

### Foto gallery

Realizzare un programma che sia in grado di visualizzare le immagini memorizzate in una cartella. Il programma dovrà permettere di:

1. Visualizzare le immagini contenute in una cartella.
2. Visualizzare una singola immagine a grandezza naturale.
3. Visualizzare le informazioni descrittive (vedi **file di titolazione**) associate a una determinata immagine.

### Immagine (tipi di file immagine da gestire)

Il programma deve essere in grado di visualizzare qualsiasi file immagine che abbia una delle seguenti estensioni: *png, jpg, jpeg, bmp, tiff*.

### File di titolazione

Per ogni immagine può esistere un file di titolazione; il suo nome deve essere:

*file-immagine.estensione.info*

Ad esempio, posta l'esistenza del file: *"tigre bengala.png"*, potrebbe esistere il file di titolazione: *"tigre bengala.png.info"*

### Formato del file

Il file deve avere il seguente formato:

```
#intestazione opz (1 riga al massimo)
Titolo (una sola riga) opz (1 riga al massimo)
Descrizione (quante righe si vuole) opz (quante righe si vuole)
```

Ad esempio:

```
#File di titolazione MyFotoGallery
Tigre del bengala
La foto ritrae un esemplare adulto di tigre del bengala.
L'immagine è stata catturata...
```

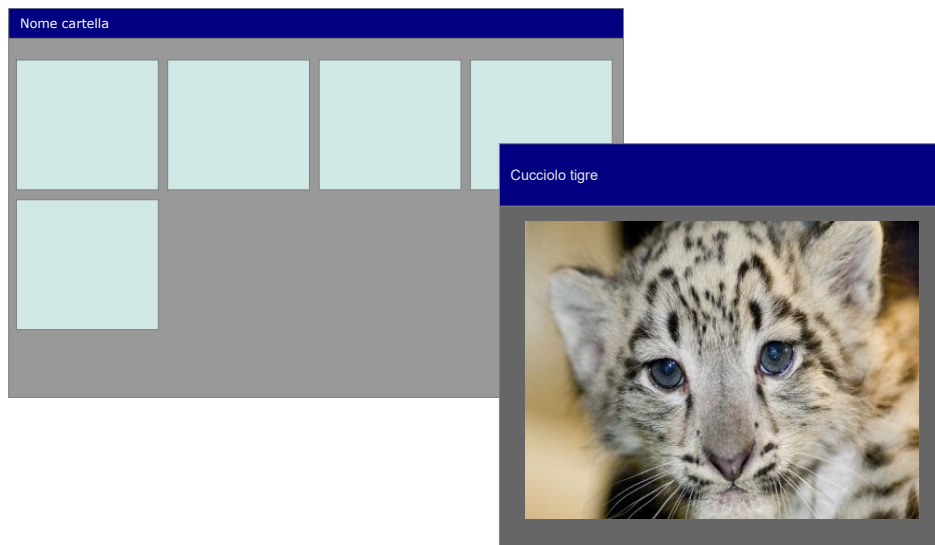
Nota bene: tutti gli elementi del file sono opzionali. Mentre l'intestazione è caratterizzata dal carattere `#`, l'eventuale assenza del titolo è indicata da una riga vuota.

```
#File di titolazione my foto gallery

La foto ritrae un esemplare adulto di tigre del bengala.
L'immagine è stata catturata...
```

## Note sulla UI

Lo *screen shot* mostra gli elementi di una ipotetica UI:



Il componente centrale della UI è la *gallery* e cioè il pannello che visualizza tutte le immagini. Allo scopo è opportuno usare un *flow layout panel*, poiché è in grado di disporre le foto disponendole automaticamente in base allo spazio disponibile.

### **Visualizzazione delle immagini all'interno della gallery**

Le singole immagini della *gallery* possono essere visualizzate mediante delle *picture box*.

### **Uso di una cartella immagini predefinita**

Conviene aggiungere al progetto una cartella contenente alcuni file immagine e di titolazione, facendo sì che il programma ne carichi direttamente il contenuto all'avvio. Successivamente si potrà sviluppare la funzionalità che consente all'utente di selezionare la cartella contenente le immagini (ad esempio mediante una *folder browser dialog*).

### **Visualizzazione delle informazioni su una singola immagine**

Si è liberi di implementare questa funzionalità come si desidera:

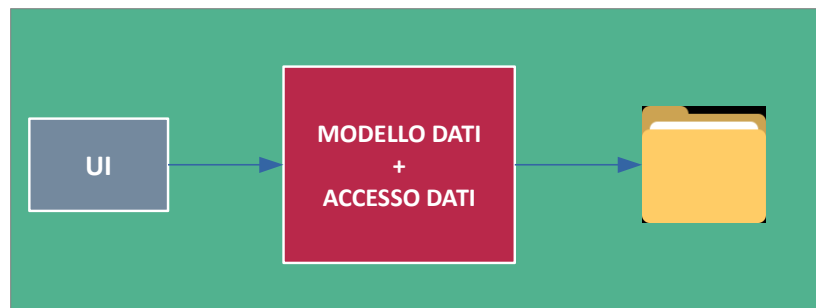
- Visualizzazione di un *tooltip* quando il mouse è posizionato sopra una foto.
- Visualizzazione mediante una seconda form, che si apre attraverso un click (o doppio click) sull'immagine

### **Visualizzazione della foto in grandezza naturale**

Si suppone che l'utente esegua un doppio-click (o altra azione) sull'immagine foto scelta e questa venga visualizzata mediante un form modale.

## Note sull'architettura del programma

Si vuole separare la gestione delle informazioni dalla gestione della UI.



Si suppone che il “rettangolo rosso” fornisca le seguenti funzionalità:

- 1 **Modello dati:** un record (o un oggetto) `Foto` (o nome a scelta) che incorpori le informazioni rilevanti sulla singola foto: nome del file immagine, titolo e descrizione.
- 2 **Accesso dati + modello dati:** un oggetto che gestisca l'elenco delle immagini e consenta di caricarle a partire dalla cartella che le contiene.

Nota bene: entrambi i componenti devono essere *UI-free*: non devono avere dipendenze dal tipo di interfaccia utente utilizzata (e dunque nemmeno con dai *namespace* collegati al tipo di UI, come `System.Drawing`).