

OOP1

KU

24-03-2021

HEUTE

- Organisatorisches
- static Beispiel 10min
- Datenstruktur der Woche - Stack und Queue 10min
- Menti Fragen 10min
- (Tom's Array)++ 25min
 - Klassenaufteilung
 - Deep Copy
- Menti Fragen end

ORGANISATORISCHES ZU ASSIGNMENT 1

- Ausgabe morgen um 17:00 nach dem VO Stream
- KU-Fragestunde: 31.3.2021
- **Deadline: 7.4.2021**

STATIC BEISPIEL

DATENSTRUKTUR DER WOCHE

Stack - std::stack

<https://www.cplusplus.com/reference/stack/stack/>

Queue - std::queue

<https://www.cplusplus.com/reference/queue/queue/>

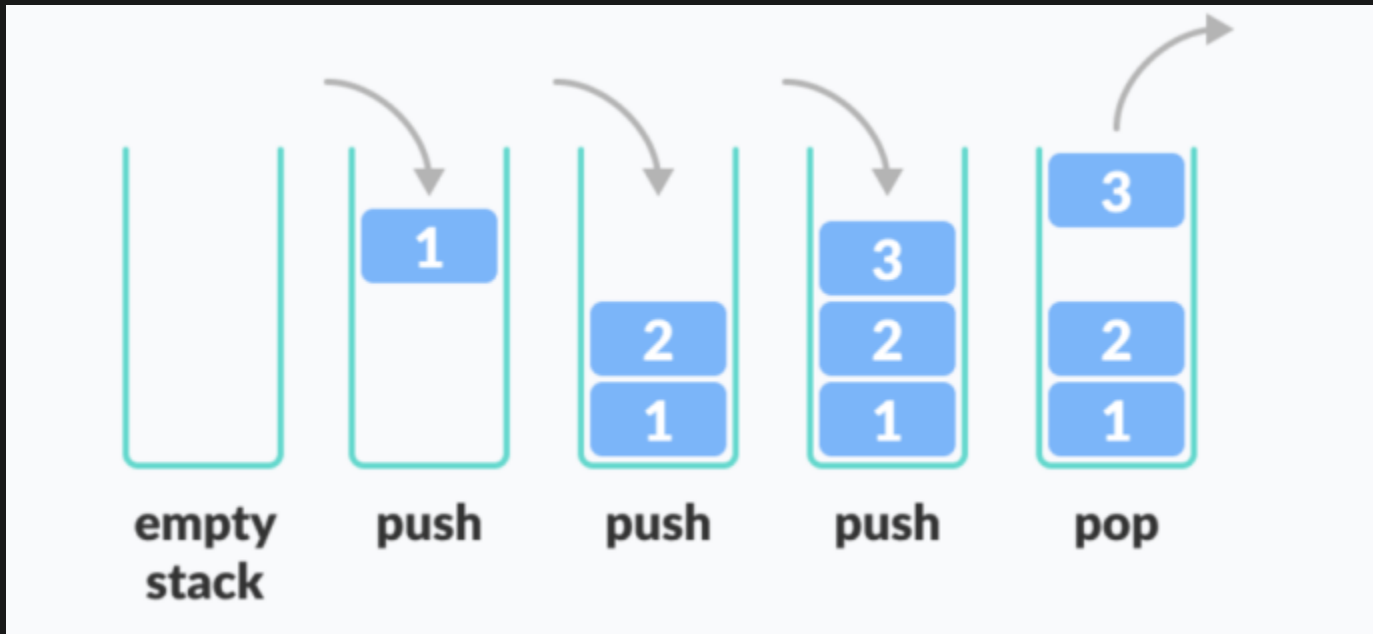
7	11	6	55	98	45	16	96	46
---	----	---	----	----	----	----	----	----

- empty()
- size()
- ...

STACK - STD::STACK

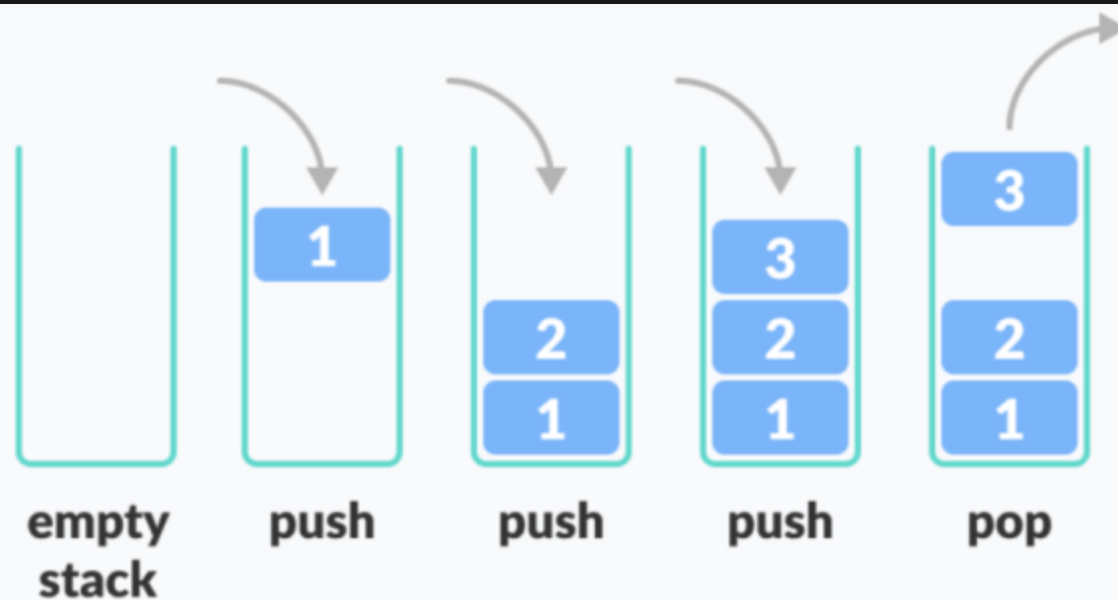
LIFO - last in first out

- top()
- pop()
- push()



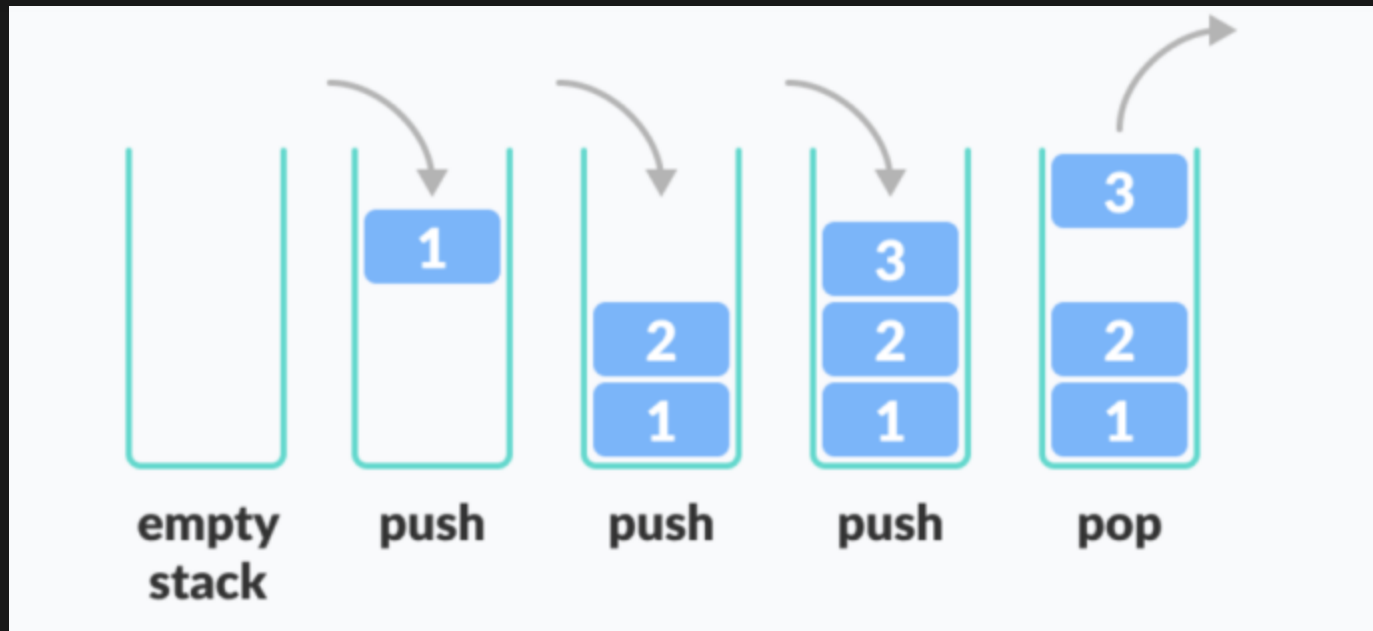
STACK - STD::STACK

```
1  std::stack<int> mystack; // erstellt einen stack
2
3  mystack.push(42); // fügt 42 am top of stack ein
4
5  // zugriff auf das oberste element
6  int number = myvector.top(); // 42
7
8  myvector.pop(); // entfernt das oberste element
```



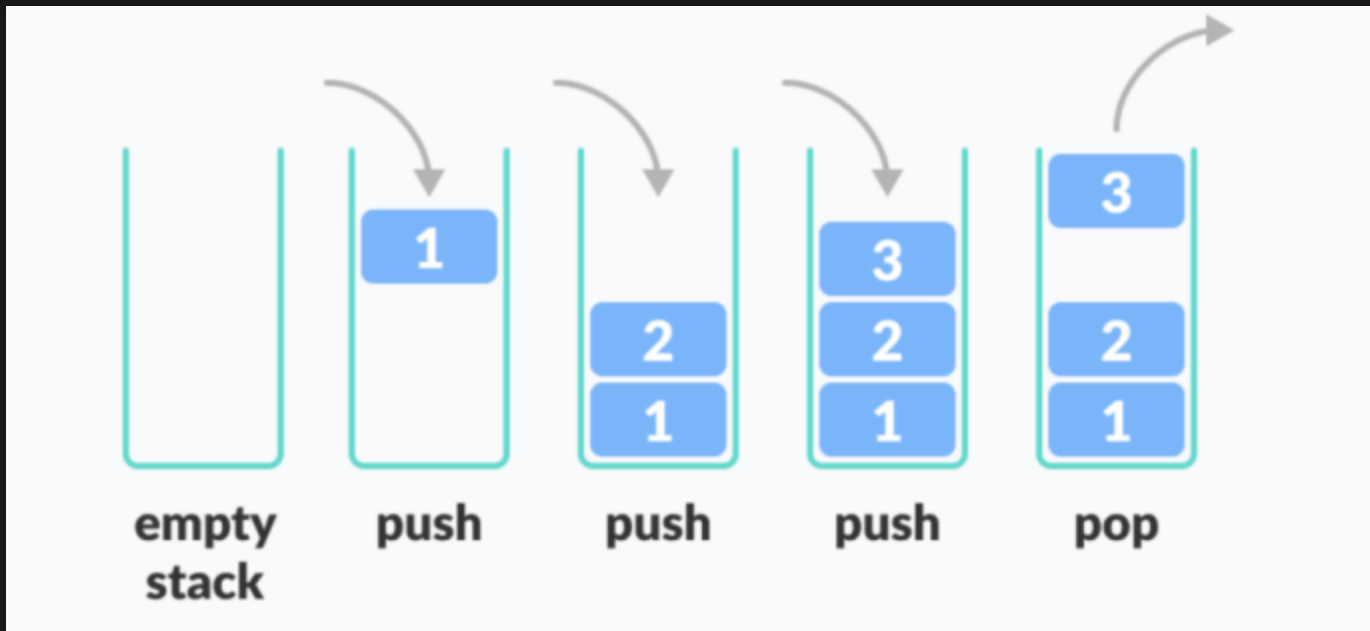
STACK - STD::STACK

```
1 std::stack<int> mystack; // erstellt einen stack
2
3 mystack.push(42); // fügt 42 am top of stack ein
4
5 // zugriff auf das oberste element
6 int number = myvector.top(); // 42
7
8 myvector.pop(); // entfernt das oberste element
```



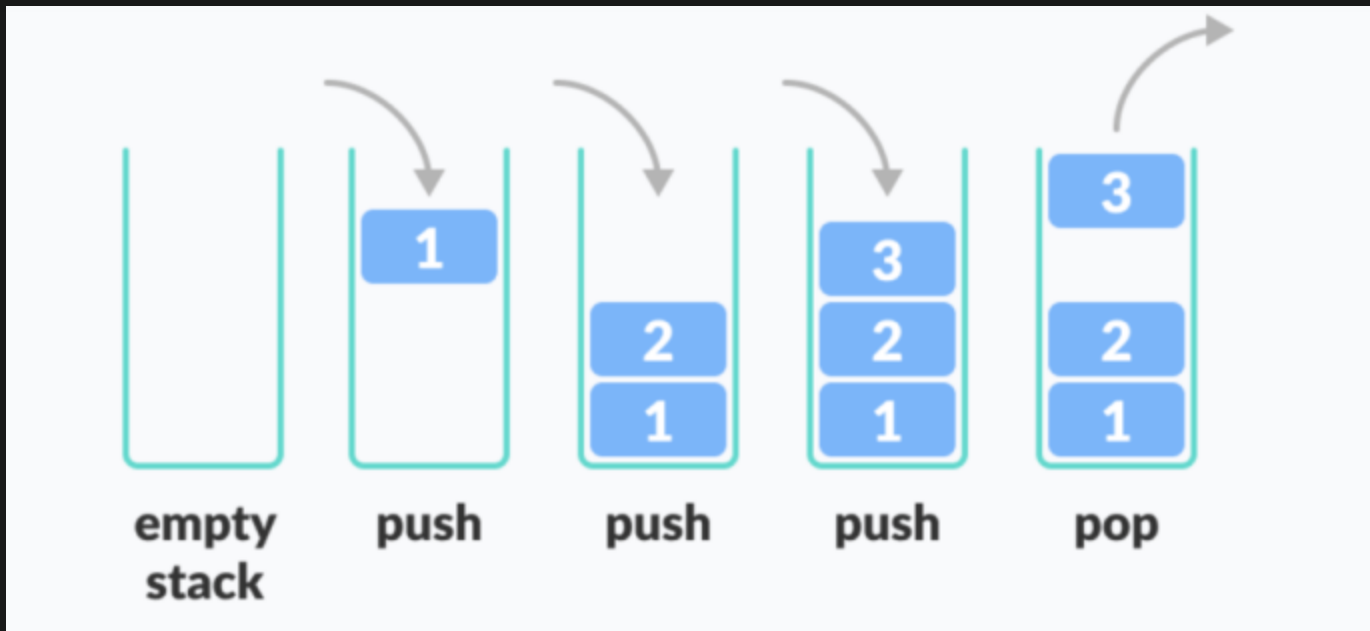
STACK - STD::STACK

```
1 std::stack<int> mystack; // erstellt einen stack
2
3 mystack.push(42); // fügt 42 am top of stack ein
4
5 // zugriff auf das oberste element
6 int number = myvector.top(); // 42
7
8 myvector.pop(); // entfernt das oberste element
```



STACK - STD::STACK

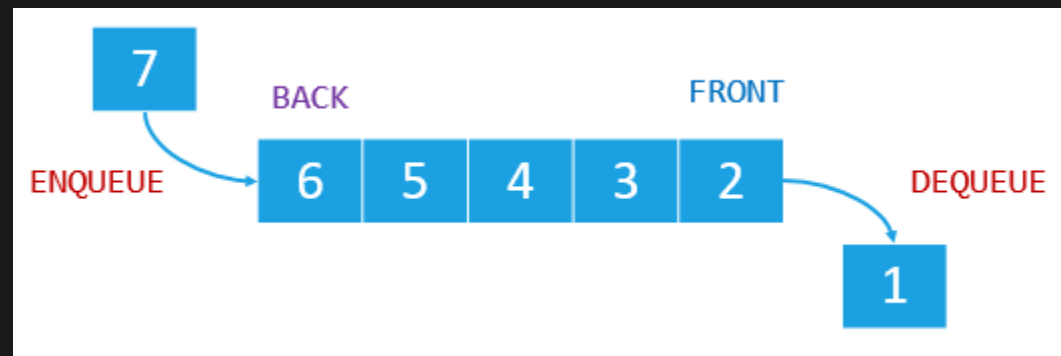
```
1 std::stack<int> mystack; // erstellt einen stack
2
3 mystack.push(42); // fügt 42 am top of stack ein
4
5 // zugriff auf das oberste element
6 int number = myvector.top(); // 42
7
8 myvector.pop(); // entfernt das oberste element
```



QUEUE - STD::QUEUE

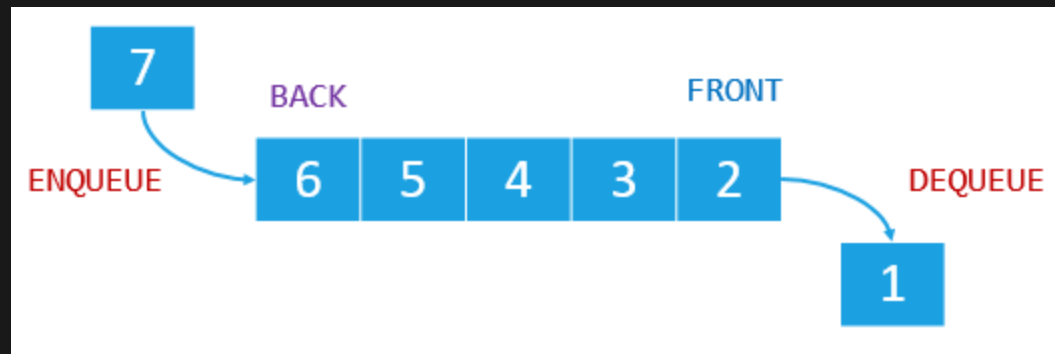
FIFO - first in first out

- front()
- back()
- pop()
- push()



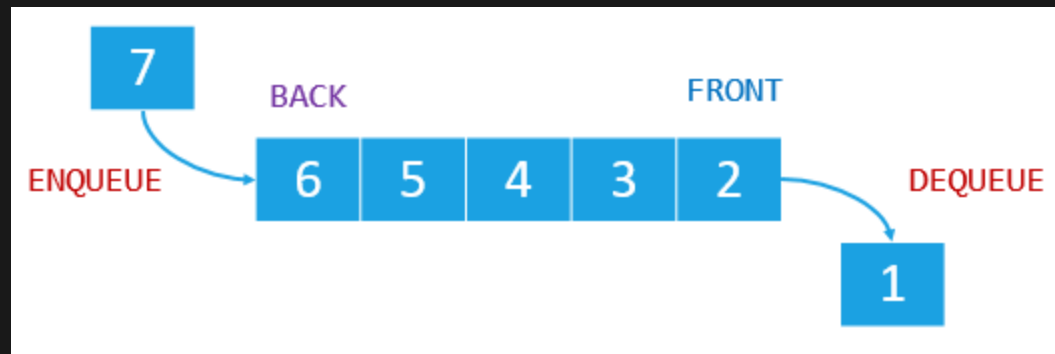
QUEUE - STD::QUEUE

```
1 std::queue<int> myqueue; // erstellt eine queue
2
3 // fügt 42,73 am hinten in die queue ein (enqueue)
4 myqueue.push(42);
5 myqueue.push(73);
6
7 // zugriff auf das erste/letzte element der queue
8 int front = myqueue.front(); // 42
9 int back = myqueue.back(); // 73
10
11 myqueue.pop(); // entfernt das erste element (dequeue)
```



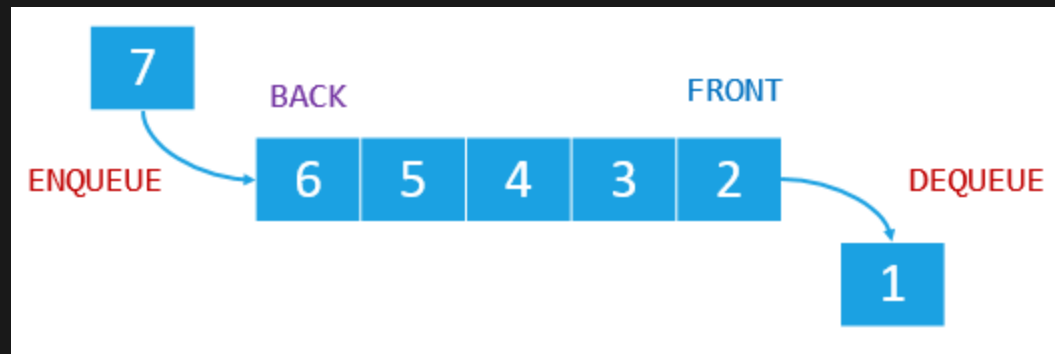
QUEUE - STD::QUEUE

```
1  std::queue<int> myqueue; // erstellt eine queue
2
3  // fügt 42,73 am hinten in die queue ein (enqueue)
4  myqueue.push(42);
5  myqueue.push(73);
6
7  // zugriff auf das erste/letzte element der queue
8  int front = myqueue.front(); // 42
9  int back = myqueue.back(); // 73
10
11 myqueue.pop(); // entfernt das erste element (dequeue)
```



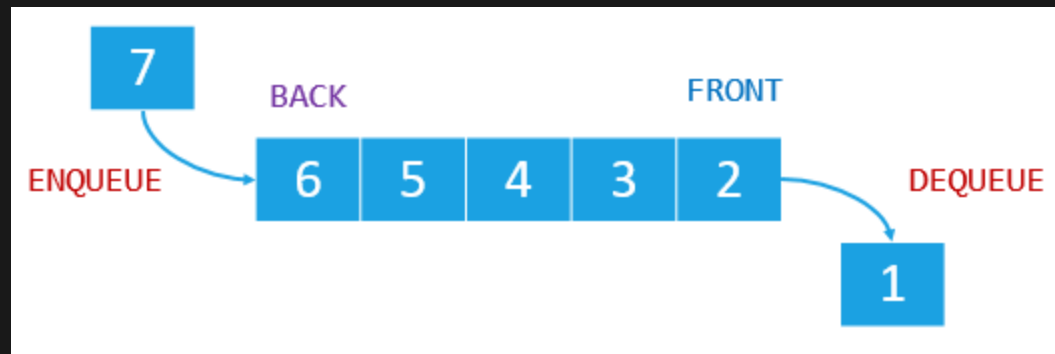
QUEUE - STD::QUEUE

```
1  std::queue<int> myqueue; // erstellt eine queue
2
3  // fügt 42,73 am hinten in die queue ein (enqueue)
4  myqueue.push(42);
5  myqueue.push(73);
6
7  // zugriff auf das erste/letzte element der queue
8  int front = myqueue.front(); // 42
9  int back = myqueue.back(); // 73
10
11 myqueue.pop(); // entfernt das erste element (dequeue)
```



QUEUE - STD::QUEUE

```
1  std::queue<int> myqueue; // erstellt eine queue
2
3  // fügt 42,73 am hinten in die queue ein (enqueue)
4  myqueue.push(42);
5  myqueue.push(73);
6
7  // zugriff auf das erste/letzte element der queue
8  int front = myqueue.front(); // 42
9  int back = myqueue.back(); // 73
10
11 myqueue.pop(); // entfernt das erste element (dequeue)
```



MENTI FRAGEN

(TOM'S ARRAY)++

ZUSAMMENFASSUNG

- static Beispiel
 - static in cpp initialisieren
 - inline
- Datenstruktur der Woche - Stack und Queue
- (Tom's Array)++
 - Klassenaufteilung
 - Deep Copy

MENTI FRAGEN