If we type in anything that is neither a filename nor a color, that text is displayed in the right-hand list box.

This sequence of steps is shown in Figure 16.4.

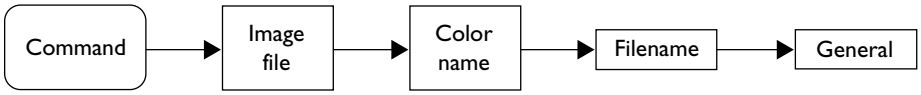

Figure 16.4 The command chain for the program in Figure 16.3.

To write this simple chain of responsibility program, we start with an abstract Chain class.

```
public interface Chain {
    public abstract void addChain(Chain c);
    public abstract void sendToChain(String mesg);
    public abstract Chain getChain();
)
```

The *addChain* method adds another class to the chain of classes. The *getChain* method returns the current class to which messages are being forwarded. These two methods allow us to modify the chain dynamically and add additional classes in the middle of an existing chain. The *sendToChain* method forwards a message to the next object in the chain.

The Imager class is derived from JPanel and implements the Chain interface. It takes the message and looks for .jpg files with that root name. If it finds one, it displays it.

```
public class Imager extends JPanel implements Chain {
    private Chain    nextChain;
    private Image    img;
    private boolean loaded;

    public Imager(){
        super();
        loaded = false;
        setBorder(new BevelBorder(BevelBorder.RAISED));
    }
//--------------
    public void addChain(Chain c) {
        nextChain = c;    //next in chain of resp
    }
//--------------
    public void sendToChain(String mesg) {
        //if there is a JPEG file with this root name
        //load it and display it.
        if (findImage(mesg))
            loadImage(mesg+".jpg");
```