

# Abstract Data Types forskellige collektions

Peter Levinsky, IT Roskilde

21.02.2023

# Abstrakte Data Typer (collections)

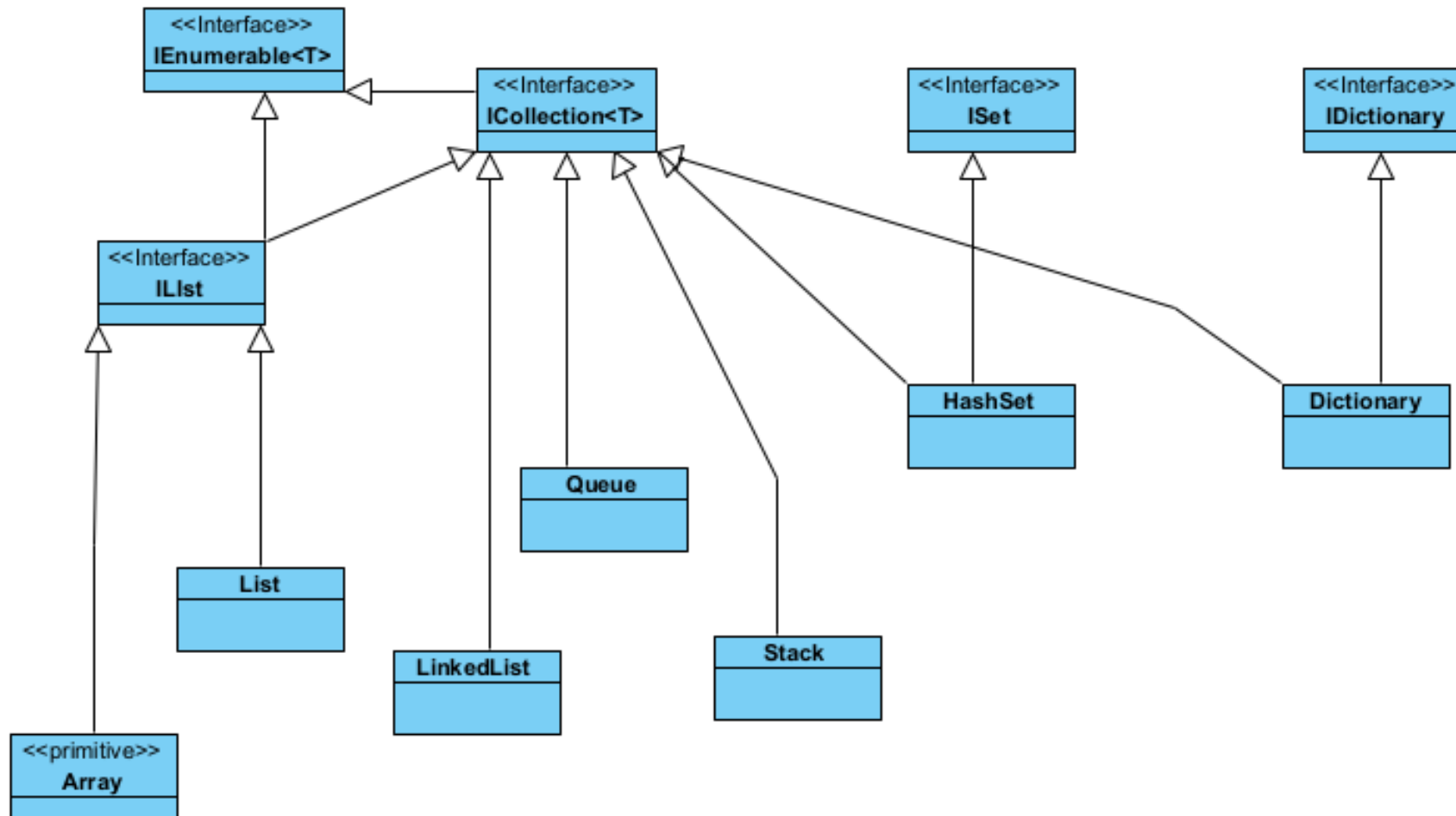
- Tidligere set:

- List
- Dictionary
- Array

- Flere:

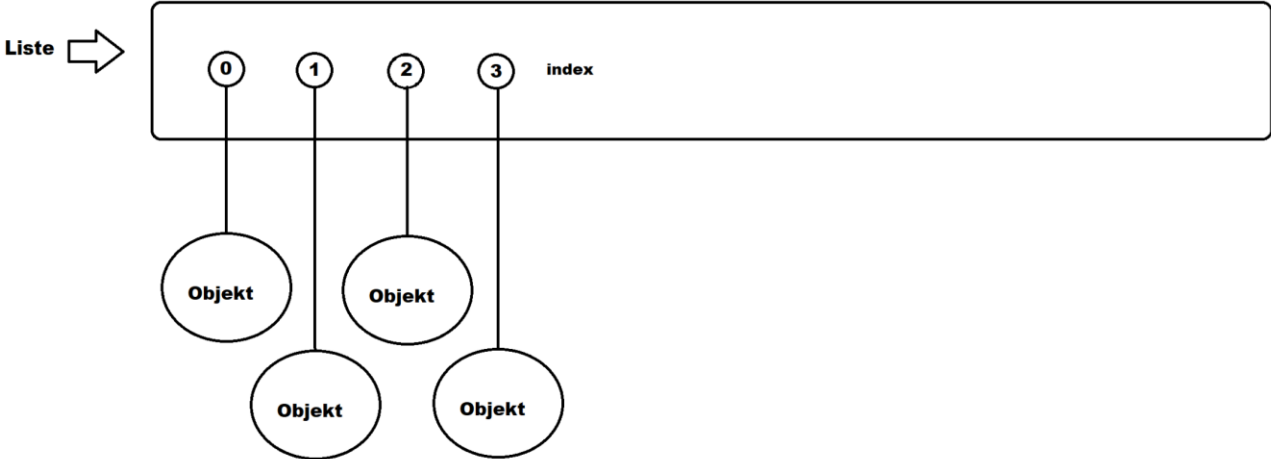
- LinkedList
- Queue
- Stack
- HashSet

# Lidt oversigt over Datastrukturer i C#



# Datastructure: List

Ide: Opbevarer objekter (simple typer) I en rækkefølge (array ligge begved)



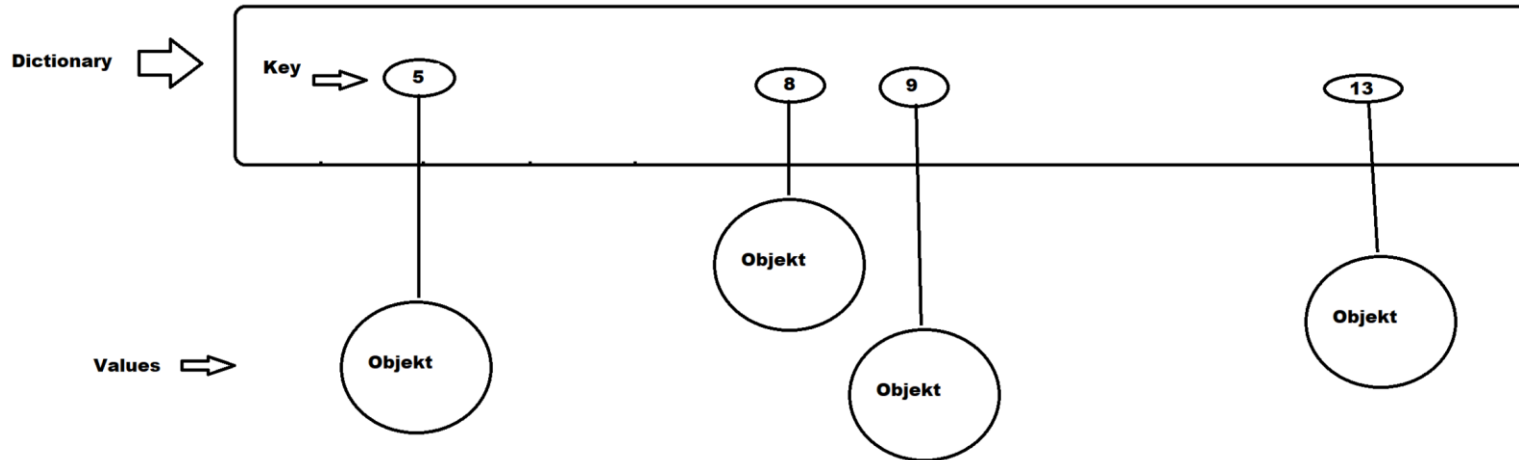
Brug:

```
List<Person> personer = new List<Person>();  
  
personer.Add(person); // tilføjer objekt  
Person p = personer[2]; // henter index 2 dvs plads 3  
  
foreach (Person p in personer){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=net-7.0>

# Datastructure: Dictionary

Ide: Opbevarer objekter (simple typer) I en slags ordbog men sorteret efter nøgler



Brug:

```
Dictionary<int, Person> personer = new Dictionary<int, Person>();
```

```
personer.Add(3, person); // tilføjer objekt til nøgle 3
```

```
Person p = personer[2]; // henter objekt til nøgle 2
```

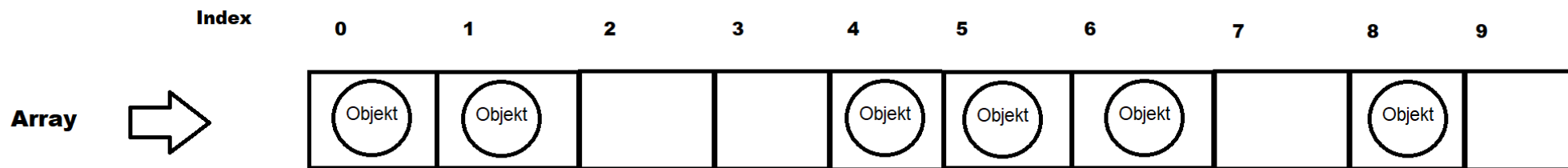
– keyNotFoundException hvis ingen

```
foreach (Person p in personer.Values){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2?view=net-7.0>

# Datastructure: Array

Ide: Opbevarer objekter (simple typer) I en buffer eller container – de behøver ikke at ligge samlet



Brug:

```
Person[] personer = new Person[10]; // der laves en buffer med plads til 10 objekter (0-9)
```

```
Personer[8] = person; // tilføjer objekt til index 8 (plads 9) programør skal selv styre index
```

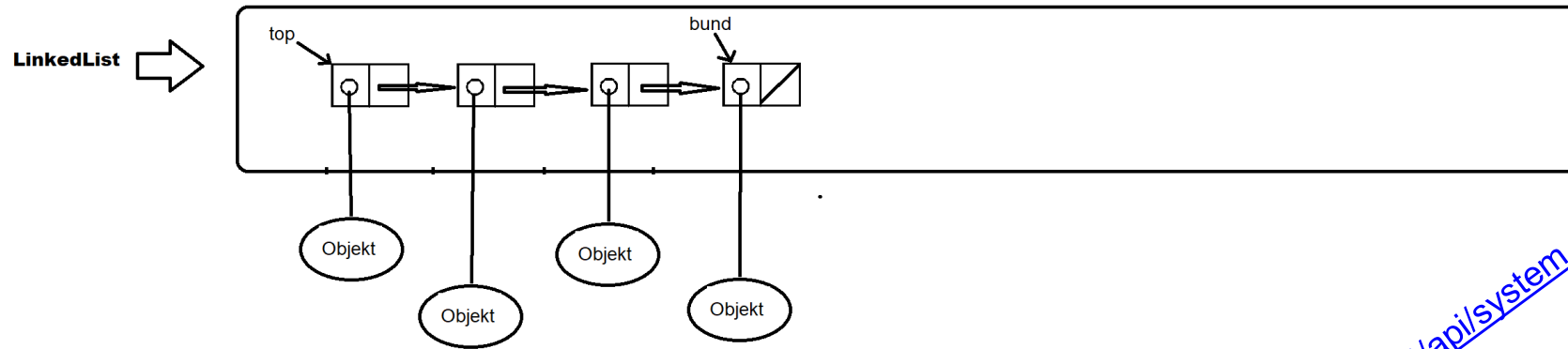
```
Person p = personer[1]; // henter index 1 dvs plads 2
```

```
foreach (Person p in personer){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/arrays>

# Datastructure: LinkedList

Ide: Opbevarer objekter (simple typer) I en rækkefølge i en dynamisk liste



Brug:

```
LinkedList<Person> personer = new LinkedList<Person>();
```

```
personer.Add(person); // tilføjer objekt
```

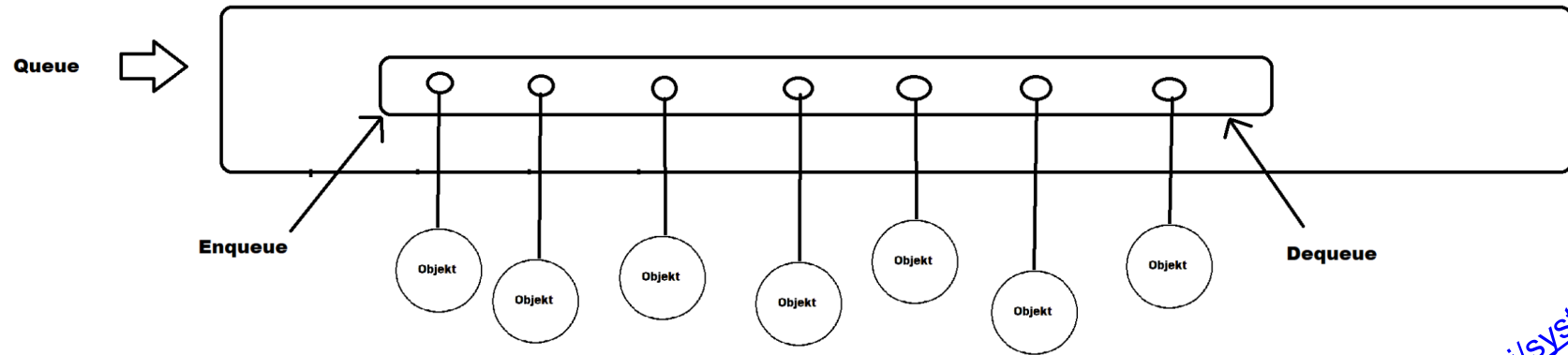
```
Person p = personer[2]; // henter index 2 dvs. plads 3
```

```
foreach (Person p in personer){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1?view=net-7.0>

# Datastructure: Queue

Ide: Opbevarer objekter (simple typer) I en kø dvs. først I køen først ud (alm kø – kultur)



Brug:

```
Queue<Person> personer = new Queue<Person>();
```

```
personer.Enqueue(person); // tilføjer objekt
```

```
Person p = personer.Dequeue() // henter den første I køen
```

```
-- InvalidOperationException hvis køen er tom
```

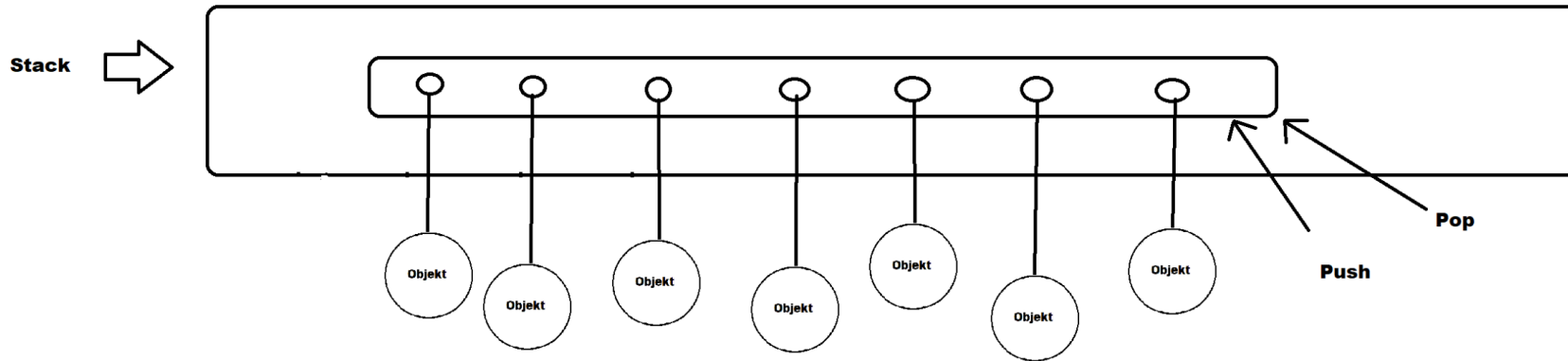
```
foreach (Person p in personer){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.queue?view=net-7.0>



# Datastructure: Stack

Ide: Opbevarer objekter (simple typer) I en stack dvs. først I stakken først ud (stabel tallerkener)



Brug:

```
Stack<Person> personer = new Stack<Person>();
```

```
personer.Push(person); // tilføjer object øverst på stakken
```

```
Person p = personer.Pop() // henter den øverste I stakken (nyeste)
```

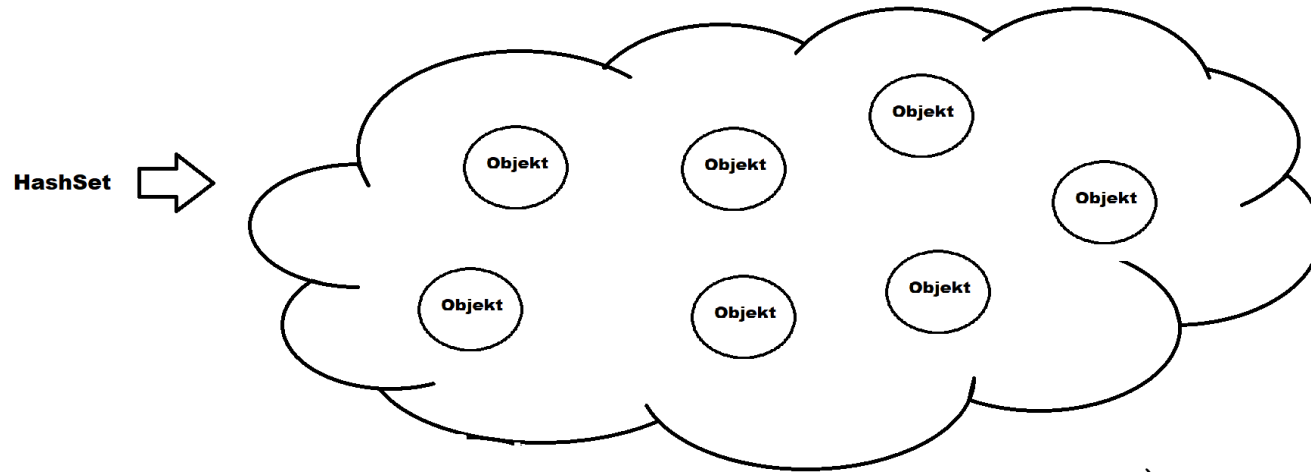
```
-- InvalidOperationException hvis stakken er tom
```

```
foreach (Person p in personer){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.stack?view=net-7.0>

# Datastructure: HashSet

Ide: Opbevarer objekter (simple typer) I en mængde ikke same rækkefølge og Objekt kun èn gang



Brug:

```
HashSet<Person> personer = new HashSet<Person>();
```

```
personer.Add(person); // tilføjer objekt
```

```
If (personer.TryGetValue(person, out helePerson) ) { ... }
```

```
// finds ikke direkte – men kan prøve at finde ud fra et objekt
```

```
foreach (Person p in personer){ ... } // gennemløb
```

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1?view=net-7.0>