

La coda con una classe definita dal programmatore

Si deve creare una classe *Coda* adatta a rappresentare una coda. Questa classe deve possedere tutte le caratteristiche della struttura di dati: numero finito degli elementi, procedura *Push* per aggiungere un elemento e procedura *Pop* per estrarre un elemento.

La classe *Coda* è rappresentata con il seguente schema UML:

Coda
Push Pop

Dati di input: il valore da inserire nella coda

Dati di output: il valore estratto dalla coda.

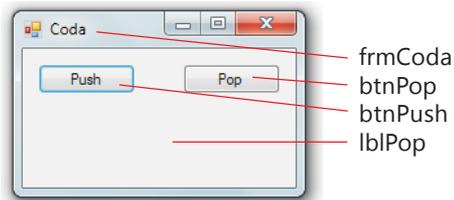
Nome del progetto

Coda di tipo *Windows Application Form*.

Disegno dell'interfaccia grafica

L'interfaccia grafica è simile a quella del progetto per la *Pila* in quanto le funzioni svolte dal programma sono analoghe.

Classe	Proprietà dell'oggetto	
Form	Name	frmCoda
	Text	<i>Coda</i>
Button	Name	btnPush
	Text	<i>Push</i>
Button	Name	btnPop
	Text	<i>Pop</i>
Label	Name	lblPop
	Text	



Definizione della classe

La **classe Coda** viene aggiunta al progetto con il comando **Aggiungi classe** del menu **Progetto**, oppure facendo clic con il tasto destro del mouse sul nome del progetto nella finestra **Esplora soluzioni** e selezionando **Aggiungi** e poi **Classe**.

La classe contiene, oltre alle variabili locali, i due metodi *Push* e *Pop*.

Metodo Push

L'aggiunta di un elemento in una coda è analogo alla procedura *Push* in una pila.

Metodo Pop

Prima di estrarre un dato, la funzione *Pop* di una coda deve verificare che la struttura non sia vuota. Quindi si effettua la verifica:

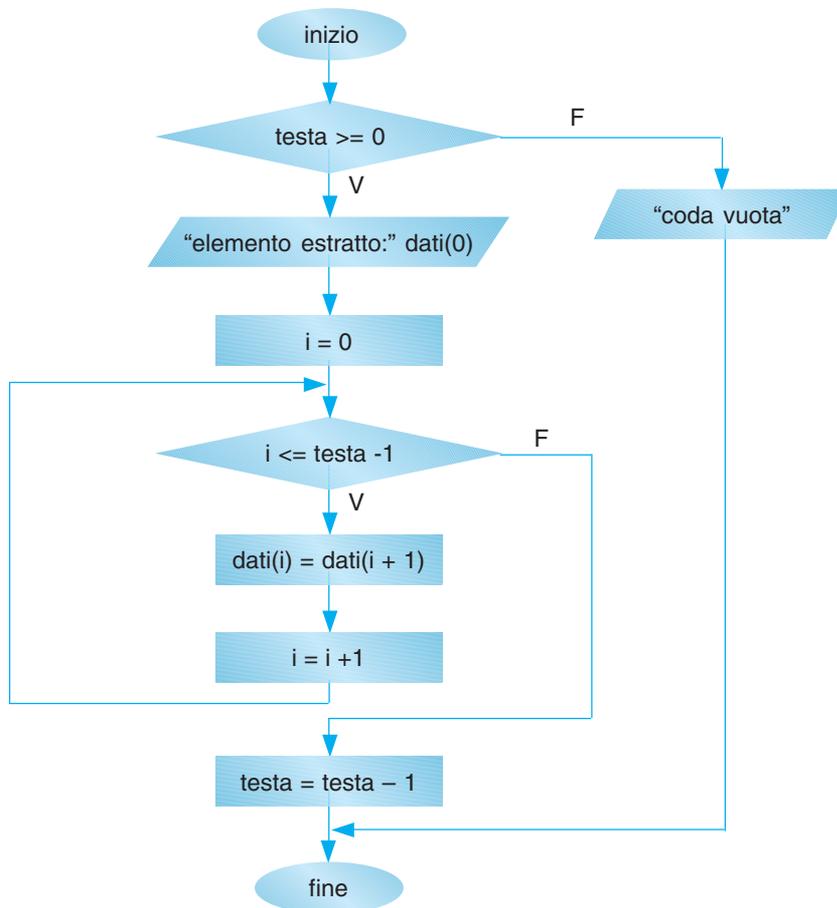
$$\text{testa} \geq 0$$

Se ciò non accade, viene avvisato l'utente che non ci sono dati da estrarre. Altrimenti viene estratto il dato in posizione 0 (il primo immesso). A questo punto è necessario spostare tutti i dati di una posizione (esattamente come nella coda a uno sportello, in cui, quando un utente viene servito, tutti gli altri avanzano di una posizione). Questa operazione viene fatta con un ciclo *For* (con l'indice che va da 1 a *testa* - 1). Dopo questa operazione il valore *testa* viene decrementato.

Algoritmo in pseudocodifica

```
inizio
se testa >= 0
  allora
    scrivi "l'elemento estratto è" dati(0)
    per i = 0 a testa - 1
      dati(i) = dati(i + 1)
    ripeti
  testa = testa - 1
altrimenti
  scrivi "la coda è vuota"
fine se
fine
```

Diagramma a blocchi



Codice Visual Basic

```
Public Class Pila
  Const max As Integer = 10
  Private testa As Integer = -1
  Private dati(max) As String
```

```

Public Sub push()
    If testa < max - 1 Then
        testa = testa + 1
        dati(testa) = InputBox("Inserisci l'elemento", "Push")
    Else
        MessageBox.Show("La coda è piena")
    End If
End Sub

Public Function pop() As String
    If testa >= 0 Then
        pop = dati(0)
        MessageBox.Show("L'elemento estratto è: " & pop, "Pop")
        For i = 0 To testa - 1
            dati(i) = dati(i + 1)
        Next
        testa = testa - 1
    Else
        MessageBox.Show("La coda è vuota", "Attenzione")
        pop = "La coda è vuota"
    End If
End Function
End Class

```

Riassumendo la classe presenta tre variabili interne:

- *max*, il numero massimo di elementi contenuti nella coda
- *testa*, l'indice dell'ultimo elemento inserito
- *dati*, il vettore che contiene i dati,

e due metodi:

- *push*, verifica se la coda è piena; in caso contrario permette, tramite *InputBox* di aggiungervi un elemento. Questa procedura è analoga a quella della pila.
- *pop*, verifica se la coda è vuota; in caso contrario permette di estrarre un elemento. Estrae l'elemento di posto 0 (il primo inserito) e sposta tutti gli elementi di una posizione.

Il programma principale definisce un oggetto *MyCoda* e utilizza due pulsanti di comando per gestire i metodi della classe:

```

Public Class frmCoda
    Public myCoda As New Coda
    Private Sub btnPush_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPush.Click
        myCoda.push()
    End Sub

    Private Sub btnPop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPop.Click
        lblPop.Text = myCoda.pop()
    End Sub
End Class

```

Si osservi che il programma non permette di visualizzare nè l'elenco di tutti gli elementi compresi nella struttura di dati, nè la lunghezza della coda stessa.