```
Queue (E, C): trait
  % FIFO operators
  includes Integer
  introduces
    empty: → C
    append: E, C → C
    count: E, C → Int
    __ ∈ __: E, C → Bool
    head: C → E
    tail: C → C
    len: C → Int
    isEmpty: C → Bool
  asserts
    C generated by empty, append
    ∀ q: C, e, e1: E
      count(e, empty) == 0;
      count(e, append(e1, q)) ==
        count(e, q) + (if e = e1 then 1 else 0);
      e ∈ q == count(e, q) > 0;
      head(append(e, q)) ==
        if q = empty then e else head(q);
      tail(append(e, q)) ==
        if q = empty then empty
        else append(e, tail(q));
      len(empty) == 0;
      len(append(e, q)) == len(q) + 1;
      isEmpty(q) == q = empty
  implies
    Container (append for insert)
    C partitioned by head, tail, isEmpty
    ∀ q: C
      len(q) ≥ 0
    converts head, tail, len
      exempting head(empty), tail(empty)
```