



INFORMATICA

ITT – Indirizzo Informatica

Classe 4^a Info

Strutture dati dinamiche:

- Lista, pila e coda

Rif. libro di testo:

- Unità 3, Lez. 1, 3, 4 (pag. 115-133)

Prof. Davide FAZZONE

Argomenti della lezione

01

Lista, pila e coda



02

Gestione di una pila



03

Gestione di una coda





Argomenti della lezione

01

Lista, pila e coda





Classificazione strutture dati

Posso **classificare** le strutture dati in:

- **Statiche**, se il numero dei loro elementi è costante durante l'esecuzione del programma
- **Dinamiche**, se il numero dei loro elementi varia durante l'esecuzione del programma

Principali strutture dati **statiche**: variabili, vettori, matrici, tabelle.



Classificazione strutture dati

Strutture dati **statiche** sono **inefficienti** per gestire grandi quantità di dati con elevato numero di inserimenti e cancellazioni:

- devo conoscere a priori il **numero di elementi** (scarsa flessibilità)
- **inserimento** di un elemento ordinato richiede lo spostamento in avanti di tutti gli elementi che lo seguono
- **eliminazione** richiede lo spostamento indietro di tutti gli elementi che seguono quello cancellato



Classificazione strutture dati

Strutture dati **dinamiche** occupano solo la memoria che di volta in volta è necessaria.

Possibile strutture dati dinamiche:

- **Lista**
- **Pila**
- **Coda**



Lista

Per **LISTA** si intende un insieme di elementi:

- collegati tra loro da un rapporto di **sequenzialità**
- cui si può accedere solo in **maniera sequenziale** (partendo dal primo fino all'ultimo)

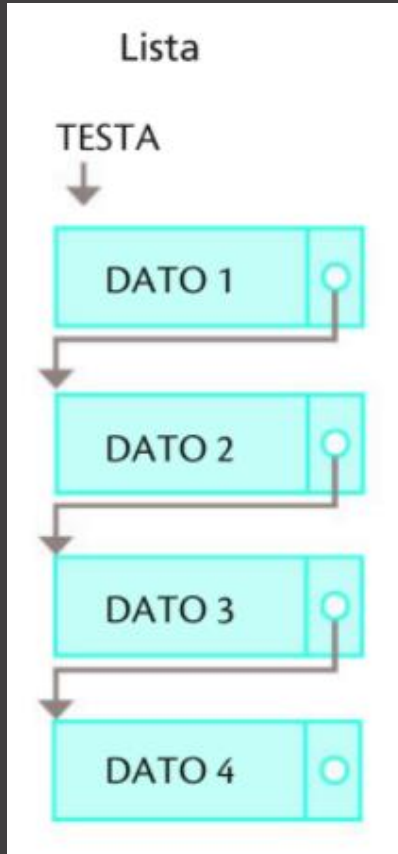


Lista

Ogni elemento della **LISTA** è composto da due parti:

- una con il **dato** (o i dati)
- una con l'indicazione (**link**) dell'elemento successivo

Lista



Il primo elemento NON ha
predecessori

L'ultimo elemento non ha
successori

Inserimento in una lista

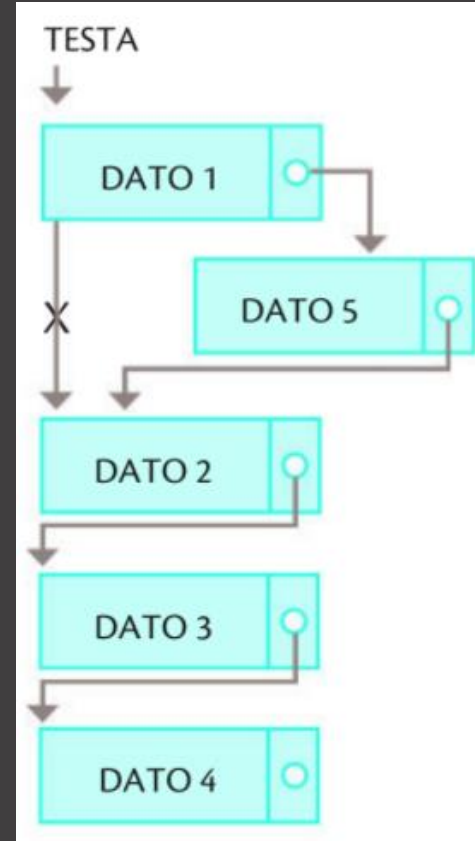
Possibile **inserire** un nuovo elemento in **qualunque posizione** all'interno della lista, modificando in modo opportuno i collegamenti (link)



Inserimento in una lista

Esempio

Inserisco DATO5 tra
DATO1 e DATO 2





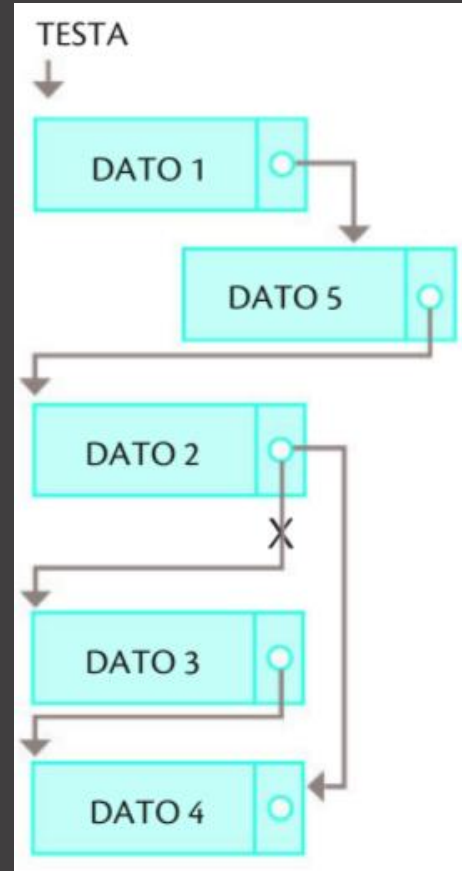
Eliminazione da una lista

Possibile **eliminare** un elemento da una lista, assegnando al link dell'elemento che lo precede l'indirizzo di quello che lo segue.

Eliminazione da una lista

Esempio

Elimino DATO3.



Pila e Coda

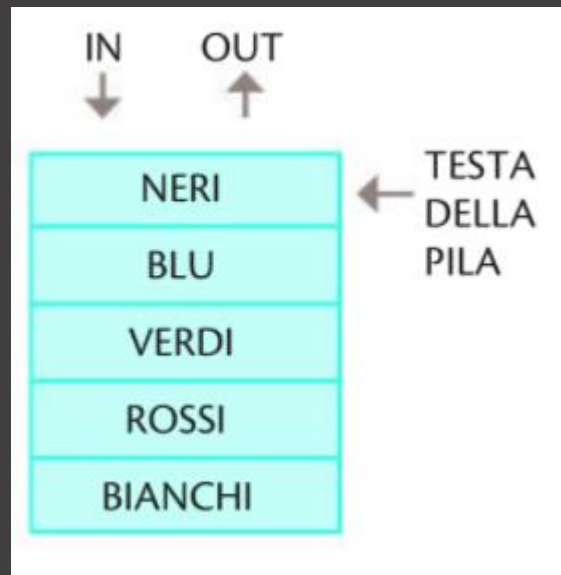
Esistono due tipi di liste: la **pila** e la **coda**



Pila

La **pila** (**stack**) è una lista gestita con il metodo **LIFO** (**Last In First Out**)

Sia l'estrazione che l'inserimento degli elementi avvengono solo da un lato
→ Importante gestire la **testa della pila**



Coda

La **coda** è una lista gestita con il metodo **FIFO** (**First In First Out**)

Si estrae sempre l'elemento in **testa alla coda**, si inserisce sempre un elemento **alla fine della coda**

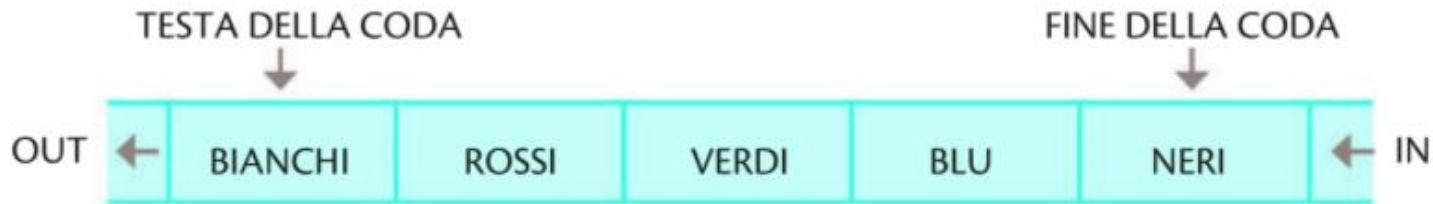


Figura 4 Coda



Esempi di pila e di coda

Esempio di **pila**: le lettere inserite in una cestello e prese una per volta a partire da quella più in alto

Esempio di **coda**: le persone in coda ad un ufficio postale

Argomenti della lezione

02

Gestione di una pila



Gestione di una PILA

Esistono linguaggi (es. C, C#, Java...) che contengono "*primitive*" adatte a gestire e definire strutture dati dinamiche tramite l'uso di **puntatori**

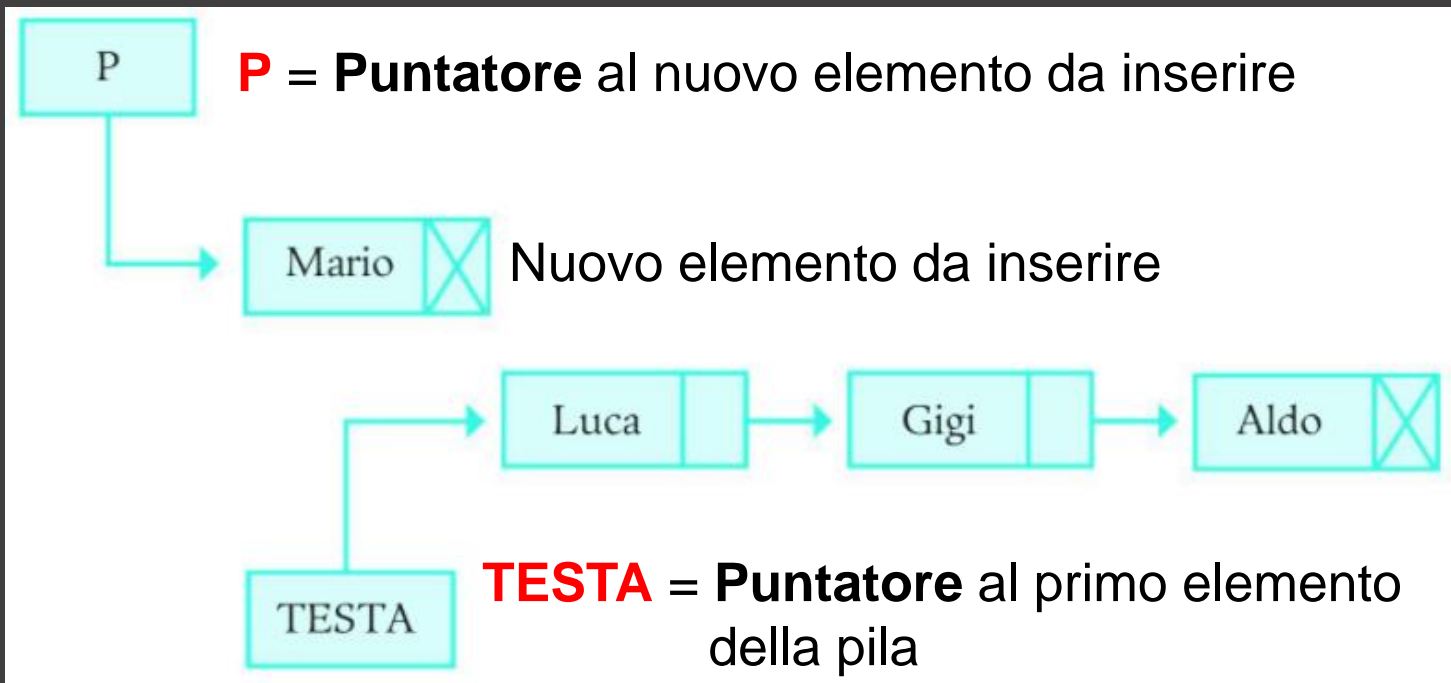
Un elemento della **pila** è un **record** o un **oggetto** formato dai **dati** (uno o più campi) e un campo **puntatore** all'elemento successivo



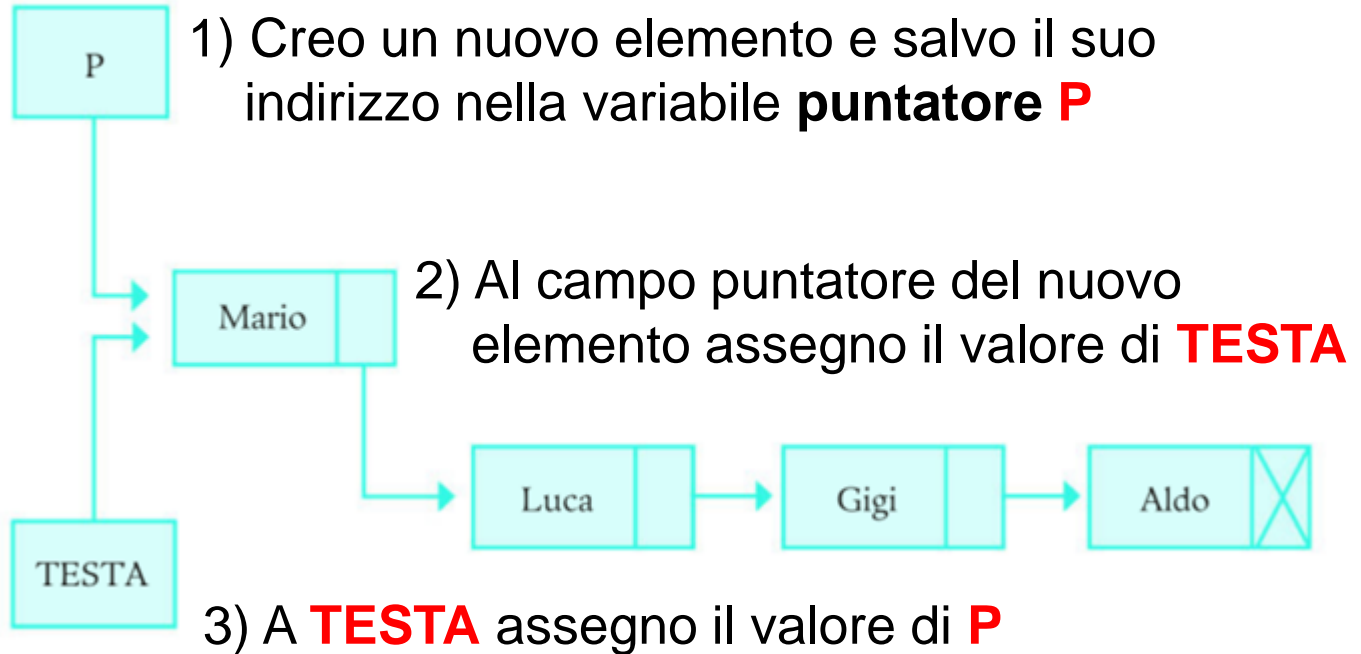
Puntatore

Inserimento (PUSH) in una PILA

L'operazione di inserimento di un nuovo elemento in una **pila** si definisce **PUSH**



Inserimento (PUSH) in una PILA



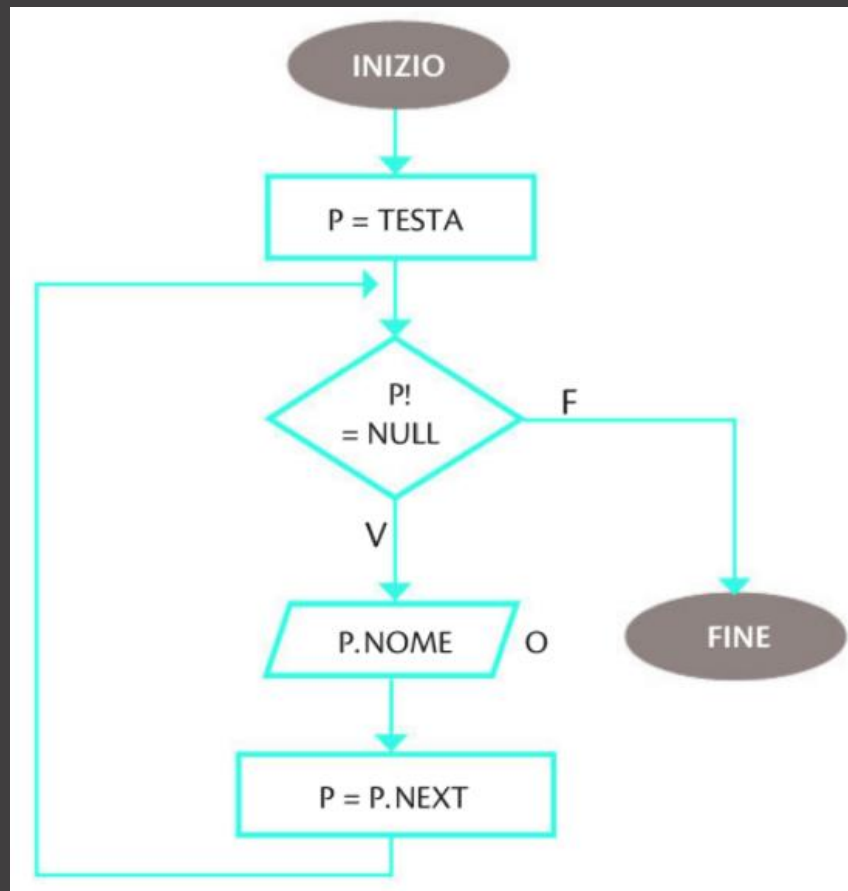


Cancellazione (POP) da una PILA

L'operazione di cancellazione di un elemento dalla **pila** si definisce **POP**:

- Salvo in **P** il valore di **TESTA** → ora **P** punta al primo elemento
- Salvo in **TESTA** il valore del puntatore dell'elemento puntato da **P** → ora **TESTA** punta al secondo elemento
- Cancello l'elemento puntato da **P**

Visualizzare elementi della PILA






Gestione di una PILA in java

Creo la classe **XXXXX** e aggiungo come attributo un **puntatore** ad un oggetto della stessa classe

Creo la classe **Pila**, contenente due attributi:

- un **puntatore** ad un oggetto della classe **XXXXX**
- un **contatore** degli elementi della pila

Gestione di una PILA in java



```
public class Vettura {  
    private String targa;  
    private int cilindrata;  
    private Vettura next;
```

```
public class Pila{  
    private Vettura first;  
    private int nodi;
```

Argomenti della lezione

03

Gestione di una coda





Gestione di una CODA

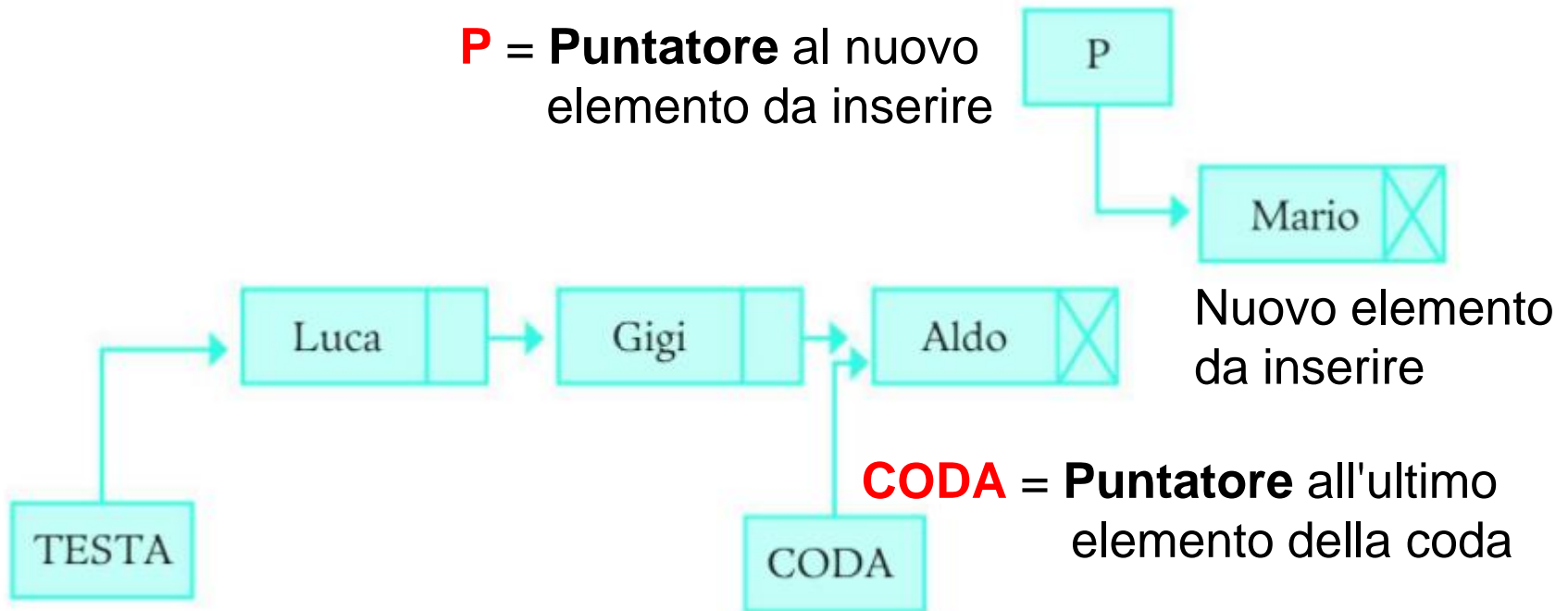
Una **coda** è gestita con il metodo **FIFO**, per cui:

- gli **inserimenti** avvengono al fondo della struttura
- le **estrazioni** avvengono sul primo elemento

Oltre al puntatore **TESTA** al primo elemento della coda, occorre un puntatore **CODA** all'ultimo elemento della **coda**.

Inserimento (PUSH) in una CODA

L'operazione di inserimento di un nuovo elemento in una **coda** si definisce **PUSH**



Inserimento (PUSH) in una CODA

1) Creo un nuovo elemento e salvo il suo indirizzo nella variabile **puntatore P**

2) Al campo puntatore dell'elemento **CODA** assegno il valore di **P**

3) A **CODA** assegno il valore di **P**





Cancellazione (POP) da una CODA

L'operazione di cancellazione (**POP**) di un elemento dalla **CODA** è uguale a quello di una pila:

- Salvo in **P** il valore di **TESTA** → ora **P** punta al primo elemento
- Salvo in **TESTA** il valore del puntatore dell'elemento puntato da **P** → ora **TESTA** punta al secondo elemento
- Cancello l'elemento puntato da **P**



Gestione di una CODA in java

Creo la classe **XXXXXX** e aggiungo come attributo un **puntatore** ad un oggetto della stessa classe

Creo la classe **Coda**, contenente tre attributi:

- un **puntatore** al primo elemento, un oggetto della classe **XXXXXX**
- un **puntatore** all'ultimo elemento, un oggetto della classe **XXXXXX**
- un **contatore** degli elementi della pila



Gestione di una CODA in java

```
public class Paziente {  
    private String nome;  
    private int num;  
    private Paziente next;
```

```
public class Coda{  
    private Paziente first;  
    private Paziente last;  
    private int numprog;
```