



Многопоточность или многопоточность

Java Core (/quests/QUEST_JAVA_CORE)
6 уровень (/quests/lectures/?quest=QUEST_JAVA_CORE&level=6), 1 лекция (/quests/lectures/questcore.level06.lecture01)

ОТКРЫТА

— Привет, Амиго! У нас новая и очень трудная тема. Сочувствую. Часто она считается одной из самых сложных не только в Java, но и в программировании вообще. Это – *многопоточность (multithreading)*.

Представь себе средненькую компьютерную игру, например, гонки на звездолетах. Ты летишь по просторам космоса, уворачиваясь от метеоритов и патрульных крейсеров, а с тобой еще два десятка таких же участников этих нелегальных гонок.

Допустим, ты решил написать такую же игру. Твоей программе придется отслеживать команды управления (ввод с клавиатуры), перемещать звездолеты, рассчитывать их траектории и последствия столкновения, а также отрисовывать все это на экране пользователя. Это очень сложная работа.

Вспомни, как мы решили «проблему большой сложности» в том примере про рост курьерской компании.

— Мы разделили ее на независимые отделы и жестко задали (стандартизировали) способы их взаимодействия.

— Но что делать, когда независимым частям нужно выполнить какой-то объем работы, параллельно с другими частями?! Ответ на этот вопрос – нити(треды) (или как их неправильно называют – потоки).

Попробуй представить программу, как *такого маленького робота*, который *бегает по коду и выполняет команды*. Сначала выполнил команду, написанную в одной строке, затем перешел на следующую, и так далее.

— Представил. Ничего сложного!

— Отлично. А теперь представь, что таких роботов у тебя несколько. И пока один занимается вводом от пользователя, второй меняет объекты в соответствии с ним. Третий выполняет код, по отображению этих объектов на экран, а четвертый несколько раз в секунду проверяет – не столкнулись ли корабли и, в случае столкновения, просчитывает его результаты.

Таким образом, мы можем разделить программу не только на независимые части/объекты, но и добиться того, что эти части будут выполнять свою работу независимо друг от друга. А чем меньше взаимодействия между отдельными частями, тем меньше сложность программы.

Представь, что ты смог *заменить менеджера – скриптом*, рассылающим письма. А остальные отделы компании об этом даже не догадались. Такие примеры уже имели место в 26 веке и показали отличные результаты. Большинство менеджеров, и даже топ-менеджеров, может быть успешно заменено скриптом средней сложности. Только после вмешательства «профсоюза офисного планктона» удалось остановить массовые увольнения менеджеров. Но это так – отвлечение от темы.

— Как интересно.

— Мало того, что в программе может быть несколько таких «маленьких роботов», исполняющих код, так эти роботы могут еще общаться друг с другом и порождать новых роботов.

— Порождать новых роботов?

— Да, для выполнения новых задач. Иногда выгодно создать еще одного робота (еще одну нить), которая будет выполнять какое-то действие одновременно с текущей нитью (роботом).

— Думаю, что это хорошая вещь, хоть пока не могу придумать, где бы я этим пользовался.

А почему это называется «нити»?

— Представь, что все роботы разного цвета, и каждый робот помечает своим цветом все команды, которые он выполнил. Таким образом, за маленьким роботом, как за карандашом, будет тянуться след. Этот след тянется за роботом, как нитка за иголкой.

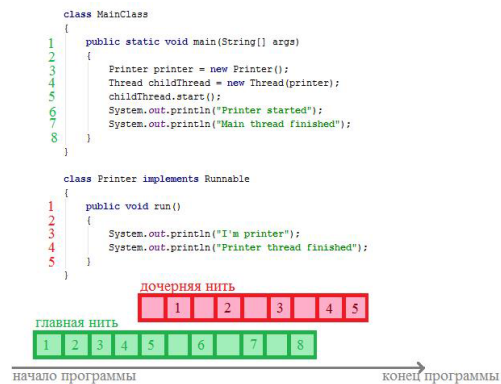
У каждого такого «маленького робота» есть задание, для исполнения которого его создали. И нить – это набор команд, выполненных в процессе исполнения этого задания.

Допустим, ты летишь на звездолете, чтобы доставить груз. Тогда «доставить груз» — это твое задание, ты в процессе его исполнения. А путь, который ты пролетел – это твоя нить. Можно сказать, что каждому новому заданию, каждой еще не решенной задаче соответствует своя нить – путь, который еще предстоит пройти.

— Другими словами, есть задание и «маленький робот», который его исполняет, а нить – это всего лишь взгляд на текущее положение дел со стороны?

— Именно так.

А вот как все это работает глубоко внутри. Т.к. процессор у компьютера один, то он может выполнять только одну команду одновременно. Поэтому происходит вот что: процессор постоянно переключается между нитями. Переключился на новую нить, выполнил несколько ее команд, затем переключился на следующую нить, выполнил несколько ее команд и так далее. Но т.к. переключение между нитями происходит сотни раз в секунду, со стороны кажется, что все нити работают одновременно.



< (/quests/lectures/questcore.level06.lecture00)

×16 > (/quests/lectures/questcore.level06.lecture02)



0



0



0



0



0

+19

Комментарии (53)

популярные

новые

старые

Никита

Дмитрий Стуков 17 уровень, Санкт-Петербург

17 февраля, 21:32



Одмен, в заголовке правильно, а в первом абзаце исправь:

Это – многопоточность (multithreading).

А эти строки выделить синим:

— Думаю, что это хорошая вещь, хоть пока не могу придумать, где бы я этим пользовался.
А почему это называется «нити»?

Ответить

+2

Redas Shuliakas 21 уровень

9 февраля, 09:39



Ответить

+2

Roman Bogdanov 20 уровень, Днепр

1 ноября 2017, 17:44



Исправьте, если я не прав: на рисунке нулевая позиция дочерней нити НЕ может существовать, так как команда нового потока сможет выполняться только ПОСЛЕ

```
1  childThread.start();
```

Ответить

0

MrDudec 22 уровень

6 ноября 2017, 17:45



Так все правильно. childThread.start(); - 5 пункт главной нити, сразу после него выполняется 1 пункт дочерней. Нулевая позиция для наглядности, чтобы было понятнее, когда запущена дочерняя нить.

Ответить

+3

soloyes 28 уровень

22 августа 2017, 06:43



Ответить

0

Азамат Ильясов 23 уровень, Алматы

8 сентября 2017, 19:17



Компилируется, просто нужно вывести class Printer за пределы MainClass (как и сделано в примере)

Ответить

+1

kotdavinci 33 уровень, Москва

21 августа 2017, 15:25



Ответить

+3

Геннадий Шевченко 32 уровень

23 сентября 2017, 06:38



Ответить

+1

NastyaGermanovich 26 уровень, Санкт-Петербург

27 сентября 2017, 11:28



Ответить

0

Alex 30 уровень

9 января, 18:14



Ответить

0

Геннадий Шевченко 32 уровень

9 января, 18:18



Ответить

+1

Alex 30 уровень

9 января, 18:24

...

я знал что ты так ответишь)
ну ясненько

Ответить

0

BlackHoleSun 17 уровень

26 января, 18:19

...

Да, если читать сначала немного из главы книги, потом лекцию и так чередовать, получается отлично :)

Ответить

0

Максим Палий 17 уровень, Днепр

18 февраля, 14:59

...

ясно

Ответить

0

fromCPPtoJava 19 уровень

18 августа 2017, 19:58

...

Очень интересно, кто-нибудь может дать МОТИВАЦИОННОЕ видео или статью, немного устал после 15го уровня((

Ответить

+1

Дмитрий 31 уровень

27 октября 2017, 11:04

...

Поддерживаю, мотивация - хороший двигатель прогресса, большинство роликов на Ютубе уже приелись. Советую посмотреть хороший фильм, для меня большим вдохновением год назад стал просмотр кинокартины "Сноуден", фильм как раз-таки про программиста, который добился гигантских высот, так же хороший фильм - "В погоне за счастьем", кто-то советует фильм - "Друзья", но я его ещё не смотрел

Ответить

0

Alex 30 уровень

9 января, 18:14

...

иди работай

Ответить

+3

Миша 20 уровень

27 января, 14:42

...

Лучшая мотивация - это количество дней подписки)

Ответить

+1

Евгений Сигитов 17 уровень

4 марта, 18:54

...

.....

Ответить

+2

Yaro Snihur 17 уровень

14 марта, 13:14

...

Мотивация? Хмм.. пустой карман, дерьмовая работа, нулевая перспектива карьеры на будущее не мотивируют? Желание быть кем-то, делать то что ТЫ выбрал, а не то что тебе попадает под руку, не мотивирует?
Если тебя мотивируют видео с ютуба "а-ля : Ты сможешь! Just do it!" то у меня плохие новости для тебя. Это все пыль самому себе в глаза. Будешь потом вкалывать на тех у кого ТА нужная мотивация как раз была.

Ответить

+1

Atom 17 уровень

31 марта, 16:00

...

Не издевайтесь над человеком. Видно же, что уровень у него вырос, значит сам справился с мотивацией)
Желаем успехов!

Ответить

0

Иван Луценко 28 уровень

9 августа 2017, 10:57

...

По-моему, действия разворачиваются раньше 26-го века... Или я что-то путаю?

Ответить

0

Donatello 32 уровень

29 августа 2017, 12:01

...

)))
История, повествуемая в лекциях, происходит якобы в 30 веке, если вы внимательно читали лекции)

Ответить

0

Иван Луценко 28 уровень

2 сентября 2017, 21:47

...

По-моему, в самом начале истории говорили то ли о 24, то ли о 25 веке...

UPD: Да, извиняюсь, история происходит в 3017 году.
Кстати, может, всё-таки, в 31 веке? ;)

Ответить

0

Михаил Чирков 16 уровень, Москва

5 июля 2017, 18:53

Я правильно понял что Многопоточность и некая "Многочитовость" это как бы одно и тоже))

Ответить

0

Олександр 19 уровень, Львов

8 июля 2017, 01:31

Правильно

Ответить

+1

Kirill Tezikov 25 уровень, Нижний Новгород

19 августа 2017, 13:51

Многочитовость - это когда два или более потока программы выполняются псевдоодновременно, при этом не важно на каком количестве ядер процессора. Многопоточность - это когда программа использует два или более ядер процессора.

Ответить

+4

Roman Bogdanov 20 уровень, Днепр

1 ноября 2017, 17:37

дайте источник, пожалуйста

Ответить

+1

Beslan Bavizhev 31 уровень

26 января, 16:38

Авторы просто не хотят смешивать термины stream (поток) и thread (нить).

Ответить

0

Sam 18 уровень, Нижний Новгород

31 мая 2017, 13:37

мне кажется "разделение времени процессора" это довольно устаревшая технология. Сейчас вроде как у процессора несколько ядер и они выполняют реально одновременно

Ответить

0

123 19 уровень, Киев

30 июля 2017, 10:34

У современного компьютера может быть несколько ядер, но их именно несколько, а количество потоков может исчисляться сотнями. Так что ядер будет всегда меньше, чем желаний у программиста.

Ответить

+3

Sam 18 уровень, Нижний Новгород

31 мая 2017, 13:32

фу, что за слово такое дурацкое "нить". Почему это "поток" неправильно называть? Все вокруг называют и всем всё ясно и смысл в этом есть. А вот поди кому скажи "нить"...

Ответить

+2

Aleksandr Cerleniuk 20 уровень

21 июня 2017, 20:00

Thread с английского нить, а поток уже и так есть - Input/Output Stream

Ответить

+2

Anton Stezhkin 19 уровень

26 июля 2017, 22:13

А Thread и Stream отличаются друг от друга?

Ответить

0

Akira Rokudo 20 уровень, Москва

6 августа 2017, 11:13

to Anton Stezhkin Конечно)
Потоки считывают\записывают данные,
Нити выполняют операции(да, в том числе они могут и записывать\считывать,но не только)

Ответить

+3

[Загрузить еще](#)

javarush.ru/, [G+ \(https://plus.google.com/114772402300089087607/\)](https://plus.google.com/114772402300089087607/), [Twitter \(https://twitter.com/javarush_ru\)](https://twitter.com/javarush_ru)



Программистами не рождаются
© 2018