



Как работает множественный catch

Java Syntax (/quests/QUEST_JAVA_SYNTAX)
9 уровень (/quests/lectures/?quest=QUEST_JAVA_SYNTAX&level=9), 7 лекция (/quests/lectures/questsyntax.level09.lecture07)

ОТКРЫТА



— И ещё немного интересных уроков. Мне так нравится преподавать.

— Хочу рассказать тебе, как работает множественный catch. Все очень просто: при возникновении исключения в блоке try, выполнение программы передаётся на первый catch.

— Если тип, указанный внутри круглых скобок блока catch, совпадает с типом объекта-исключения, то начинается выполнение кода внутри блока {}. Иначе переходим к следующему catch. Там проверка повторяется.

— Если блоки catch закончились, а исключение так и не было перехвачено, то оно выбрасывается дальше, а текущий метод аварийно завершается.

— Ясно. Будет выполнен тот catch, тип в котором совпадает с типом исключения.

— Да, верно. Но в реальности все немного сложнее. Дело в том, что классы можно наследовать друг от друга. И если класс «Корова» унаследовать от класса «Животное», то объект типа «Корова» можно хранить не только в переменной типа «Корова», но и в переменной типа «Животное».

— И?

— Т.к. все исключения унаследованы от классов Exception или RuntimeException (который тоже унаследован от Exception), то их все можно перехватить командами catch (Exception e) или catch (RuntimeException e).

— И?

— Отсюда два вывода. Во-первых, с помощью команды catch(Exception e) можно перехватить любое исключение вообще. Во-вторых — порядок блоков catch имеет значение.

Примеры:

— Возникший при делении на 0 ArithmeticException будет перехвачен во втором catch.

Код

```
1 try
2 {
3     System.out.println("Before method1 calling.");
4     int a = 1 / 0;
5     System.out.println("After method1 calling. Never will be shown.");
6 }
7 catch (NullPointerException e)
8 {
9     System.out.println("Reference is null. Exception has been caught.");
10 }
11 catch (ArithmeticException e)
12 {
13     System.out.println("Division by zero. Exception has been caught.");
14 }
15 catch (Exception e)
16 {
17     System.out.println("Any other errors. Exception has been caught.");
18 }
```

— В примере ниже возникший `ArithmeticException` будет перехвачен в первом catch, т.к. классы всех исключений унаследованы от `Exception`. Т.е. `Exception` **захватывает любое исключение**.

Код

```
1 try
2 {
3     System.out.println("Before method1 calling.");
4     int a = 1/0;
5     System.out.println("After method1 calling. Never will be shown.");
6 }
7 catch (Exception e)
8 {
9     System.out.println("Any other errors. Exception has been caught.");
10 }
11 catch (NullPointerException e)
12 {
13     System.out.println("Reference is null. Exception has been caught.");
14 }
15 catch (ArithmeticException e)
16 {
17     System.out.println("Divided by zero. Exception has been caught.");
18 }
```

— В примере ниже исключение `ArithmeticException` не будет перехвачено, а будет выброшено дальше в вызывающий метод.

Код

```
1 try
2 {
3     System.out.println("Before method1 calling.");
4     int a = 1/0;
5     System.out.println("After method1 calling. Never will be shown.");
6 }
7 catch (NullPointerException e)
8 {
9     System.out.println("Reference is null. Exception has been caught.");
10 }
```

— Ну, вроде все понемногу проясняется. Непростая штука эти исключения.

— Это только кажется так. На самом деле – это чуть ли не самая простая вещь в Java.

— Не знаю, радоваться по этому поводу или огорчаться...

[< \(/quests/lectures/questsyntax.level09.lecture06\)](/quests/lectures/questsyntax.level09.lecture06)[×9 > \(/quests/lectures/questsyntax.level09.lecture08\)](/quests/lectures/questsyntax.level09.lecture08)



0



0



0



0



0

+44

Комментарии (29)

популярные

новые

старые

Никита

Vodevil 11 уровень, Киев

22 марта, 00:36



"— Это только кажется так. На самом деле — это чуть ли не самая простая вещь в Java."
Когда лектор спустил тебя с небес на землю, и намекает что дальше ты будешь страдать))

Ответить

0

Anastasiia 11 уровень

2 марта, 18:31



Ответить

0

Komers 19 уровень

5 октября 2017, 20:13



Я человек простой, вижу собачек ставлю 2 лайка,
вижу кошек отмечаю, как троллинг

Ответить

+17

theBaldSoprano 16 уровень, Санкт-Петербург

13 января, 18:08



Ответить

+2

Volodymyr 17 уровень

4 февраля, 23:19



Ответить

+8

Den 14 уровень, Одесса

2 марта, 16:41



Ответить

0

Евгений 15 уровень

2 сентября 2017, 23:29



Всё элементарно: сперва ловите тот exception который вы хотите поймать, но всегда завершайте общим exception е дабы не пропустить какое-то неведомое ранее исключение.

Ответить

+48

Hexogon 20 уровень

20 сентября 2017, 23:32



Ответить

+1

MrDudec 22 уровень

13 июля 2017, 18:41



Исключения гораздо проще, чем StackTrace. Когда робот плохо что-то понимает, то я понимаю это что-то отлично. А где роботу все дается просто, я мало, что понимаю

Ответить

+33

Радхараман Скороход 14 уровень

16 марта, 09:05



Скорее всего так специально сделано, чтобы в первом случае дать учащимся уверенности, а во втором показать, что не все так страшно)

Ответить

0

Vitaly 21 уровень, Москва

10 июля 2017, 11:34



Ответить

0

Dmitry Kaltovich 35 уровень, Минск

6 июля 2017, 16:31



Ответить

+3

Андрей Комратов 12 уровень, Москва

9 июня 2017, 16:28

То самое чувство когда умнее робота и схватываешь быстрее)

Ответить

+9

Vikentsi 22 уровень, Минск

21 июня 2017, 21:12

)) +

Ответить

0

Михаил 35 уровень, Санкт-Петербург

16 мая 2017, 14:00

Неожиданно, что 9 уровень гораздо проще предыдущих, видимо дальше ещё интересней!)

Ответить

+5

Vladislav 40 уровень, Москва

26 апреля 2017, 16:50

И снова не сказали, что сначала надо дочерние исключения обрабатывать, а после родительские.

Ответить

+5

Akira Rokudo 20 уровень, Москва

23 июня 2017, 11:36

"— В примере ниже возникший ArithmeticException будет перехвачен в первом catch, т.к. классы всех исключений унаследованы от Exception. Т.е. Exception захватывает любое исключение." are you kidding me?

Ответить

+2

Harvey 27 уровень, Минск

31 августа 2017, 23:09

Пока что темы наследования и полиморфизма в принципе не раскрывали, поэтому рано еще подробно об этом рассказывать

Ответить

+2

 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), [Facebook \(https://www.facebook.com/javarush\)](https://www.facebook.com/javarush)



Программистами не рождаются
© 2018