



# **CORSO DI PROGRAMMAZIONE**

## **PILE E CODE DI DATI**

### **DISPENSA 11.02**

[11-02\\_Pile\\_Code\\_\[ver\\_15\]](#)



Questa dispensa è rilasciata sotto la licenza Creative Common CC BY-NC-SA. Chiunque può copiare, distribuire, modificare, creare opere derivate dall'originale, ma non a scopi commerciali, a condizione che venga riconosciuta la paternità dell'opera all'autore e che alla nuova opera vengano attribuite le stesse licenze dell'originale.

Versione del: **07/11/2015**

Revisione numero: **15**

Prof. Andrea Zoccheddu  
Dipartimento di Informatica

**DIPARTIMENTO  
INFORMATICA E TELECOMUNICAZIONI**





# PILE E CODE

## LA CODA O TIPO QUEUE

### SCOPO E FUNZIONAMENTO DELLA CODA

#### LE POLITICHE FIFO

Una coda è una struttura che gestisce i dati con politica FIFO. La coda è progettata per inserire ed estrarre elementi rispettando l'ordine di arrivo. La coda consente di inserire elementi aggiungendoli (appunto) in coda, ovvero alla fine; ma quando si tratta di estrarre un elemento (restituendolo) lo si preleva dalla testa.

La coda è la tipica fila che si fa in ordine di arrivo, come per esempio le file coi numeretti della posta o del supermercato, per essere serviti nell'ordine di arrivo.

Una **coda** è definita come un elenco che gestisce gli elementi con politica FIFO.

**FIFO** significa First In First Out ovvero il primo che entra è il primo che esce.

La coda consente di mantenere l'**ordine di arrivo** degli elementi.

In Visual C# la coda è implementata con la classe **Queue**.

Anche **Queue** è una collezione (lista) di dati.

#### CARATTERISTICHE DELLE CODE

La coda è caratterizzata dai seguenti aspetti:

- È di natura dinamica e aumenta o diminuisce il numero di elementi collezionati;
- Non consente altri accessi salvo quelli predefiniti; non è possibile un accesso diretto (tipo vettore);
- Può memorizzare elementi eterogenei nel tipo;
- Mantiene gli elementi ordinati per momento di inserimento;

#### LA LIBRERIA OBBLIGATORIA

Per usare una collezione occorre aggiungere una direttiva using di inclusione dello spazio System.Collections. Prima di usare perciò dovrai scrivere:

```
using System.Collections;
```

dopo le altre direttive using del tuo programma.

## FUNZIONALITÀ DELLA CODA

#### DICHIARAZIONE DI CODA

La coda si dichiara in modo consueto:

```
Queue miacoda;
```

#### CONSTRUTTORI DI CODE

Dopo aver dichiarato la coda è possibile invocare un costruttore per prepararla in memoria. Le nuove versioni di C# consentono di istanziare la coda sia nella dichiarazione che in seguito.

```
Queue miacoda = new Queue();  
Queue altracoda;  
altracoda = new Queue();
```

il costruttore crea una coda vuota, con zero elementi dentro.



## PROPRIETÀ DI CODE

### *Proprietà Count*

La proprietà Count restituisce il numero degli elementi presenti nella coda. Gli usi consueti sono per esempio assegnare a una variabile intera il numero degli elementi:

```
int quanti = miacoda.Count;
```

## OPERAZIONI SULLA CODA

### *Metodo Clear*

**Prototipo:** void Clear()

Rimuove tutti gli elementi dalla lista. Dopo l'esecuzione di questo metodo, la lista ha lunghezza zero. Per cancellare tutti gli elementi è possibile ad esempio eseguire l'istruzione:

```
collezione.Clear();
```

### *Metodo Enqueue*

**Prototipo:** void Enqueue (Object valore)

Inserisce un nuovo elemento alla fine della coda.

Esempi:

```
miacoda.Enqueue (13); //aggiunge l'elemento 13 alla coda  
miacoda.Enqueue (17); //aggiunge l'elemento 17 alla coda  
miacoda.Enqueue ("ciao"); //aggiunge l'elemento alla coda  
miacoda.Enqueue (True); //aggiunge l'elemento alla coda
```

### *Metodo Dequeue*

**Prototipo:** Object Dequeue ( )

Rimuove e restituisce il primo elemento della coda. La Dequeue solleva un errore se la coda è vuota.

Esempi:

```
Object elemento = miacoda.Dequeue (); //estrae l'elemento 13 dalla coda  
int numero = (int) miacoda.Dequeue (); //estrae l'elemento 17 dalla coda  
string frase = (string) miacoda.Dequeue (); //estrae il "ciao" dalla coda  
bool logico = (bool) miacoda.Dequeue (); //estrae il True dalla coda  
miacoda.Dequeue (); //solleva errore perché la coda è vuota
```

## NESSUN ACCESSO DIRETTO ALLA CODA

La coda non consente l'accesso diretto.

## ISTRUZIONE FOREACH SULLA CODA

Quando una lista ha degli elementi puoi esplorarli con la istruzione foreach. Se per esempio si dispone di una listBox1 in cui visualizzarli, allora puoi fare:

```
foreach (object valori in miacoda)  
    listBox1.Add (valori);
```

osserva che se si desidera accodare solo numeri (o altri tipi) è necessario convertirli di volta in volta. Se usi foreach non puoi aggiungere o rimuovere elementi dalla lista, o avrai un errore.



## LA PILA O STACK

### FUNZIONAMENTO DELLA PILA

#### LE POLITICHE LIFO

Una pila è una struttura che gestisce i dati con politica LIFO. LIFO significa Last In First Out ovvero l'ultimo che entra è il primo che esce.

La pila è progettata per inserire ed estrarre elementi rovesciando l'ordine di arrivo. La pila consente di inserire elementi aggiungendoli (appunto) in cima, ovvero all'inizio; quando si tratta di estrarre un elemento (restituendolo) lo si preleva sempre dalla testa.

Un esempio di pila è la campana di DVD col perno al centro. Quando si inserisce un disco nella campana, esso copre quelli già presenti e quindi si trova ad essere in cima. Quando si prova a estrarre un DVD si è costretti a prelevare l'ultimo inserito.

Una **pila** è definita come un elenco che gestisce gli elementi con politica LIFO.

**LIFO** significa Last In First Out ovvero l'ultimo che entra è il primo che esce.

La coda consente di mantenere l'**ordine rovesciato di arrivo** degli elementi.

In Visual C# la coda è implementata con la classe Stack.

Anche **Stack** è una collezione (lista) di dati.

#### CARATTERISTICHE DELLE PILE

La pila è caratterizzata dai seguenti aspetti:

- È di natura dinamica e aumenta o diminuisce il numero di elementi collezionati;
- Non consente altri accessi salvo quelli predefiniti; non è possibile un accesso diretto (tipo vettore);
- Può memorizzare elementi eterogenei nel tipo;
- Mantiene gli elementi ordinati per momento di inserimento;

#### LA LIBRERIA OBBLIGATORIA

Per usare una collezione occorre aggiungere una direttiva using di inclusione dello spazio System.Collections. Prima di usare perciò dovrai scrivere:

```
using System.Collections;
```

dopo le altre direttive using del tuo programma.

### FUNZIONALITÀ DELLA PILA

#### DICHIARAZIONE DI PILA

La pila si dichiara in modo consueto:

```
Stack miaCoda;
```

#### CONSTRUTTORI DI PILA

Dopo aver dichiarato la pila è possibile invocare un costruttore per prepararla in memoria. Le nuove versioni di C# consentono di istanziare la pila sia nella dichiarazione che in seguito.

```
Stack miaPila = new Stack();  
Stack altraPila;  
altraPila = new Stack();
```

il costruttore crea una Pila vuota, con zero elementi dentro.



## PROPRIETÀ DELLE PILE

### *Proprietà Count*

La proprietà Count restituisce il numero degli elementi presenti nella Pila. Gli usi consueti sono per esempio assegnare a una variabile intera il numero degli elementi:

```
int quanti = miaPila.Count;
```

## OPERAZIONI SULLA PILA

### *Metodo Clear*

**Prototipo:** void Clear()

Rimuove tutti gli elementi dalla lista. Dopo l'esecuzione di questo metodo, la lista ha lunghezza zero. Per cancellare tutti gli elementi è possibile ad esempio eseguire l'istruzione:

```
collezione.Clear();
```

### *Metodo Push*

**Prototipo:** void Push (Object valore)

Inserisce un nuovo elemento in cima alla Pila.

Esempi:

```
miaPila.Push (13); //aggiunge l'elemento 13 alla Pila
miaPila.Push (17); //aggiunge l'elemento 17 alla Pila
miaPila.Push ("ciao"); //aggiunge l'elemento alla Pila
miaPila.Push (True); //aggiunge l'elemento alla Pila
```

### *Metodo Pop*

**Prototipo:** Object Pop ( )

Rimuove e restituisce il primo elemento della Pila. La Pop solleva un errore se la Pila è vuota.

Esempi:

```
Object elemento = miaPila.Pop (); //estrae il True dalla Pila
string frase = (string) miaPila.Pop (); //estrae il "ciao" dalla Pila
int numero = (int) miaPila.Pop (); //estrae l'elemento 17 dalla Pila
miaPila.Pop (); //estrae e perde l'elemento 13
miaPila.Pop (); //solleva un errore perché la pila è vuota
```

## NESSUN ACCESSO DIRETTO ALLA PILA

La Pila non consente l'accesso diretto.

## ISTRUZIONE FOREACH SU PILE

Quando una lista ha degli elementi puoi esplorarli con la istruzione foreach. Se per esempio si dispone di una listBox1 in cui visualizzarli, allora puoi fare:

```
foreach (object valori in miaPila)
    listBox1.Add (valori);
```

osserva che se si desidera acPilare solo numeri (o altri tipi) è necessario convertirli di volta in volta. Se usi foreach non puoi aggiungere o rimuovere elementi dalla lista, o avrai un errore.



## ESERCIZI

### ESERCIZI SULLE PILE E SULLE CODE

#### Esercizio 1) CODA DI NOMI

Prepara un programma visuale con un pulsante che memorizza in una coda dei nomi presi da una textBox; un pulsante mostra il numero di elementi in coda; un terzo pulsante elimina dalla coda il prossimo nome e lo mostra in una etichetta

#### Esercizio 2) CODA DI STRUTTURE

Di sicuro ti sei chiesto se sia possibile costruire una coda di strutture; prova a dichiarare una struttura Cliente con i campi Nome, Età, Richiesta; il programma deve prendere in ingresso i dati di un cliente, costruire una struttura e metterlo in coda (ad es. "Gigi", 23, "Informazioni"); un altro pulsante invece estrae dalla coda il cliente e lo visualizza in una casella di testo

#### Esercizio 3) PRONTO SOCCORSO

Forse sai che al Pronto Soccorso i pazienti sono classificati con un colore: rosso (urgentissimo), giallo (grave), verde (moderato), bianco (lieve). Funziona così: quando una persona arriva al pronto soccorso l'infermiere lo registra (nome e età) e decide anche il colore di priorità; appena un medico è libero deve gestire prima quelli con codice rosso, poi giallo, poi verde e poi bianco. Dichiarare una struttura **Paziente** con i campi Nome, Età, Colore; il programma deve gestire 4 code diverse e inserire il paziente nella lista corretta; quando un medico è libero preme il pulsante e il programma fornisce subito il paziente da servire (quindi sceglie il primo dei pazienti codice verde solo se non ci sono codici rosso e giallo). Le informazioni vanno visualizzate in una etichetta; i dati prelevati da caselle di testo

#### Esercizio 4) PILA DI NOMI

Prepara un programma visuale con un pulsante che memorizza in una pila dei nomi presi da una textBox; un pulsante mostra il numero di elementi in pila; un terzo pulsante elimina dalla pila il prossimo nome e lo mostra in una etichetta

#### Esercizio 5) PILA DI STRUTTURE

Sono ormai certo che ti sei chiesto se sia possibile costruire una pila di strutture; prova a dichiarare una struttura FILM con i campi Titolo, Anno, Genere; il programma deve prendere in ingresso i dati di un film, costruire una struttura e metterla in pila (ad es. "Leon", 1994, "Azione"); un altro pulsante invece estrae dalla pila il film e lo visualizza in una casella di testo

#### Esercizio 6) CONTAINER AL DOCK

Forse sai che alle banchine del Porto i Container sono dei grandi contenitori di merci. La gru solleva i Container e li mette in pila uno sull'altro. Poi dalla banchina li carica nelle stive delle navi. Dichiarare una struttura Container coi campi Codice, Peso e Tara; poi prepara un programma che carica in una pila i dati dei Container prelevati da caselle di testo; un altro pulsante invece toglie dalla pila i Container e mostra a video i dati





# SOMMARIO

- LA CODA O TIPO QUEUE ..... 2**
- SCOPO E FUNZIONAMENTO DELLA CODA ..... 2**
  - Le Politiche FIFO.....2
  - Caratteristiche delle code.....2
  - La Libreria obbligatoria .....2
- FUNZIONALITÀ DELLA CODA..... 2**
  - Dichiarazione di coda.....2
  - Costruttori di code .....2
  - Proprietà di code.....3
  - Operazioni sulla Coda .....3
  - Nessun accesso diretto alla coda .....3
  - Istruzione foreach sulla coda.....3
- LA PILA O STACK ..... 4**
- FUNZIONAMENTO DELLA PILA ..... 4**
  - Le Politiche LIFO.....4
  - Caratteristiche delle pile.....4
  - La Libreria obbligatoria .....4
- FUNZIONALITÀ DELLA PILA ..... 4**
  - Dichiarazione di pila .....4
  - Costruttori di pila .....4
  - Proprietà delle pile .....5
  - Operazioni sulla Pila .....5
  - Nessun accesso diretto alla pila .....5
  - Istruzione foreach su pile.....5
- ESERCIZI SULLE PILE E SULLE CODE ..... 6**
  - Esercizio 1) Coda di nomi .....6
  - Esercizio 2) Coda di strutture .....6
  - Esercizio 3) Pronto soccorso .....6
  - Esercizio 4) Pila di nomi .....6
  - Esercizio 5) Pila di strutture .....6
  - Esercizio 6) Container al dock .....6