

# Interfaces

---

I denne øvelse skal I implementere 2 klasser - hhv. **ArrayStack** og **LinkedStack**, der hver især implementerer interfacet **IStack**. Der ser således ud:

```
interface IStack {  
    void push(String e); // Tilføjer et nyt element  
    String pop(); // Fjerne og returnere det første element  
    boolean isEmpty();  
    boolean isFull();  
    void show(); // Printer alle elementer  
}
```

**Du skal:**

- 1) oprette en klasse **ArrayStack**, som implementerer interfacet **IStack**. Brug et array til at gemme elementerne på stakken. Bemærk at et array har en fast størrelse, men at en stak ændrer størrelse - hvordan håndteres det? Hvordan holder man styr på hvor næste element skal indsættes?
- 2) Opret derefter en klasse **StackTest**, som opretter en stak (med din implementering), og gemmer følgende elementer på stakken: "Dette", "er", "en", "mærkelig", "sætning". Du skal sørge for at teste alle metoderne i denne delopgave. Metoden **show()** bør eks. skrive sætningen "sætning mærkelig en er Dette".
- 3) Opret derefter en klasse **LinkedStack** som også implementerer interfacet **IStack**. Denne gang skal der anvendes en hægtet liste til at gemme elementerne på stakken. En hægtet liste består af elementer, hvor hvert element kender det næste element i Gentag delopgave 2) men nu med klassen **LinkedStack**
- 4) Forsøg også gerne med andre elementer, end de anførte i 2)

## Stack

En stack er en klassisk datastruktur, som undertiden kaldes LIFO (last in first out). En stack er en af flere strukturer indenfor de Abstrakte Data Typer (ADT), disse indeholder, naturligt nok, data og nogle operationer, som er defineret til at arbejde på disse data.

**P.s. Det forventes IKKE, at metoden show() tømmer stakken. Stakken kan kun tømmes vha. Successive pop()**