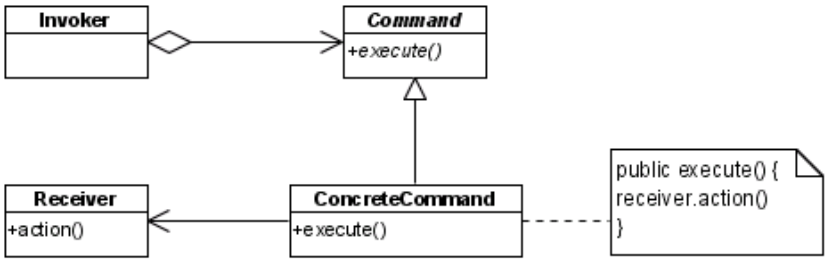
Supplementary Exercise on Command Pattern

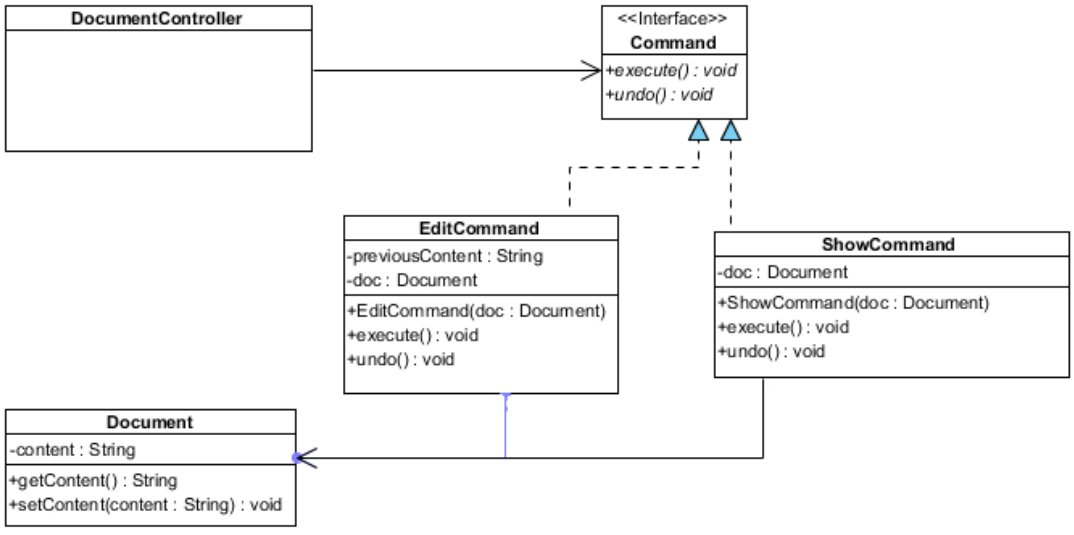
(a)



(b)

- The **Invoker** holds a command and can get the **Command** to execute a request by calling the execute method.
- The **Command** is an object that encapsulates a request to the receiver. **Command** declares an interface for all commands, providing a simple execute() method which asks the **Receiver** of the command to carry out an operation.
- **Receiver**. The receiver is the component that is acted upon by each request.
- The **ConcreteCommand** defines a binding between the action and the receiver. When the **Invoker** calls execute the **ConcreteCommand** will run one or more actions on the **Receiver**.

(c)



(d)

```
public interface Command {
        public void execute();
        public void undo();
}

public class EditCommand implements Command {
        private Document doc ;
        private String previousContent;

        public EditCommand (Document doc){
                this.doc = doc;
        }
        public void execute() {
                // save previous content
                previousContent = doc.getContent();
               // get new content
                Scanner s = new Scanner(System.in);
                System.out.println("Please input your content");
                String value = s.nextLine();
                doc.setContent(value);
        }

        public void undo() {
                doc.setContent(previousContent);

        }
}

public class ShowCommand implements Command {
        private Document doc ;

        public ShowCommand (Document doc){
                this.doc = doc;
        }
        public void execute() {
                System.out.println("The current content is "+
                                                        doc.getContent());
        }

        public void undo() { }
}
```