eBook Gratuit

APPRENEZ concurrency

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#concurrency

Table des matières

À propos

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

Chapitre 1: Démarrer avec la concurrence

Remarques

Cette section fournit une vue d'ensemble de la concurrence et de la raison pour laquelle un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans la concurrence, et établir un lien vers les sujets connexes. Étant donné que la documentation pour la simultanéité est nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Examples

Installation ou configuration

Instructions détaillées sur la configuration ou l'installation de la concurrence

Exemple d'exécution simultanée en 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));
     }
}
```

Cela produira une sortie de quelque chose de similaire à

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

et ainsi de suite, bien que les résultats puissent varier. C'est parce que le code dans `myThread` est exécuté simultanément, dans un thread différent, en tant que flux principal. C'est-à-dire que l'intervalle 1-32 est géré par un thread et que l'intervalle az est traité par un autre.

Comme il n'y a pas de synchronisation entre les threads, il n'y a aucune garantie que l'on s'exécutera en premier ou même qu'ils produiront un résultat parfaitement imbriqué.

Lire Démarrer avec la concurrence en ligne:
https://riptutorial.com/fr/concurrency/topic/4563/demarrer-avec-la-concurrence

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec la concurrence	Bex, Community