

LISTE A PUNTORI INSERIMENTO IN CODA.

Riprendiamo le considerazioni degli appunti precedenti riferiti all'esempio pilota sull'inserimento in coda.

Se testa è null

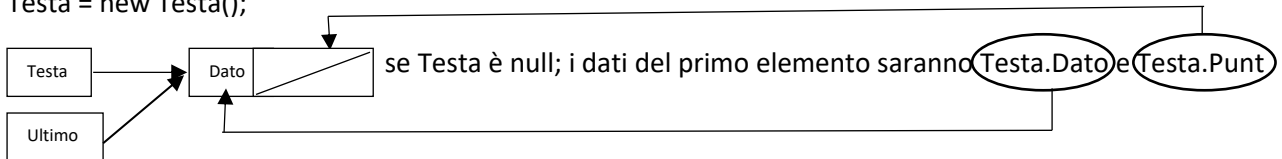
Nodo Testa;

Testa è stato dichiarato l'oggetto Testa che conterrà l'indirizzo dell'area di memoria.

Nodo Ultimo;



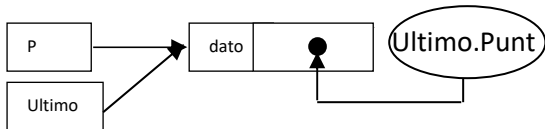
Testa = new Testa();



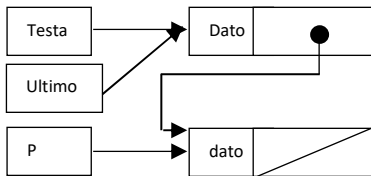
Ultimo = Testa;

altrimenti

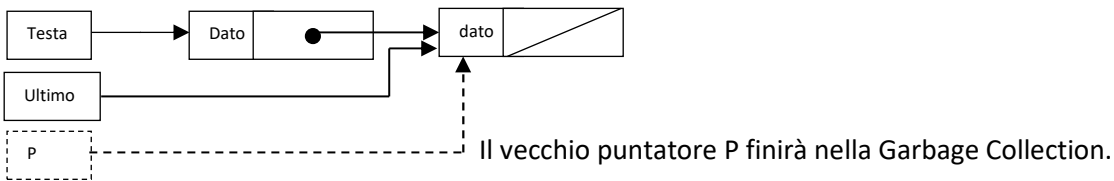
Nodo P = new Nodo();



Ultimo.Punt = P;

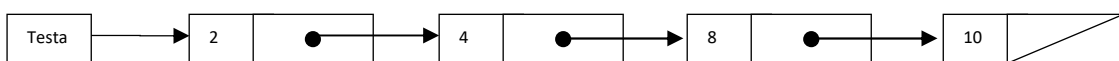


Ultimo = P;



INSERIMENTO DI UN ELEMENTO IN UNA POSIZIONE INTERMEDIA.

Supponiamo di voler inserire un nuovo elemento in una posizione che non sia ne in testa ne in coda ma in una posizione intermedia:

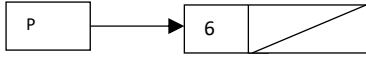


supponiamo di voler inserire un elemento che deve essere inserito tra gli elementi che contengono i dati 4 e 8 rispettivamente.

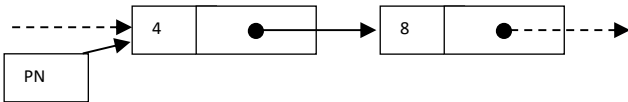
Il primo passo è quello di posizionarsi all'elemento precedente all'elemento che vogliamo inserire. Nel nostro caso l'elemento contenente il dato 4. Quindi creiamo il nuovo elemento:

Nodo P = new Nodo ();

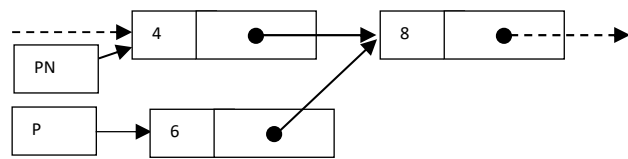
P.Dato = 6;



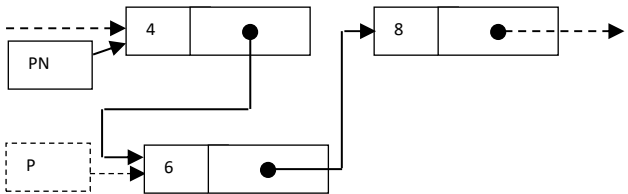
Supponiamo che il puntatore PN sua puntato all'elemento che contiene il dato 4:



P.Punt = PN.Punt;

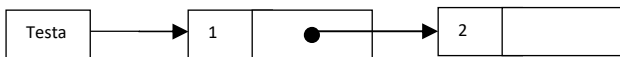


PN.Punt = P;

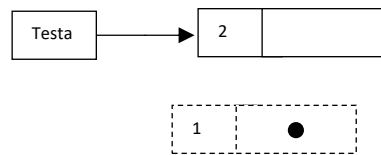


ELIMINAZIONE IN TESTA, CODA ED IN UNA POSIZIONE INTERMEDIA DI UN ELEMENTO.

L'eliminazione di un elemeto di testa è semplice:



Testa = Testa.Punt;



In coda (o l'eliminazione degli elementi da un certo punto in poi):

supponiamo che PN sia il puntatore che punta all'ultimo elemento (o all'elemento da cui vogliamo eliminare la lista) basta;

PN.Punt = null;

Per eliminare un elemento intermedio:

se PN è il puntatore all'elemento precedente a quello che vogliamo eliminare, basta porre:

PN.Punt = PN.Punt.Punt;