

# Corso sul linguaggio C++

Modulo 5

Pila

Coda

5 – Strutture dinamiche

## Prerequisiti

- Strutture dinamiche e loro operazioni
- Memoria heap

# Introduzione

In questa Unità si illustra la tecnica dell'allocazione dinamica, che consiste nell'allocare spazio per i dati durante l'esecuzione del programma.

L'allocazione viene fatta prelevando celle dalla memoria **heap**.

Applichiamo la tecnica di allocazione dinamica per la creazione di **array dinamici**.

# Obiettivi

- Gestione memoria dinamica
- Utilità degli array dinamici

## Conoscenze (sapere)

- Istruzioni per allocare e deallocare memoria
- Utilizzo di un array dinamico

## Competenze (saper fare)

- Allocare e deallocare memoria dinamicamente
- Usare array dinamici

# Argomenti

- Allocazione dinamica
- Array dinamico
- Classe CStringa

# Classe CStringa

Vediamo una applicazione dell'array dinamico alla costruzione della classe CStringa.

La classe avrà inizialmente la seguente definizione:

```
class CStringa
{ private:
    unsigned lunghezza;      /* solo valori positivi */
    char *s;                /* stringa da allocare dinamicamente
                             nel costruttore */

    public:
    CStringa(int l);         /* costruttore */
    void GetString();       /* visualizzazione stringa */
    void SetString();       /* acquisizione stringa */
};
```

# Classe CStringa

```

CStringa::CStringa(int l)
{
    lunghezza=l;
    s=new char[l+1];
}
void CStringa::SetString()
{
    cin>>s;
}
void CStringa::GetString()
{
    cout<<s;
}
int main()
{
    CStringa stringa(15);
    stringa.SetString();
    stringa.GetString();
    system("Pause");
    return 0;
}
    
```

Allocazione dinamica di *s*

Alloca oggetto *stringa* al momento della dichiarazione, con dimensione 15

# Classe CStringa

```

class CStringa
{
private:
    unsigned lunghezza;
    char *s;
public:
    CStringa(int l);
    CStringa(const char *c)
    void GetString();
    void SetString();
};

CStringa::CStringa(int l)
{
    lunghezza=l;
    s=new char[l+1];
}
void CStringa::SetString() { cin>>s;}
void CStringa::GetString() { cout<<s; }
CStringa::CStringa(const char *c)
{
    lunghezza=strlen(c);
    s=new char[lunghezza+1];
    strcpy(s,c);
}
int main()
{
    CStringa stringa(15);
    CStringa prompt("Immetti stringa: ");
    prompt.GetString();
    stringa.SetString();
    stringa.GetString();
    system("Pause");
    return 0;
}
    
```

In questo caso si definisce un secondo costruttore *CStringa()* che ha come parametro una stringa costante, allocata anche essa dinamicamente

Dichiarazione oggetto *prompt*

L'istanza consente la visualizzazione del messaggio "Immetti stringa: "

# Classe CStringa

```
class CStringa
{ private:
    unsigned lunghezza;
    char *s;
public:
    CStringa(int l);
    CStringa(const char *c)
    ~CStringa();
    void GetString();
    void SetString();
};
```

```
CStringa::CStringa(int l)
{ lunghezza=l; s=new char[l+1]; }
void CStringa::GetString() {cin>>s;}
void CStringa::SetString() {cout<<s;}
CStringa::CStringa(const char *c)
{ lunghezza=strlen(c);
  s=new char[lunghezza+1];
  strcpy(s,c); }
CStringa::~CStringa()
{ lunghezza=0; delete [] s;
  cout<<"Spazio deallocato!"<<endl; }
int main()
{ CStringa stringa(15);
  CStringa prompt("Immetti stringa: ");
  prompt.GetString(); stringa.SetString();
  stringa.PrintString();
  system("Pause");
  return 0;}
```

Il distruttore `~CStringa()` consente di deallocare l'oggetto al termine dell'esecuzione

M. Malatesta 5-Strutture dinamiche-05

11  
19/07/2010

## Attività

- 1) Caricare e provare la classe CStringa mediante l'implementazione con array dinamico
- 2) Caricare e provare la classe CStringa mediante l'implementazione con array statico
- 3) Creare una classe ArrayClass per la gestione di un array dinamico
- 4) Modificare il programma BubbleSort.cpp, mostrato nell'Unità, in modo che l'array rappresenti una **struct** *dipendente* (formata dagli attributi cognome, nome, reparto e stipendio) e che realizzi le stampe dei dati dell'array ordinate, rispettivamente, per cognome e per stipendio

M. Malatesta 5-Strutture dinamiche-05

12  
19/07/2010

# Terminologia

- Allocazione dinamica
- Deallocazione
- Memopria heap
- Array dinamico
- **new()** e **delete ()**

# Riepilogo rivedere

In questa Unità abbiamo mostrato le istruzioni per gestire l'allocazione dinamica della memoria (**new ()** e **delete ()**) e le abbiamo applicate ad un primo esempio di struttura dinamica: l'**array dinamico**.

Tramite array dinamico abbiamo realizzato la costruzione della classe CStringa con i relativi metodi.

## Altre fonti di informazione

- P.Gallo, F.Salerno – Informatica Generale 1, ed. Minerva Italica
- M.Romagnoli, P.Ventura – Linguaggio C/C++, ed. Petrini
- A.Garavaglia, F.Petracchi,S.Forte-INFORMATICA-ed. Masson