

Esempio Pilota:
EreditarietaPersonaAlunno

Definizione della Form

Codice della Form

```
using System;
using System.Windows.Forms;
```

```
namespace EreditarietaPersonaAlunno
{
    public partial class frmAvvio : Form
    {
        public frmAvvio()
        {
            InitializeComponent();
        }
    }
}
```

```
// serve UN solo oggetto, che sarà una Persona oppure un Alunno ...
Persona miaPersona;
```

```
private void plsVerifica_Click(object sender, EventArgs e)
{
    if (chkAlunno.Checked)
```

```
/* Crea l'oggetto ALUNNI e assegno a miaPersona ...
   (... è possibile perchè "un ALUNNO è anche una PERSONA"
   ossia tutte le funzionalità della classe PERSONE
   le ha anche la classe ALUNNI) */
```

```
miaPersona = new Alunni
    ( txtNominativo.Text, (int) nudAnnoNascita.Value,
      (double) nudMediaVoti.Value );
```

```
else
```

```
// Crea l'oggetto PERSONE e assegno a miaPersona ...
miaPersona = new Persona
    ( txtNominativo.Text, (int) nudAnnoNascita.Value );
```

```
// calcola e visualizza i risultati ...
```

```
lblTesto.Text = miaPersona.Testo( );
```

```
lblContributo.Text = miaPersona.ContributoStatale( ).ToString();
```

```
/* Nota Bene: al momento dell'esecuzione, a seconda dell'oggetto
   presente in miaPersona (ALUNNO o PERSONA), viene eseguito il
   metodo ContributoStatale() della classe relativa */
```

```
    }
}
}
```

Codice della Classe Base "Persone"

```
namespace EreditarietaPersonaAlunno
```

```
{
    class Persone
```

```
{
    // attributi ...
```

```
protected string _Nominativo;
protected int _AnnoNascita;
```

```
// costruttori ...
```

```
public Persone ( string nuovoNominativo, int nuovo Anno )
{
    _Nominativo = nuovoNominativo;
    _AnnoNascita = nuovoAnno;
}
```

```
// metodi ...
```

```
public virtual int ContributoStatale( )
{
    if ( _AnnoNascita < 1960 )
        return 200;
    else
        return 100;
}
```

```
Public virtual string Testo( )
```

```
{
    return _Nominativo + " (Persona)" + (char)13 +
           "nato nel " + _AnnoNascita.ToString();
}
}
```

Codice della Classe Derivata "Alunni"

```
namespace EreditarietaPersonaAlunno
```

```
{
    class Alunni: Persone // ... Alunni eredita da Persone
    {
        private double _MediaVoti;
```

```
// costruttori ...
```

```
public Alunni (string nuovoNominativo, int nuovoAnno,
              double nuovaMedia) :
              base (nuovoNominativo, nuovoAnno)
{
    _MediaVoti = nuovaMedia;
}
```

```
// metodi ...
```

```
public override int ContributoStatale()
{
    // richiama il ContributoStatale della classe base ...
    int ContributoDiBase = base.ContributoStatale();

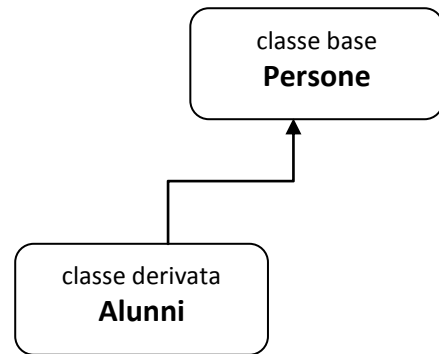
    if ( _MediaVoti < 6 )
        return ContributoDiBase;

    else if ( _MediaVoti >= 8 )
        return ContributoDiBase * 150 / 100;

    else
        return ContributoDiBase * 120 / 100;
}
```

```
public override string Test()
{
    return _Nominativo + " (Alunno)" + (char)13 +
           "nato nel " + _AnnoNascita.ToString() + (char)13 +
           "media voti " + _MediaVoti;
}
}
```

Gerarchia delle Classi e Diagrammi UML



classe base Persone

attributi

- **_Nominativo** string
 - **_AnnoNascita** string

costruttori

+ **Persone** (string nuovoNominativo, int nuovoAnno)

metodi

+ **ContributoStatale** () **metodo virtual**
 + **Testo** () **metodo virtual**

classe derivata Alunni

attributi

- **_MediaVoti** double
 ... eredita gli attributi **_Nominativo, _AnnoNascita**

costruttori

+ **Alunni** (string nuovoNominativo, int nuovoAnno, double nuovaMedia)

metodi

+ **ContributoStatale** () **override del metodo virtual**
 + **Testo** () **override del metodo virtual**