**Module 6**

Opdracht 3

Het toevoegen van edges aan een arraylist kan niet goed samen met het multithreaden. Dus je moet een synchronized list aanmaken zodat hij steeds opnieuw een nieuwe collectie aanmaakt met voldoende ruimte voor alle edges.

Opdracht 4

* Level 8: 19 mSec
* Level 9: 67 mSec
* Level 10: 266 mSec

**Module 7**

Opdracht 4 t/m 7

Deadlock: Deadlock is een situatie waar een actie moet wachten op een andere actie die op zijn plaats weer wacht op een andere etc. Hierdoor kan het programma in een oneindige loop terechtkomen en er nooit meer uit komen omdat de acties op elkaar wachten.

Livelock: Livelock betekent dat twee verschillende acties continue op elkaar reageren. Dus als er bijvoorbeeld twee personen tegenover elkaar staan en elkaar moeten passeren kan het voorkomen dat ze steeds op elkaar reageren. A gaat naar links zodat B erdoor kan terwijl B naar rechts gaat zodat A erdoor kan en dit blijft zich oneindig herhalen.

Starvation: Starvation betekent dat er een proces is wat niet de benodigde middelen kan krijgen of geen voorrang krijgt voor een lange tijd tijdens zijn executie. Als er bijvoorbeeld in een programma kleine processen voorrang krijgen van het grotere proces en er continue nieuwe kleine processen bijkomen kan het grote proces nooit uitgevoerd worden.

Module 8

1. S, Alle klasses hebben nu een verantwoordelijkheid in plaats van meerdere.

I, Er zijn nu verschillende interfaces in plaats van een algemene interface.

1. De MyTask class is een Observer.
2. Ik maak gebruik van een Fixed Thread Pool omdat er altijd 3 Threads zijn in deze applicatie.
3. De get() methode wacht tot de task done is met het retourneren van het resultaat. Anders kun je een half resultaat hebben wat niet af is.