



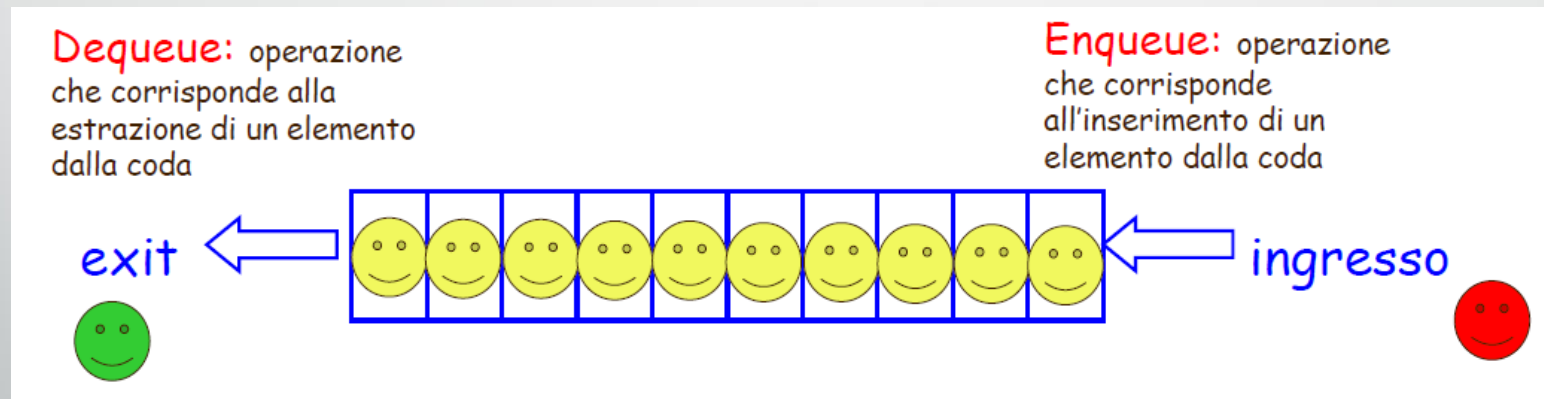
Java

Capitolo 6 – Code

Prof. Ivan Gentile

Coda (Queue)

- (leggi “chiù”) ADT fatto da sequenza di elementi che può essere “accorciata” da un lato e allungata da un altro lato:
- FIFO (First In First Out).



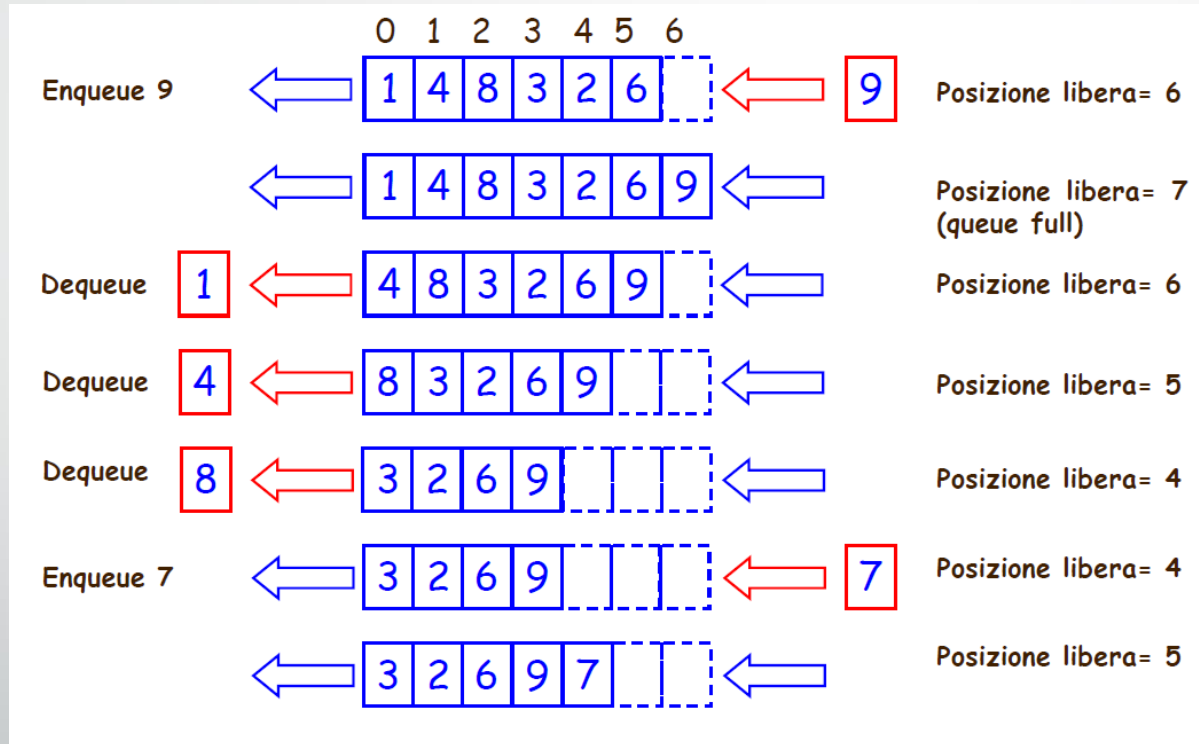
Operazioni

- inserire un elemento x in coda (**enqueue(x)**);
- togliere un elemento dalla coda (**dequeue()**);
- verificare se la coda è vuota (**isEmpty()**);
- cancellare tutti i dati (**clear()**);
- leggere (senza toglierlo dalla coda) il primo elemento in attesa (**readHead()**);
- nel caso in cui si preveda una coda con capacità massima limitata verificare se la coda ha raggiunto la sua massima capacità: (**isFull()**).

Implementazione con array lineare (1/2)

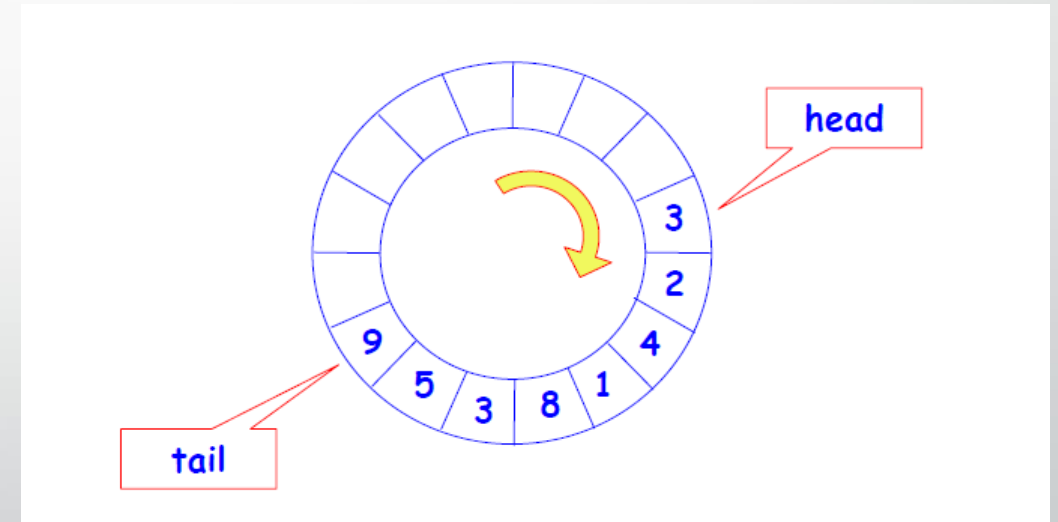
- Array di dimensioni fissate dove mettere in sequenza gli elementi della coda.
- L'estrazione dell'elemento di indice 0
- Una volta estratto tutti gli altri elementi successivi vengono "scivolati" avanti di un indice.
- L'inserimento avviene inserendo un elemento nel primo indice libero in fondo all'array.
- Bisogna mantenere in una variabile il valore di questo indice e aggiornarlo sia all'inserimento che alla estrazione
- Inefficiente!

Implementazione con array lineare (2/2)

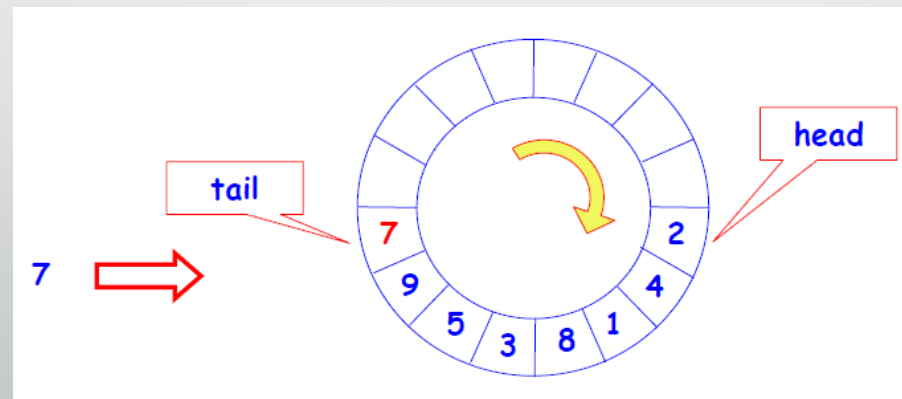
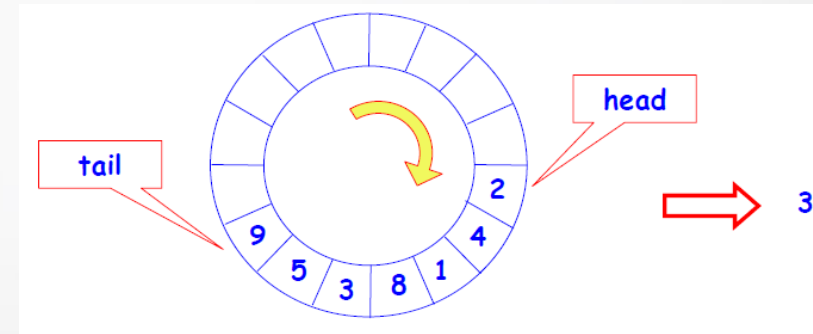
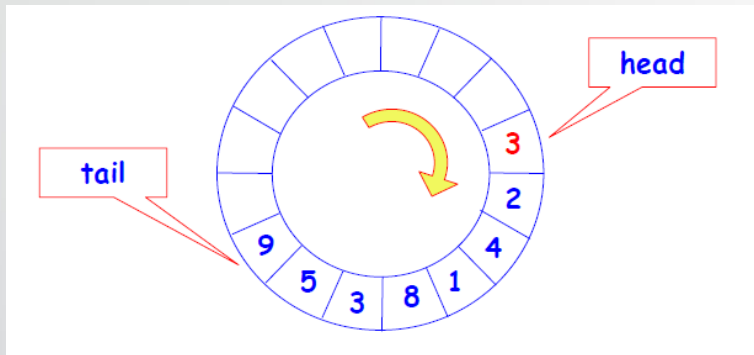


Implementazione con array circolare (1/3)

- Esistono due indici head e tail che indicano il primo elemento e l'ultimo.
- Non necessario effettuare shift per ogni inserimento
 - basta una sola assegnazione (più la modifica della variabile head).
- Ogni operazione di Enqueue o Dequeue comporta l'avanzamento di uno degli indici (tail per Enqueue, head per Dequeue).



Implementazione con array circolare (2/3)



Implementazione con array circolare (3/3)

head avanza
verso destra

