

# Week 1: Strategy pattern

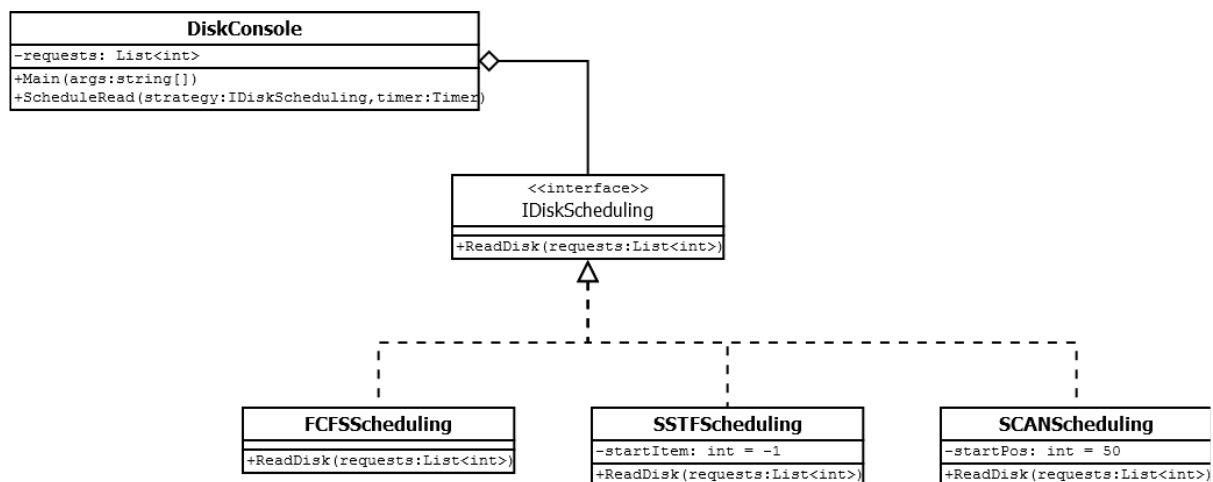
Minh-Triet Diep - Lars Jaeqx - T61

# Het patroon

We hebben een interface klasse IDiskScheduling die alle disk-read methodes implementeren. Hierdoor hoef je in de onderliggende klassen alleen de ReadDisk methode te implementeren.

In de DiskConsole wordt dan een instantie van een object gemaakt van een klasse (FCFSScheduling, SSTFScheduling of SCANScheduling) die IDiskScheduling implementeert. De keuze hiervoor is run-time in te stellen door het programma. Dit object wordt dan gebruikt om de ReadDisk methode aan te roepen die met het juiste algoritme leest.

## UML



# Test run results

Het cli-programma is simpel in gebruik. Eerst laat het zien welke disk scheduling opties gekozen kunnen worden, optie 1, 2 en 3:

1. First-come First-serve
2. Shortest seek time first
3. Scan

Waarna het programma wacht tot een valide input: 1, 2 of 3. Daarna wordt het uitgevoerd, de resterende requests worden na elke request geprint.

```
1: First-Come First-Serve
2: Shortest Seek Time First
3: SCAN
Choose scheduling strategy:
1
Using:
19 4 21 3 97 40 50 62 10 86 2
4 21 3 97 40 50 62 10 86 2
21 3 97 40 50 62 10 86 2
3 97 40 50 62 10 86 2
97 40 50 62 10 86 2
40 50 62 10 86 2
50 62 10 86 2
62 10 86 2
10 86 2
86 2
2
```

```
1: First-Come First-Serve
2: Shortest Seek Time First
3: SCAN
Choose scheduling strategy:
2
Using:
19 4 21 3 97 40 50 62 10 86 2
4 21 3 97 40 50 62 10 86 2
4 3 97 40 50 62 10 86 2
4 3 97 40 50 62 86 2
3 97 40 50 62 86 2
97 40 50 62 86 2
97 40 50 62 86
97 50 62 86
97 62 86
97 86
97
```

```

1: First-Come First-Serve
2: Shortest Seek Time First
3: SCAN
Choose scheduling strategy:
3
Using:
19 4 21 3 97 40 50 62 10 86 2
19 4 21 3 97 40 62 10 86 2
19 4 21 3 97 62 10 86 2
19 4 3 97 62 10 86 2
4 3 97 62 10 86 2
4 3 97 62 86 2
3 97 62 86 2
97 62 86 2
97 62 86
97 86
97

```

## Voor en nadelen

### Reusability

- + Als men de interface weet, kunnen de implementaties goed worden hergebruikt.
- + Tijdens runtime is te kiezen welke implementatie wordt gebruikt

### Maintainability

- + Elke implementatie is makkelijk te veranderen zonder dat andere implementaties er iets van merken
- + Makkelijk van implementatie te wisselen
- Wanneer de interface verandert, moeten alle implementaties mee veranderen

### Extensibility

- + Een extra implementatie is makkelijk toe te voegen
- Een extra implementatie is niet meteen duidelijk beschikbaar wanneer een developer er niet van weet.
- De developer moet extra moeite doen om nieuwe implementaties te vinden en te begrijpen

## Reflectie

Het implementeren van de design pattern was niet erg moeilijk. Het standaard-patroon was duidelijk waardoor het matchen van de diskscheduling implementaties met het patroon geen probleem was. Voor dit probleem is dit een prima patroon.