

# LISTE

---

Angelo Di Iorio

Università di Bologna

# Esercizio 1

- Implementare in Java una lista di oggetti generici, su cui sono possibili le seguenti operazioni:
  - Inizializzazione lista vuota
  - Inserimento di un elemento in testa alla lista
  - Rimozione di un elemento in testa alla lista
  - Inserimento di un elemento dopo l'elemento  $i$ -esimo ( $i=0 \rightarrow$  testa)
  - Rimozione dell'elemento  $i$ -esimo ( $i=1 \rightarrow$  testa)
- Per semplicità le operazioni di inserimento e rimozione ritornano `void` e non fanno nulla se i parametri non sono corretti (es. elemento  $i$ -esimo non esiste)

# Esercizio 1: una soluzione

- Realizzazione basata su una *lista monodirezionale (non circolare) senza sentinella*
- Definiamo una classe `MyNode<T>` che rappresenta un nodo in una lista e contiene un oggetto di tipo `T` e un puntatore al nodo successivo
- Definiamo la class `MyList<T>` che rappresenta la lista e contiene un puntatore al primo elemento della lista e un intero con il numero di elementi nella lista
  - `MyList<T>` implementa i metodi della slide precedente e sovrascrive il metodo `toString()` per stampare i valori nella lista
- Se le specifiche avessero incluso operazioni in coda? Lista bidirezionale?

## Esercizio 2

- Scrivere un algoritmo che, dato in input una lista realizzata monodirezionale, restituisce il  $k$ -esimo elemento a partire dalla fine.
- Ad esempio, se  $k = 1$  l'algoritmo deve restituire l'ultimo elemento; se  $k = 2$  il penultimo, e così via.

# Esercizio 3

- Scrivere un algoritmo che, dato in input una lista realizzata monodirezionale, restituisce il k-esimo elemento a partire dalla fine.
- Ad esempio, se  $k = 1$  l'algoritmo deve restituire l'ultimo elemento; se  $k = 2$  il penultimo, e così via.
- **La classe che implementa la lista NON ha un campo con la lunghezza della lista**
- **L'algoritmo DEVE effettuare una singola scansione della lista.**