

# Bounded Buffer

## Mission

At forstå og implementere en 'Bounded buffer' til at udveksle data mellem to eller flere tråde

## Background

C#Note Prog04 pp.1-21, samt slides [Concurrency.pdf](#)

Se evt: [Producer-consumer problem - Wikipedia](#)

## Set up scenario

### Opgave 1 – Lav 'forsøgsopstilling' Producer - Consumer

Forestil dig der er en eller flere producenter af 'item' (eller noget andet). Disse skal sendes videre til nogle konsumere det tager dem og udskriver dem.

Step 1: Lav en modelklasse Item med properties **Id**, **Value(int)**, sørg for at modelklassen selv generer et id, der gerne skal være unikt.

Step 2: Lav en klasse 'Experiment'

Step 3: i Klassen Experiment lav en metode Producer som med tilfældige mellemrum (Hint `Thread.Sleep(-- random.next(20) )`) laver et nyt Item med en tilfældig værdi (benyt `random...`) og indsætter i en kø (queue)

Step 4: i Klassen Experiment lav en metode Consumer som med tilfældige mellemrum (Hint `Thread.Sleep(-- random.next(10) )`) tager et Item fra køen og udskriver det.

Step 5: I klassen Experiment lav en metode `Start()`, som opretter en kø, samt starter 4 producenter i hver sin tråd, samt starter 2 konsumere i hver sin tråd.

Step 6: i Program.cs opret et objekt af Experiment, og kald `Start()`, hvad sker der?

## Nu skal du lave en Bounded Buffer som producere og konsumere kan benytte

### Opgave 2 – Design og implementer en BoundedBuffer

Du skal lave en ny klasse BoundedBuffer, gerne generisk, til at kommunikere Items med.

Step 1: Opret klassen BoundedBuffer

Step 2: Lav fire instans felter

- en kø ('buffer').
- en semaphor ('empty') til indikation af at køen er tom.
- en semaphor ('full') til indikation af at køen er fuld.
- en lås/lock ('lockObj') til at låse adgang.

Step 3: Lav en metode Insert (item)

- Den skal være låst (benyt lock), så kun en tråd kan være i metoden ad gangen
- Den skal sikre at bufferen ikke allerede er fuld (benyt full), indsæt i køen, og frigiv empty

Step 4: Lav en metode Take (item)

- Den skal være låst (benyt lock), så kun en tråd kan være i metoden ad gangen
- Den skal sikre at bufferen ikke er tom (benyt empty), tag et item fra køen, og frigiv full.

### Opgave 3 – Benyt Din BoundedBuffer i din forsøgsopstilling

I programmet fra opgave 1 udskift køen med din bounded buffer

Kør programmet, hvad sker der nu?

Hvad sker der hvis der starter 10 producere? Eller 10 konsumere?

Ekstra E1: I C# findes en datastruktur `ConcurrentQueue`, hvorfor kan du ikke bare benytte den i stedet for din BoundedBuffer?