yield — пропуск хода

Java Core (/quests/QUEST_JAVA_CORE)
7 уровень (/quests/lectures/?quest=QUEST_JAVA_CORE&level=7), 7 лекция (/quests/lectures/questcore.level07.lecture07)

ОТКРЫТА

— Привет, Амиго! У нас сегодня будет небольшой и интересный урок. Я расскажу тебе про yield – статический метод класса Thread.

Элли тебе уже рассказывала, что процессор постоянно переключается между нитями. Каждой нити выделяется небольшой кусочек процессорного времени, называемый квантом. Когда это время истекает – процессор переключается на другую нить и начинает выполнять ее команды. Вызов метода `Thread.yield()` позволяет досрочно завершить квант времени текущей нити или, другими словами, переключает процессор на следующую нить.

— А зачем нити может понадобиться уступить свое время другой нити?

— Необходимость в этом возникает не часто. Вызов yield приводит к тому, что «наша нить досрочно завершает ход», и что следующая за yield команда начнется с полного кванта времени. Значит шансы, что ее прервут – меньше. Особенно, если она небольшая (по времени). Такой подход можно использовать при оптимизации некоторых процессов.

Еще могу добавить, что метод Thread.sleep(0) работает фактически так же. Думаю, ты вначале будешь использовать метод yield не очень часто, но знать о нем — полезно.

‹ (/quests/lectures/questcore.level07.lecture06)

×17 › (/quests/lectures/questcore.level07.lecture08)

0 0 0 0 0 +8

Комментарии (11) популярные новые старые

Никита

Введите текст комментария

Aleksander 40 уровень 24 августа 2017, 18:23

"Вызов yield приводит к тому, что «наша нить досрочно завершает ход», и что следующая за yield команда начнется с полного кванта времени. Значит шансы, что ее прервут – меньше." - какой трешак то внутри одной нити думать о прерываемости остальных! Нас же учили, что нити - это классное упрощение, что мы пишем только логику одной нити, почти не думая о других. И тут - на!

Ответить 0

сергей климов 27 уровень 11 сентября 2017, 20:08

фраза "шансы, что ее прервут – меньше" относится к первой нити, у которой вызвали метод yield.
до истечения своего кванта времени она будет простаивать, не отпуская процессор к другой нити (если она быстро исполняет свой код).
и вот в это время ее могут прервать, хотя этого можно избежать, вызвав метод yield.
Не знаю, насколько понятно, но как-то так понимаю.

Ответить +8

Сергей Черник 33 уровень 29 сентября 2017, 18:50

Смотри.

```
1  class Threadishshe1 extends Thread {
2    void run() {
3      operation1();                  // Обычная комманда. Прерывай - не хочу.
4      //yield();                     // Если не раскомментить, господин CPU может пре
5      superUNLINKABLEoperation9000(); // СУПЕР НЕПРЕРЫВАЕМОЙ, ТОЧНОЙ, УБЕРИ РУКИ, ХРУПК
6    }
7  }
```

Ответить +30

Mr.Robot 24 уровень, Москва 12 октября 2017, 23:20

Вот это - реально понятно! Спасибо! Я (подозреваю, как и многие) не заакцентировался при чтении на слове "команда", считая, что речь идет о другой нити. И все думал, с чего бы нам думать о благополучии другой нити в нашей? В этом примере все стало ясно.

Ответить +2

Александр 18 уровень, Минск 26 ноября 2017, 22:54

Если на подходе у нас какой-то важный суко метод и мы боимся, что он вдруг прервется, то на всякий случай перед таким методом можно бахнуть yield) как-то так)

Ответить +2

Dudchenko Andrei 26 уровень, Киев 30 января, 23:16

Вообще, эта команда делает ровно наоборот - отдает процессорное время другим потокам. Непонятно, как вы ей собираетесь гарантировать непрерывность выполнения какого-то куска кода?

По ссылке можно почитать один из немногих реальных примеров использования этого подхода. https://stackoverflow.com/questions/6979796/what-are-the-main-uses-of-yield-and-how-does-it-differ-from-join-and-interr/24999481#24999481

Вкратце - yield используется для крайне узконаправленной оптимизации, когда по какой-то причине поток не может временно продолжать свою работу, но эта задержка крайне мала и в случае использования других способов (join, условные переходы и т.д.) падение производительности из-за возобновления работы потока с "нуля" будут выше, чем если использовать yield, когда поток остается в "горячем" состоянии и готов продолжить работать с лету,

Ответить +1

Vitya 33 уровень, Нижний Новгород 20 июня 2017, 13:47

yield можно использовать, если поток ждёт какого-то результата от других потоков. Ему нечего делать, например, пока что-то не произойдёт в другом потоке. Тогда незачем тратить процессорное время.

Ответить +5

Sam 18 уровень, Нижний Новгород 2 июня 2017, 16:08

Разделение времени происходит только если имеется всего одно ядро у процессора. Сейчас даже в

телефонах такое редко встретишь, а на серверах вовсе процессоров несколько. Сейчас потоки выполняются параллельно.
См. https://stackoverflow.com/questions/1897993/what-is-the-difference-between-concurrent-programming-and-parallel-programming

Так что yield будет использована только если программа работает на древней тачке

Ответить 0

Александр 20 уровень, Москва 5 июля 2017, 09:57

Очевидно же что количество потоков может быть несравнимо больше, чем ядер у процесоора. Так что yeld можно использовать для оптимизации.

Ответить +14

Kroll 36 уровень, Владивосток 18 марта 2017, 19:00

Чё-то типа return для циклов.

Ответить 0

Denis Beck 37 уровень, Екатеринбург 8 июня 2017, 13:31

может continue?

Ответить +5

ush.ru/) (https://plus.google.com/114772402300089087607) (https://twitter.com/javarush_ru)

JAVARUSH

Программистами не рождаются
© 2018