

Image Album Project

Ante Hakstok

Marko Pancirov

Antonio Labriola

Ane Hakstok

Image I/O

- Handles pictures, albums and JSON I/O
- Album class
- It has Album class which aggregates Pictures.
- It can add tags, remove and get tags to itself.
- It can add parent albums and remove child albums.
- Album can add, get and remove Picture objects.

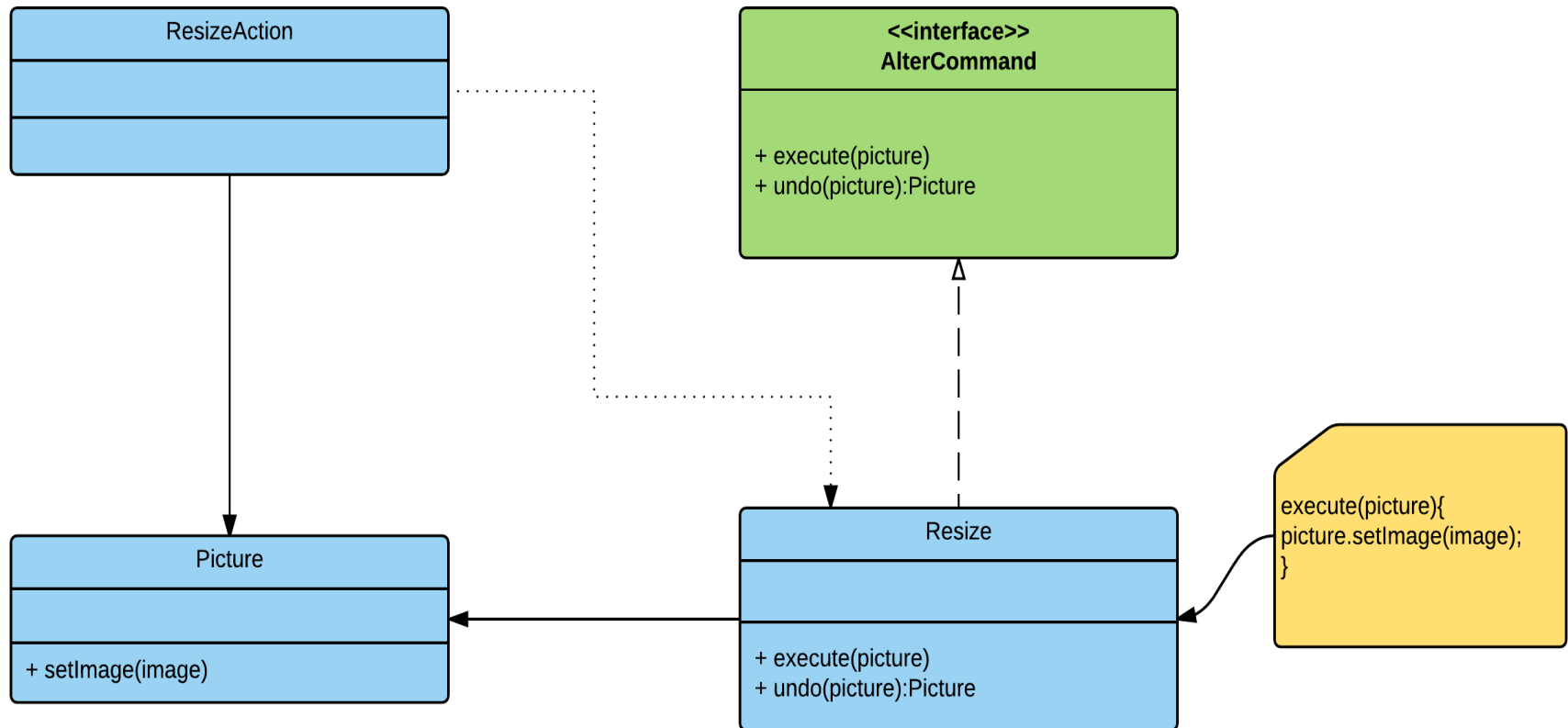
JSON I/O

- JSONReader and JSONWriter manage reading and writing to json file which holds data about images.
- Picture class represents a picture. Each picture has its tag list and those tags can be added or removed. Also each picture has a list of all parent albums.
- The picture has its name, id and file path.

AlterCommands – Command Pattern

- AlterCommand interface is a layout for the editing commands. Each command class implements this interface, so it has to implement execute and undo methods. This is command pattern.
- The commands and corresponding classes are: Crop, Flip, Mirror, Resize, Rotate and Search.
- Command: AlterCommand
- Concrete Command: Resize
- Client/Invoker: ResizeAction
- Receiver: Picture

AlterCommands – Command Pattern



UndoRedo

- BasicUndoRedoStack has a undo and redo stacks which store the commands which represent the changes made to the picture.
- It has undoPicture and redoPicture methods.

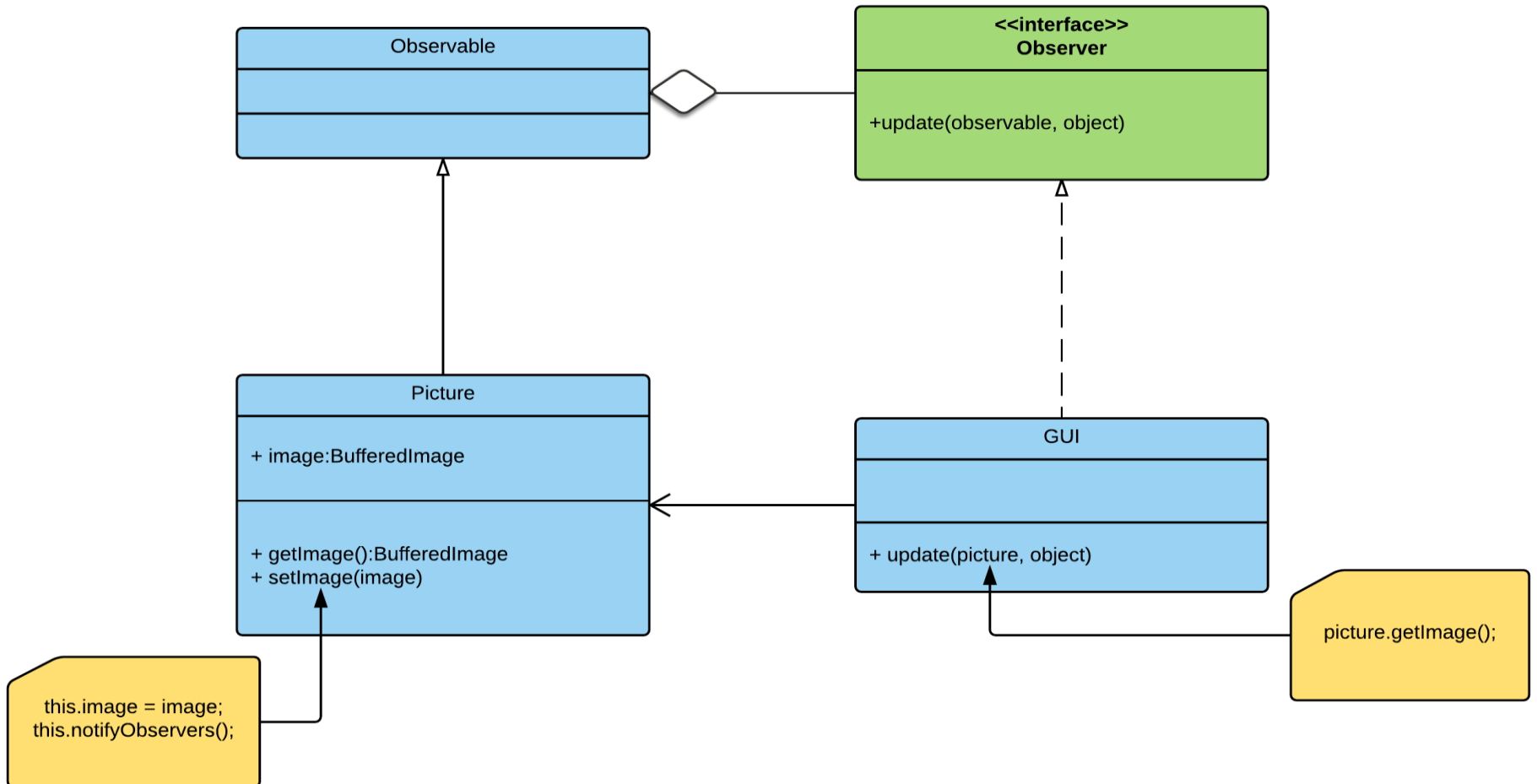
GUI

- The main class is GUI which holds and manages all the components of the graphical user interface.
- It stores the picture which is currently being edited, so it retains a reference of the Picture class.
- For each element on the GUI there is a class that is a Listener which performs the corresponding action when the user clicks that element.
- For example the class CropAction retains a reference to the GUI and instantiates the Crop class and calls its execute method on the picture that is currently being edited which is stored in the GUI class.
- Then it notifies the Observers to let them know that a change happened and it adds the crop to the undo/redo stack.

GUI – Observer Pattern

- In this case the Picture is Observable and the GUI is the Observer
- GUI needs to update the screen when a change happens to a picture. For example when the Crop command makes changes to the picture it uses the Picture's setImage() method which notifies the observers – in our case the GUI class
- We are using standard java Observable class and Observer interface
- Concrete subject: Picture
- Concrete observer: GUI

GUI – Observer Pattern



Controller

- Contains the main method which sets up everything and starts the application. It instantiates the master album and the GUI.