



Køen

OOP / Queue

ZBC

Køen (C#)

FIFO, first-in-first-out, er en samling repræsenteret af den generiske Queue T klasse i .NET. Køer er samlinger, hvor et nyt element er placeret sidst i samlingen og fjernes sidst, når elementerne hentes.

Lad os sige, at du holder en fest, hvor du følger en køpolitik for dine gæster. Som tiden går, vil du gerne have dem alle til at forlade festen i samme rækkefølge som de kom.

Først terminologi - metoder:

- Placerer et element i køen hedder **Enqueue**
- Fjerner et første element fra køen **Dequeue**
- Se næste element i køen (uden at fjerne) **Peek**

Overvej følgende gæsteklasse:

```
public class Guest
{
    public string Name { get; set; }
    public int Age { get; set; }
}
```

Følgende kodeeksempel viser alle 3 metoder:

```
Queue<Guest> guests = new Queue<Guest>();
guests.Enqueue(new Guest() { Age = 25, Name = "John" });
guests.Enqueue(new Guest() { Age = 24, Name = "Barbara" });
guests.Enqueue(new Guest() { Age = 24, Name = "Phil" });
guests.Enqueue(new Guest() { Age = 23, Name = "Fred" });
guests.Enqueue(new Guest() { Age = 26, Name = "Hannah" });
guests.Enqueue(new Guest() { Age = 27, Name = "Cindy" });

Debug.WriteLine("Full guest list in the queue:");

foreach (Guest guest in guests)
{
    Debug.WriteLine("{0}, {1}", guest.Name, guest.Age);
}

Debug.WriteLine("");

Guest firstToLeave = guests.Dequeue();
Debug.WriteLine(string.Format("First to leave: {0}", firstToLeave.Name));

Debug.WriteLine("Guest list after the first goodbye:");
foreach (Guest guest in guests)
{
    Debug.WriteLine("{0}, {1}", guest.Name, guest.Age);
}

Guest soonToLeave = guests.Peek();
Debug.WriteLine(string.Format("Just checking who'll leave next: {0}", soonToLeave.Name));
```

Resultat:

Full guest list in the queue:

John, 25
Barbara, 24
Phil, 24

Fred, 23
Hannah, 26
Cindy, 27
First to leave: John
Guest list after the first goodbye:
Barbara, 24
Phil, 24
Fred, 23
Hannah, 26
Cindy, 27
Just checking who'll leave next: Barbara

Opgave Queue Operations:

- Sammen med din sidemand skal du lave en brainstorm, på jeres tavle, hvor man kan benytte en kø datastruktur.
- Vælg et af jeres emner og lav et program der illustrere emnet – programmet skal bestå af minimum to klasser en logik og en GUI
- Brugergrænsefladen skal ligne nedenstående og indeholde "menu-logikken" logik.

=====

H1 Queue Operations Menu

=====

1. Add items
2. Delete items
3. Show the number of items
4. Show min and max items
5. Find an item
6. Print all items
7. Exit

Enter your choice:1

Når ovenstående syv punkter er opfyldt kan der tilføjes lidt lækkert til brugergrænsefladen (for Camilla skyld) nedenstående er uforpligtende design der kan bruges til inspiration.

Søg efter ascii art.

