

Image Viewer Compulsory

I denne opgave arbejde vi meget med forståelsen og brugen af Threads i Java og JavaFX. I starten fokuserede vi mest på hvordan at en Thread er den mindste sekvens af instruktioner som bliver eksekveret. Vi gik hurtig videre til multi-threading hvor i vi lærte at dele den op i flere threads I den her fetch-execute cycle hvor processoren henter noget at den skal gøre for så bagefter at eksekvere det. Ved at dele en større udregnings proces op i flere Threads i den her cycle kan man optimere dens udregnings hastighed.

Threads

I selve Image Viewer projektet er der blevet taget meget brug af multi-threading i form af forskellige abstrakte klasser, interfaces og API 'er som. Der kan oprettes en ny thread for enhver operation og med brugen af Lambda-udtryk kan det gøres nemt og stream-lined. Sådan en thread kan eksekveres af en Executor som fås fra ExecutorService interfacet.

RGBCounter

I klassen RGBCounter bliver der implementeret et interface ved navn Callable som er et generic interface der kan bruges til multi-threaded tasks som skal returnere en værdi. Hvor imod sætning til interfacet Runnable som bare udfører arbejdet og kan ikke returnere nogen værdi. Callable bruges fordi der skal returneres en liste hvor i et billedes RGB-værdier er lagret.

SlideShow

SlideShow klassen nedarver fra java's Task. Task eksponerer yderligere state og observable properties der er brugbart til asynchronous tasks i JavaFX. Slideshow tråden bliver eksekveret i controlleren med et Lambda-udtryk.

ImageWithName

I denne klasse eksekverer RGBCounter via en ExecutorService, ExecutorServicesen har pool af threads som den kan genbruge og hvis alle threads er i brug venter fremtidige opgave i en queue.

Thread Safety

Thread Safety er der blev arbejdet med for at vise hvad der sker når et program ikke er Thread safe. Det der kan ske når et program har flere tråde som læser noget data fra den samme adresse i hukommelsen for så at override den data på selvsamme adresse, kan der ske noget der hedder concurrency fejl. En concurrency error er resultatet af forkert synkroniseringsbeskyttelse i software med flere tråde. For at undgå det kan man i stedet for at eksekvere sine tråde asynchronous eller asynkront kan man synkronisere sine tråde fra starten.

Thread Synchronization

I Image Vieweren var der et miniprojekt hvor den skulle kunne køre flere slideshows samtidig, hvilket kunne gøres ved at bruge en BlockingQueue hvor en thread queuer noget som skal eksekveres og 1 eller flere threads på den anden side eksekvere de queuede tasks. En anden måde er at bruge Synchronized keywordet og ExecutorService hvor i der er en limited pool af threads hvor i nye tasks bliver queued for eksekvering.

Link til GitHub

<https://github.com/WackyGlow/ImageViewer>