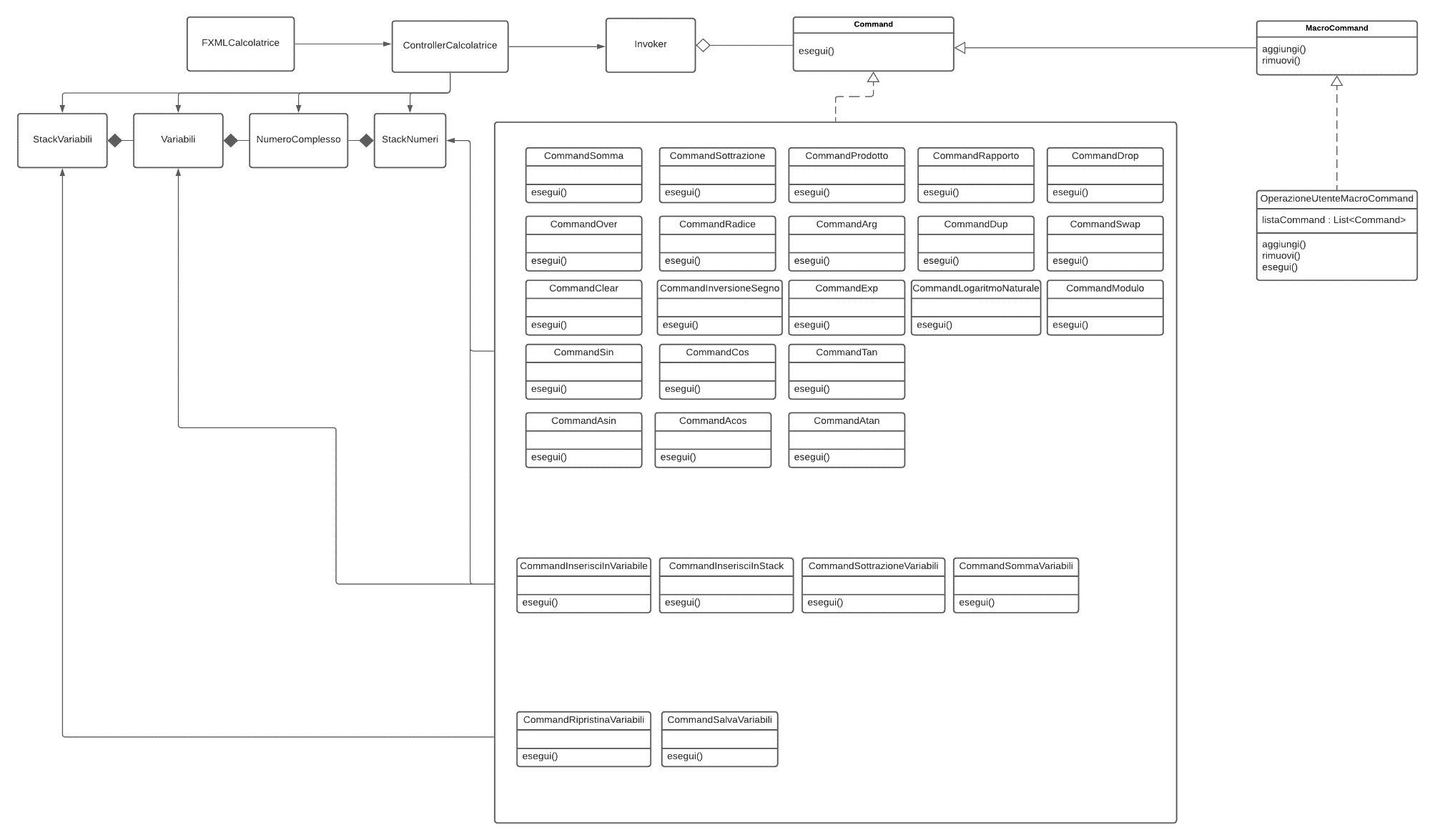
**ARCHITETTURA SOFTWARE**

****L’architettura del software è basata sul pattern MVC (Model-View-Controller), il quale si basa sulla separazione della logica di funzionamento dell’applicazione dalla parte di visualizzazione dei dati e di interazione con l’utente.

**MODEL:**

MODEL:

* NumeroComplesso: modella il tipo di dato numero complesso sul quale si basa il funzionamento della calcolatrice e gestisce le operazioni sui numeri (addizione, sottrazione, etc);
* StackNumeri: memorizza i numeri inseriti e i risultati delle operazioni secondo una logica lifo, gestisce le operazioni sullo stack (swap, over, etc)
* Variabili: gestisce le 26 variabili messe a disposizione dalla calcolatrice;
* OperazioniUtente: gestisce la creazione e la gestione delle funzioni personalizzate dell’utente;
* StackVariabili: memorizza gli stati delle 26 variabili della calcolatrice.
* NumeroComplesso: modella il tipo di dato numero complesso sul quale si basa il funzionamento della calcolatrice;
* StackNumeri: memorizza i numeri inseriti e i risultati delle operazioni secondo una logica lifo;
* Variabili: gestisce le 26 variabili che la calcolatrice mette a disposizione;
* StackVariabili: memorizza gli stati delle 26 variabili secondo una logica lifo.
* Command: interfaccia che rappresenta i comandi da eseguire, ovvero le diverse operazioni che la calcolatrice deve svolgere. È presente un command concreto per ogni operazione.
* Invoker: oggetto che si occupa di invocare i command.
* MacroCommand: interfaccia che rappresenta le operazioni definite dall’utente. Implementato dalla classe OperazioneUtenteMacroCommand, la quale esegue una lista di comandi.

**VIEW:**

interfaccia grafica che visualizza i dati del model e con la quale l’utente interagisce.

**CONTROLLER:**

componente che riceve i comandi dell’utente, attraverso l’interfaccia, e modifica lo stato di model e view. Implementato dalla classe ControllerCalcolatrice.