

DV1497: Inlämningsuppgift C

Generella krav:

Inga variabler får vara globala.

Alla funktioner som kan vara konstanta (const) ska vara det.

Lösningen får inte generera några minnesläckor.

Alla medlemsvariabler ska vara **privata**.

ADT Kö (Queue)

Uppgiften är implementera en klassmall Queue.

En kö följer principen FIFO (First In – First Out).

En dynamiskt allokerad array av typen T ska användas som intern datastruktur för kön.

Kön ska ärva från interfacet IQueue nedan – detta får inte modifieras:

```
#ifndef IQUEUE_H
#define IQUEUE_H

template <typename T>
class IQueue
{
public:
    virtual void enqueue(const T& element) = 0;
    virtual T dequeue() throw(...) = 0;
    virtual T front() const throw(...) = 0;
    virtual bool isEmpty() const = 0;
};
#endif
```

Beskrivning av medlemsfunktionerna:

- `enqueue(const T& element)`: lägger till element i kön. Eftersom kön ska kunna hantera många `enqueue(...)`- och `dequeue()`-anrop är det viktigt att Du tänker till så att implementationen blir minnesmässigt effektiv (Tips: tänk cirkulärt utnyttjande av arrayen!). Om arrayens kapacitet trots detta inte räcker till ska den expandera (även här får Du tänka till så att ordningen i kön inte ändras).
- `dequeue()`: tar bort och returnerar det element som är först i kön. Undantag ska hanteras.
- `front()`: returnerar det element som är först i kön. Undantag ska hanteras.

- `isEmpty()`: returnerar sant om kön är tom, annars falskt.

Utöver medlemsfunktionerna ska Du implementera konstruktör (vilken kan ta emot ett startvärde för kapaciteten), destruktör, kopieringskonstruktör och tilldelningsoperator.

Du ska även bifoga en testfil i vilken Du noggrant testat samtliga publika medlemmar i klassmallen `Queue`.

Klassmallen ska testas för 2 olika datatyper.

Tillse att inga minnesläckor genereras!

Denna fil ska vara väl kommenterad avseende vad som testas var i koden.