

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.005 Elements of Software Construction  
Fall 2008

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

# 6.005 Elements of Software Construction | Fall 2008

## Exploration 3: Photo Organizer

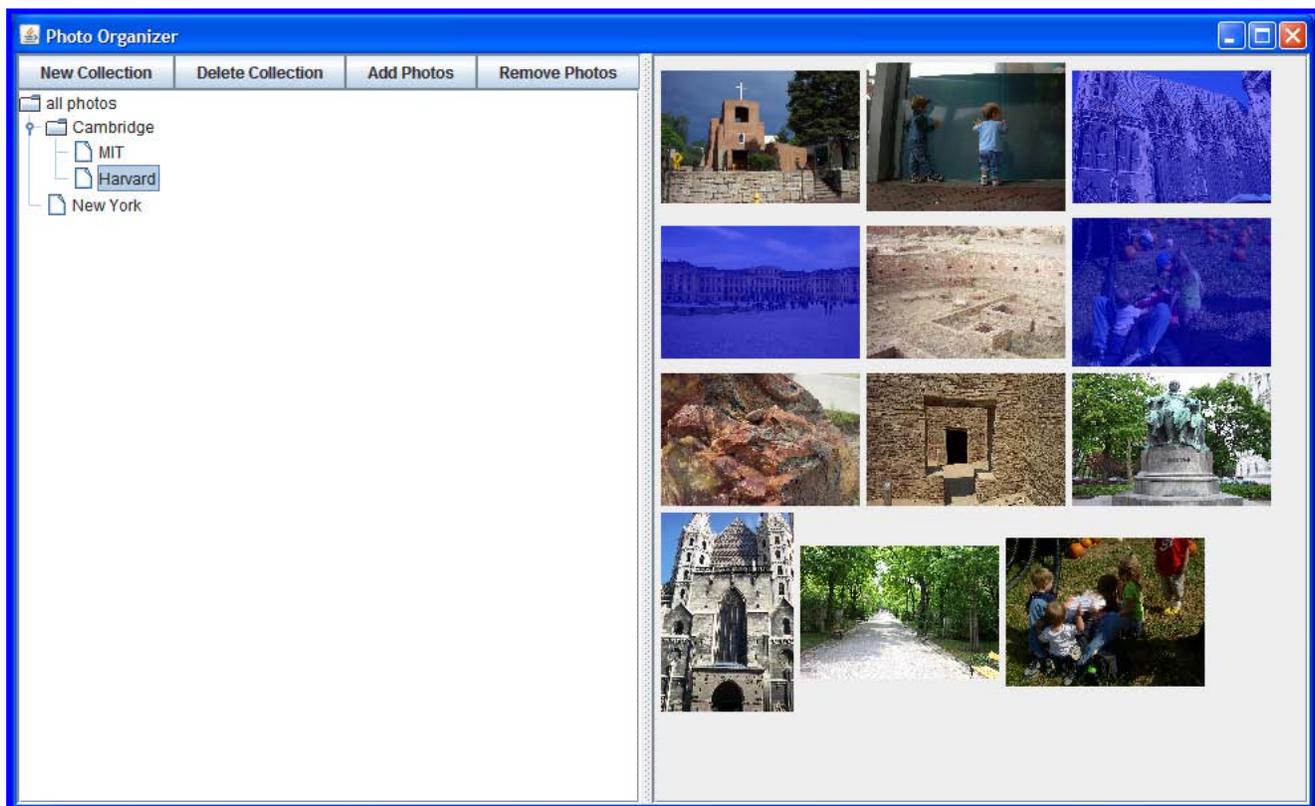
---

The purpose of this exploration is to give you practice coding in Java, and to introduce you to the case study that will be presented in the lectures on programming with mutable data and relations. Your solution will be judged by (1) correctness -- whether it meets the specification, and (2) clarity -- whether your code is modular, well-organized, carefully commented, and makes proficient use of Java. You are *not* required to use any of the ideas or patterns that have not yet been taught in lecture.

## Specification

---

Your task is to build an organizer for digital photos, similar to the screenshot below.



The photo organizer should have the following features:

- A set of photos is loaded by finding all the JPEG files in a subtree of the filesystem specified by a command-line argument. (When run with no arguments, the organizer

should just use the set of sample photos provided.)

- The user can create and manage named *albums* of photos. An album is a subset of all the loaded photos. A photo can appear in more than one album. There is an album, "all photos", containing all the loaded photos.
- Albums are hierarchical. An album may have subalbums, which are subsets of the parent album's photos. For example, the user might create a "Cambridge" album for all the photos taken in Cambridge, and an "MIT" subalbum under it that contains all photos from MIT. The "all photos" album is the root of the album hierarchy.
- The photo organizer window has two parts: a tree on the left showing the album hierarchy, and an array of photo thumbnails on the right.
- Clicking on an album name in the tree selects it. At most one album may be selected at a time.
- Double-clicking on an album name displays the thumbnails for the album's photos on the right. (Note that the *displayed* album is not necessarily the same as the *selected* album, since another album can be single-clicked to select it without changing the displayed thumbnails.)
- Clicking on a photo thumbnail selects or deselects it. Selected photos are highlighted in blue. Any number of photos may be selected.
- Buttons on the window allow making a new album with a user-provided name, deleting the selected album, and adding and removing the selected photos from the selected album.
- Adding and removing a photo from an album should immediately affect the album. If you remove a photo from an album, then that album should stop displaying that photo. Subsequent user operations may put the photo back in the album, however. Other albums may also be affected by adding or removing a photo, as necessary to preserve the album hierarchy property.
- As an optional extension, display the size of each album in the tree (e.g. "MIT (45)" if the MIT album has 45 photos in it).

If this specification is incomplete, you should use your own design judgement to fill in the details. In general, your program should not crash, throw uncaught exceptions, or allow the user to put the interface into a state that violates this specification.

## Infrastructure

---

The `exploration3` project in your repository contains this assignment, some skeleton code for the photo organizer, and some sample photos. Your work should be submitted by committing to this project.

## Hints

---

Note that albums are *not* the same as folders in the filesystem, so you don't have to make a folder on disk when the user creates an album. The notion of an album is just internal to your photo organizer. It's also acceptable if your photo organizer completely forgets the albums the user created when it restarts. (As an optional extension, however, you might make the albums persistent by saving and loading them from a file.)