧАТЬ ОБУЧЕНИЕ

<

Ответить

Илья

Backend Developer в СберТех

7 февраля 2021, 12:53

...

поддерживаю, в свое время как-то посмотрел и сразу все понял. Вынес для себя, что всегда нужно держать в голове мысленно return (в том числе и в методах void)

Ответить

-

0

+

Александр один Эс

Backend Developer

20 ноября 2020, 09:19

...

Повторение – мать учения

Ответить

-

+3

+

Дмитрий

Уровень 26, Ровно, Украина

21 июля 2020, 23:00

...

После задач о построении дерева, факториала (через Big Integer) пора и рекурсию изучить. А то уже подраслабились мы серилизацией и xml парсингами.

Ответить

-

+8

+

Aleksey Serikov

Уровень 41, Гомель, Беларусь

10 мая 2020, 18:37

...

Если кто не понял почему в примере такой вывод в консоль: 321 boom! 321, то рекомендую посмотреть это видео: [рекурсия](#).

Ответить

-

+9

+

Ivan D

Уровень 35

5 декабря 2020, 07:23

...

3 2 1 1 2 3

Ответить

-

+1

+

Anton Rymarenko

Уровень 41

24 февраля 2020, 19:19

...

Спасибо что через два уровня после задач на рекурсию мы учим рекурсию))

Ответить

-

+20

+

Vladimir Kashin

Backend Developer в ГОСНИИАС

4 апреля 2020, 14:05

...

Действительно xD

Ответить

-

+1

+

Alexandr Ikert

Уровень 35, Москва, Россия

15 июля 2020, 20:43

...

а это такая фишка JavaRush сразу отсеивает тех у кого пукан горит)))

Ответить

-

+5

+

Vorlock

Уровень 31, Днепр, Украина

10 января 2020, 23:17

...

обычно рекурсия дается на уровне начитки java core.... жжете в общем)

Ответить

-

+9

+

↺

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



НАЧАТЬ ОБУЧЕНИЕ

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

 Русский

▼

