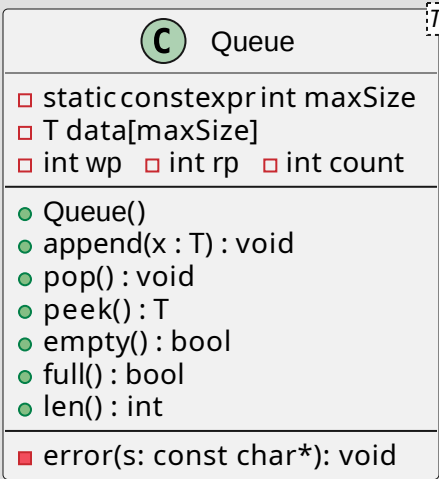


Übung zu Objektorientierter Programmierung

① Es sei die folgende Klassenkarte einer Warteschlangenklasse nach FIFO-Prinzip zu betrachten:



In der modernen Softwareentwicklung werden Tools eingesetzt, die aus UML-Klassenkarten C++-Klassen mit vordeklarierten Attributen und Methodenrumpfen erzeugen. Ein solcher Klassenrumpf sei für die FIFO-Warteschlange gem. Vorgaben zu vervollständigen:

```
template <typename T> class Queue {
private:
    static constexpr int maxSize = MAXSIZE;
    T data[maxSize];
    int wp;      // Write Pointer (WP)
    int rp;      // Read Pointer (RP)
    int count;   // Elementcount

    void error(const char* s) {
        std::cerr << s << std::endl;
        exit(1);
    }

public:
    Queue() : 

    void append(T x) {
        if (full()) { error("Queue is full!"); }
        
    }

    void pop() {
        if (empty()) { error("Queue is empty!"); }
        
    }

    T peek() {
        if (empty()) { error("Queue is empty!"); }
        
    }

    bool empty() {  }
    bool full() {  }
    int len() const {  }

    friend std::ostream& operator<<(std::ostream& os, const Queue<T>& q) {
        // „Queue to String“-Funktionalität die meist nicht in Klassenkarte gelistet wird.
        int idx = 0; // Fügen Sie mithilfe os << ■ Elemente den Ausgabestrom hinzu.
        
        return os; // Der Ausgabestrom soll dem Format ■, ■, ■ Format entsprechen.
    }
};
```