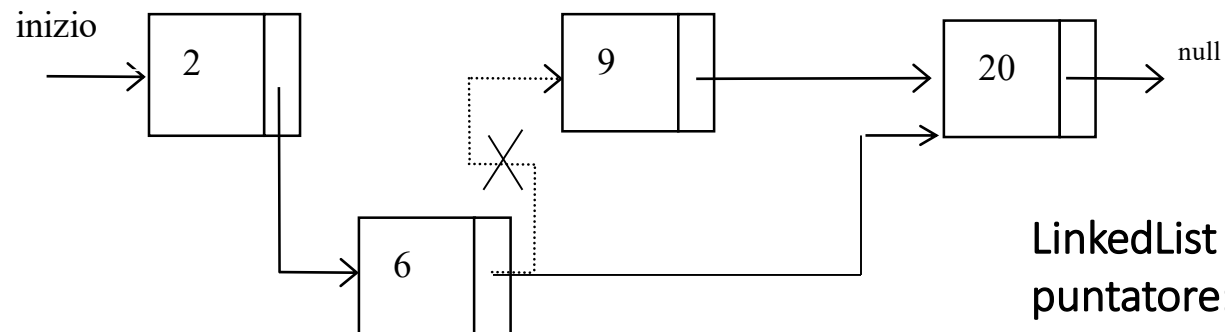
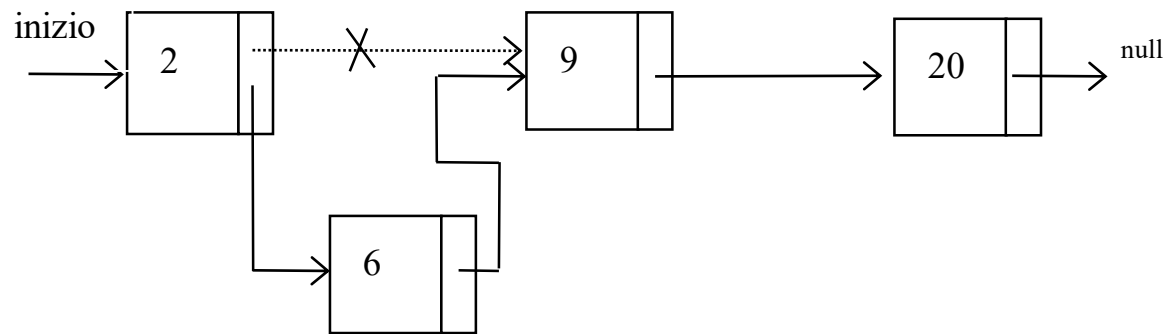
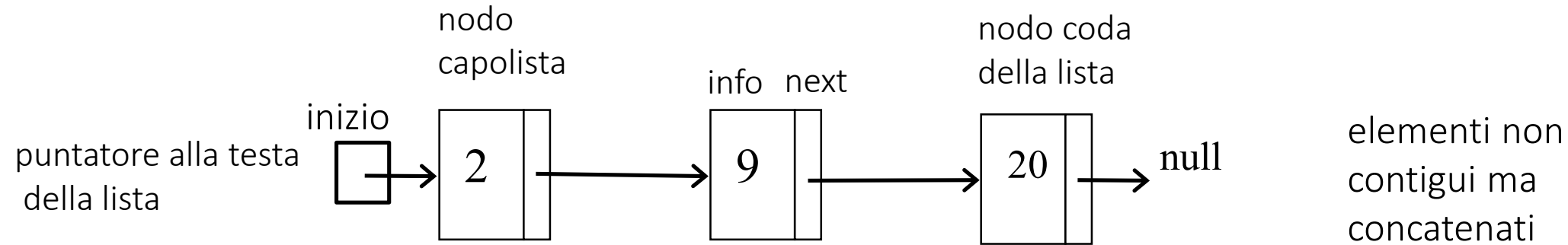


# Programmazione orientata agli Oggetti

*Concetti di lista concatenata a puntatori espliciti*

Libero Nigro

# Lista concatenata semplice



**LinkedList** usa in realtà uno schema a **doppio puntatore**: al successore e al predecessore, e permette anche la scansione backward

# Note

- Per gestire una lista concatenata a puntatori espliciti occorre
  - definire una classe per il generico nodo, che tipicamente contiene un campo informazione ed un campo puntatore (riferimento) al nodo successivo
  - dotarsi di una variabile che contenga il riferimento al primo elemento (primo nodo). Questo campo è stato denotato col nome «inizio». Si tratta del *puntatore alla testa* della lista
  - perchè la lista sia ben definita, occorre che l'ultimo nodo (estremo coda della lista) abbia come successore null
  - la lista è detta semplice, se in ogni nodo esiste solo un riferimento al nodo successore; si parla di lista doppia, quando ogni nodo contiene due puntatori: al nodo successore e al nodo predecessore. La LinkedList di Java è un esempio di lista concatenata doppia
- Vantaggi rispetto ad un array
  - non occorre dimensionare la lista
  - la lista si espande e si contrae come necessario, senza bisogno di eseguire spostamenti (shift) di elementi; un inserimento/rimozione in una posizione qualsiasi, si può portare a termine con un semplice aggiustamento di puntatori come indicato in figura
- Svantaggi rispetto ad un array
  - anche se gli elementi di una lista concatenate si succedono in ordine, es. dal minimo verso il massimo, la ricerca di un elemento è sempre **ricerca lineare** e mai può beneficiare dell'efficienza della ricerca binaria
  - è possibile recuperare la ricerca binaria se l'organizzazione concatenata dalla forma a lista si modifica alla forma albero binario (oggetto di successive lezioni).

# Note

- Per scandire una lista concatenata, occorre l'equivalente di un indice su array. È evidente che qui occorre una variabile riferimento, che possa spostarsi da un nodo al successivo, prendendo come valore il campo next del nodo corrente
- Una situazione particolare è quella della **lista vuota**, che si ha quando il puntatore di testa è null, es. **inizio==null**. Non si può seguire (dereferenziare) un puntatore nullo. Si solleva una NullPointerException in tutto simile alla IndexOutOfBoundsException degli array quando si cerca di selezionare un elemento con un indice errato (fuori da [0..array.length-1])
- Quando si inserisce un elemento in una lista vuota, il nuovo nodo va fatto puntatore dal puntatore di testa, ed il suo next deve valere null
- Anche una lista concatenata può disporre del suo iteratore, es. programmato con una inner class (come si vedrà a lezione)
- Sebbene il campo puntatore al nodo successivo, per convenzione, lo si indica spesso col nome next, attenzione a non confondere le cose con l'iteratore. L'iteratore ha il metodo next() che ritorna un elemento, ossia l'informazione di un nodo. Nulla a che vedere con il campo puntatore next.