

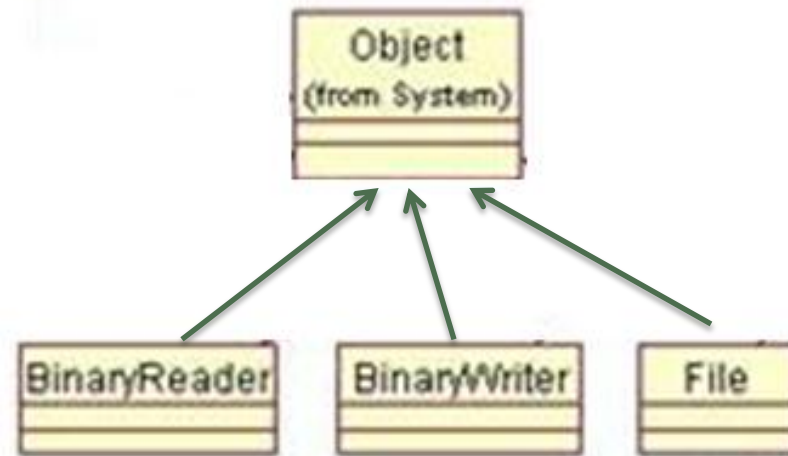
---

# File BinaryReader BinaryWriter

---

Prof. Francesco Accarino  
IIS Altiero Spinelli Sesto San Giovanni

In C# abbiamo due oggetti **BinaryReader** e **BinaryWriter** per lavorare con i file di tipo binario. L'ambiente mette a disposizione anche una classe statica **File** per la manipolazione dei file (creazione copia spostamento ecc-)



**BinaryReader** e **BinaryWriter** sono dei lettori e scrittori di stream di tipo binario che realizzano operazioni di input/output su un **FileStream** di tipo formattato. Permettono cioè la lettura e la scrittura di Int Float Double String ecc. Il semplice **FileStream** permette di realizzare un semplice flusso di byte dall'applicazione al file viceversa.

# FileStream

Member	Description
CanRead	Returns a Boolean value indicating whether the stream can read from the data source.
CanWrite	Returns a Boolean value indicating whether the stream can write to the data source.
Length	Returns the length or number of bytes in the current stream.
Position	Gets/Sets the current position of the stream. Any read/write operation on the stream is carried out at the current position.
Close()	Closes the stream.
Flush()	Writes all the data stored in the stream buffer to the data source.
Seek()	Sets the current position of the stream.
Read(byte[] buffer, int offset, int count) (Return type: int)	Reads the specified number of bytes from the current position of the stream into the supplied array of bytes and returns the number of bytes actually read from the stream.
ReadByte()	Reads a single byte from the current position of the stream and returns the byte casted into an int. The '-1' return value indicates the end of the stream of data.
Write(byte[] buffer, int offset, int count) (Return type: void)	Writes the specified number of bytes at the current position of the stream from the supplied array of bytes.
WriteByte()	Writes a single byte at the current position of the stream.

## Esempio di lettura di un file binario come vettore di byte

```
FileStream fs;  
  
fs = File.Open("dati.dat", FileMode.Open, FileAccess.Read, FileShare.None);  
  
byte[] testo = new byte[fs.Length];  
  
fs.Read(testo, 0, testo.Length);  
  
txtDati.Text = System.Text.Encoding.ASCII.GetString(testo);  
  
fs.Close();
```

## Esempio di scrittura su un file binario prendendo il contenuto da una TextBox

```
FileStream fs;  
  
fs = File.Open("dati.dat", FileMode.Open, FileAccess.Write, FileShare.None);  
  
byte[] testo = System.Text.Encoding.ASCII.GetBytes(txtDati.Text);  
  
fs.Write(testo, 0, testo.Length);  
  
fs.Close();
```

# Utilizzo di BinaryWriter

```
FileStream fs;
BinaryWriter bw;
if(File.Exists("dati.dat")==true)//utilizzo la classe statica File per testare se il file esiste
fs = File.Open("dati.dat", FileMode.Append);//se esiste lo apro in append
else
    fs = File.Open("dati.dat", FileMode.Create);//altrimenti lo creo

bw = new BinaryWriter(fs);//creo un streamwriter di tipo binario sul file

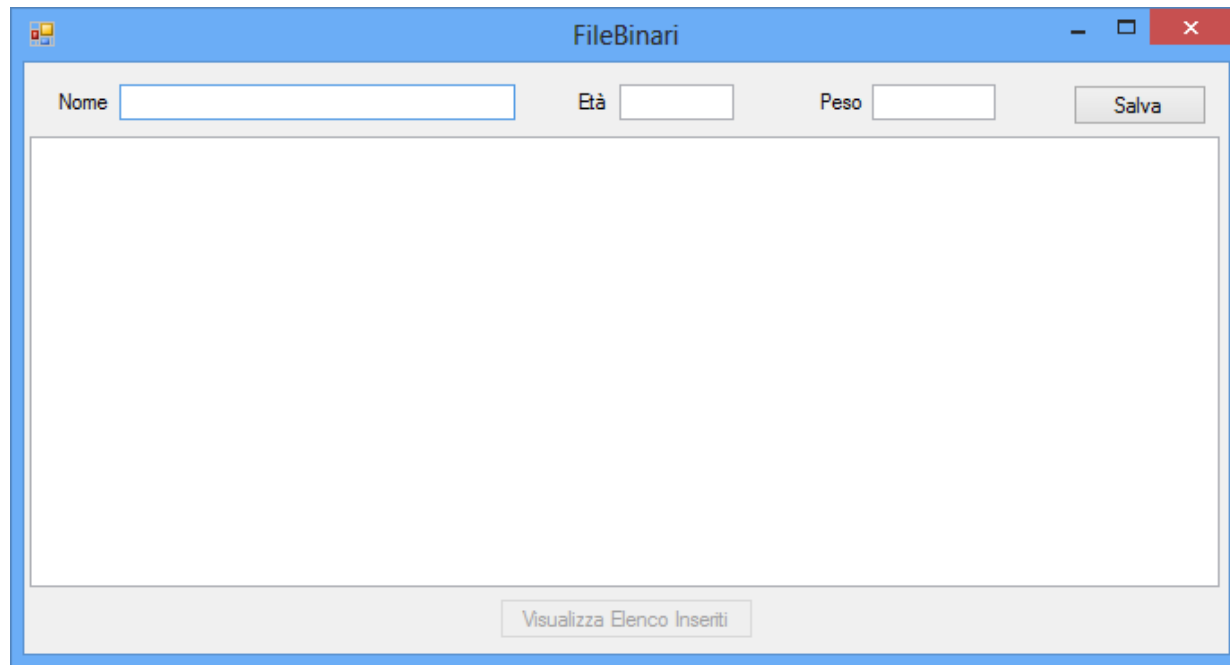
nome = txtNome.Text;
eta = Convert.ToInt32(txtEta.Text);
peso = Convert.ToDouble(txtPeso.Text);
bw.Write(nome);
bw.Write(eta);
bw.Write(peso);
bw.Close();
fs.Close();
```

## Utilizzo di BinaryReader

```
long lunghezza;
FileStream fs;
BinaryReader br;
fs = File.Open("dati.dat", FileMode.Open);
lunghezza = fs.Length;
br = new BinaryReader(fs);
string nome;
int eta;
double peso;
txtDati.Text = "";
while (fs.Position < lunghezza) {

    nome = br.ReadString();
    eta = br.ReadInt32();
    peso = br.ReadDouble();
    txtDati.Text = txtDati.Text + "Nome: " + nome + " " + "Età: " + eta.ToString() + " "
        + "Peso : " + peso.ToString() + "\r\n";
}
fs.Close();
br.Close();
```

**Esercitazione:** Costruire un'applicazione come quella mostrata in figura in cui l'utente inserisce un nominativo una età e un peso e i dati vengono salvati in un file di tipo binario cioè una stringa per il nome un int per l'età e un double per il peso.



Il bottone salva creerà il file se esso non esiste oppure lo apre in append e salva su di esso i dati inseriti  
Il bottone visualizza sarà disabilitato se il file non è stato ancora creato oppure visualizza il suo contenuto nella textBox