


Exceptions in threads

Theory

Practice

 100% completed, 0 problems solved ▼

Theory

⌚ 2 minutes reading

Unskip this topic

Start practicing

If one of the threads of your program throws an exception that is not caught by any method within the invocation stack, the thread will be terminated. If such an exception occurs in a single-threaded program, the entire program will stop, because the JVM terminates the running program as soon as there are no more **non-daemon** threads left.

Here is a tiny example:

```
1 public class SingleThreadProgram {
2     public static void main(String[] args) {
3         System.out.println(2 / 0);
4     }
5 }
```

This program outputs:



```
1 Exception in thread "main" java.lang.ArithmeticException: / by zero
2 at
org.example.multithreading.exceptions.SingleThreadProgram.main(SingleThreadProgram.java:6)
3
4 Process finished with exit code 1
```

The code `1` means the process was finished with an error.

If an error occurs inside a new thread we've created, the whole process will not be stopped:

```
1 public class ExceptionInThreadExample {
2     public static void main(String[] args) throws
InterruptedException {
3         Thread thread = new CustomThread();
4         thread.start();
5         thread.join(); // wait for thread with exception to
terminate
6         System.out.println("I am printed!"); // this line will be
printed
7     }
8 }
9
1
0 class CustomThread extends Thread {
1
1
1
2     @Override
1
3     public void run() {
1
4         System.out.println(2 / 0);
```

1 required topic

  Thread management ▼

6 dependent topics


-  Shared data ▼
- Sockets ▼
- Executors ▼
- Interruptions ▼
- States of a thread ▼
- Timers ▼

Table of contents:

- 1 Exceptions in threads
- Discussion

```
1  
5     }  
1  
6 }
```

Despite the uncaught exception, the program will be successfully completed.

```
1  Exception in thread "Thread-0" java.lang.ArithmeticException: / by  
zero at  
org.example.multithreading.exceptions.CustomThread.run(ExceptionInThreadE  
xample.java:15)  
2  I am printed!  
3  
4  Process finished with exit code 0
```

The code 0 means the process successfully finished.

Exceptions in different threads are handled independently. While the process has an alive non-daemon thread, it won't be stopped in case an uncaught exception occurs. Still the good practice is to handle exceptions in threads.

 Report a typo

351 users liked this piece of theory. 1 didn't like it. **What about you?**



Start practicing

Unskip this topic

[Comments \(7\)](#)

[Useful links \(0\)](#)

[Show discussion](#)



Tracks

Pricing

For organizations

About

Contribute

Careers

 Become beta tester

Be the first to see what's new

[Terms](#) [Support](#)



Made with  by Hyperskill and JetBrains