FREE eBook

LEARNING concurrency

Free unaffiliated eBook created from
Stack Overflow contributors.

#concurrenc

y

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: concurrency

It is an unofficial and free concurrency ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official concurrency.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with concurrency

Remarks

This section provides an overview of what concurrency is, and why a developer might want to use it.

It should also mention any large subjects within concurrency, and link out to the related topics. Since the Documentation for concurrency is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting concurrency set up or installed.

Example of concurrent execution in Java

```
import java.util.stream.IntStream;

public class Concurrent {
    public static void printAndWait(String s) {
        System.out.println(s);
        try {
            Thread.sleep(1000);
        } catch (Exception e) {}
    }

    public static void main(String[] args) {
        Thread myThread = new Thread() {
            public void run() {
                IntStream.range(1,32)
                .forEach(x -> printAndWait(""+x));
            }
         };
         myThread.start();
         IntStream.range('a', 'z').forEach(x -> printAndWait(""+(char)x));
     }
}
```

This will produce an output of something similar to

```
a
1
b
2
c
3
```

and so on, though results may vary. This is because the code in `myThread` is executed simultaneously, in a different thread, as the main flow. That is, the range 1-32 is handled by one thread, and the range a-z is handled by another.

Since there is no synchronization between the threads, there is no guarantee which one will execute first or indeed even that they will produce a result that is perfectly intertwined.

Read Getting started with concurrency online:
https://riptutorial.com/concurrency/topic/4563/getting-started-with-concurrency

Credits

S. No	Chapters	Contributors
1	Getting started with concurrency	Bex, Community