
PROGETTO ESAME “PROGRAMMAZIONE 3 E LAB.”

# Casa Domotica

Vincenzo D’Alò (0124/1726)

Docenti:

Prof. Angelo Ciaramella, Prof Raffaele Montella

## Descrizione progetto

Si vuole sviluppare un sistema domotico per il monitoraggio di una casa.
Diversi sensori sono installati per il monitoraggio (e.g., temperatura, corrente elettrica, ecc.) o per l'intervento (e.g., getto d'acqua, getto d'aria, ecc.).
Ogni sensore è nello stato acceso o spento e quelli di monitoraggio registrano i valori di riferimento.
L'accesso al sistema può avvenire in modalità collaudo o in modalità attivato.

## Definizione dei requisiti

In modalità collaudo si possono effettuare le seguenti operazioni:

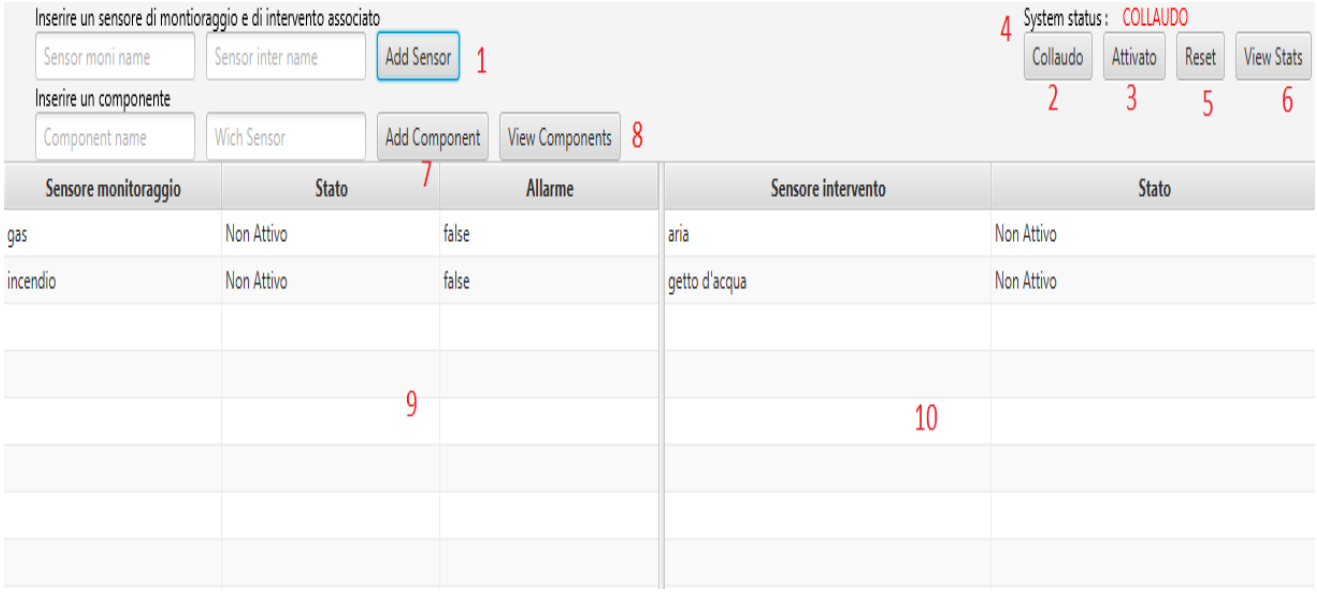
- installare un nuovo sensore
- aggiungere componenti ad un sensore (e.g., ad una telecamera si aggiunge un modulo audio)
- resettare tutti i sensori
- periodicamente è possibile mostrare le statistiche dei sensori (e.g., data e ora dell'allarme)

In modalità attivato si possono effettuare le seguenti operazioni:

- se un sensore per il monitoraggio lancia un allarme viene attivato il corrispondente sensore per l'intervento. In caso di più allarmi vengono gestiti con una politica FIFO
- se un sensore per il monitoraggio cessa l'allarme viene disattivato il corrispondente sensore per l'intervento. In caso di più allarmi vengono gestiti con una politica FIFO

## Interfaccia grafica

Per completezza è fornita un'interfaccia grafica che consente di interagire con la casa domotica.
Tale interfaccia fornisce gli strumenti base per interagire con la casa domotica e effettuando varie operazioni, vedendo in tempo reale il suo contenuto cambiare e come i sensori interagiscano tra loro. In seguito, una breve descrizione delle sue funzionalità:



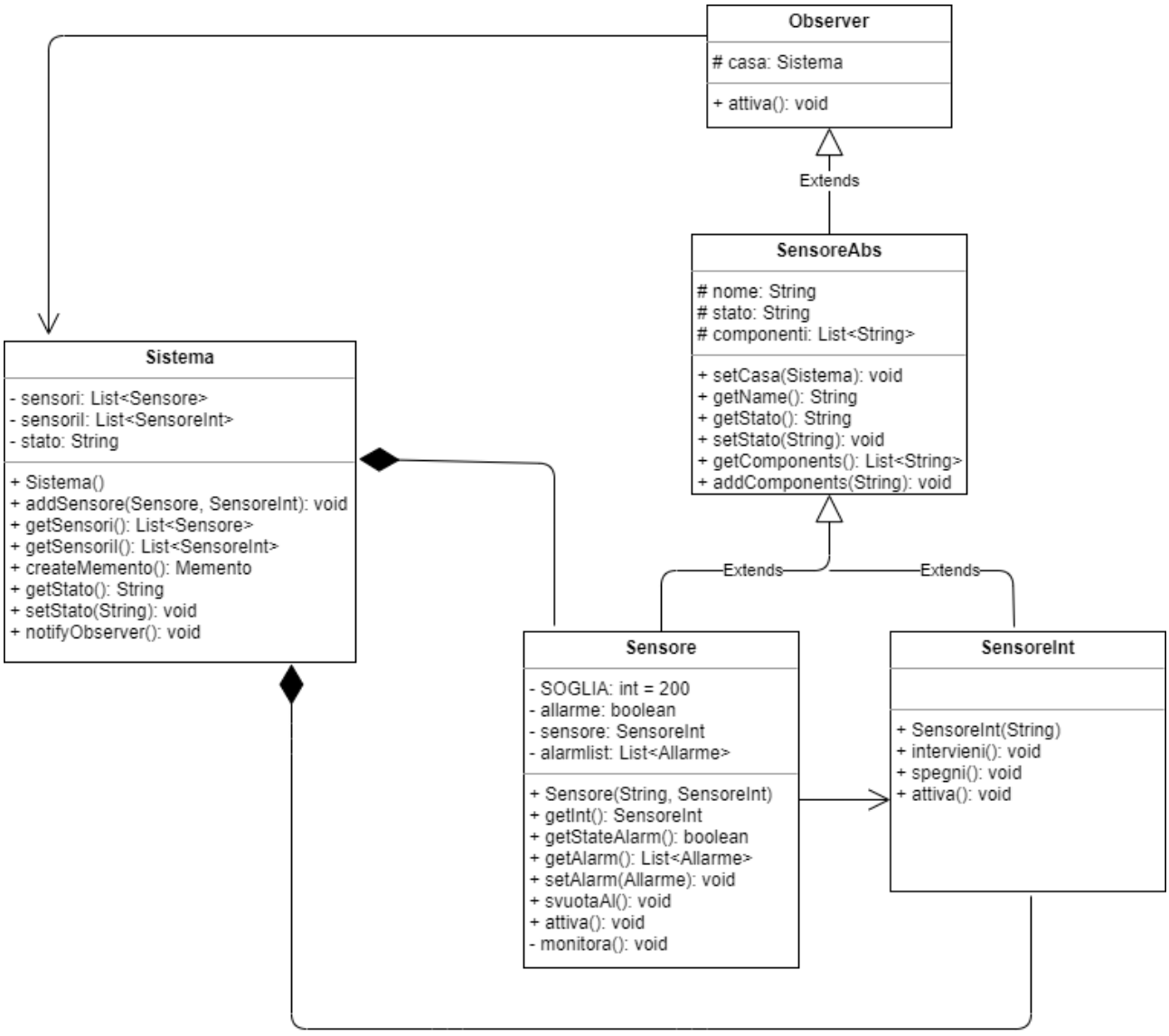

1. Inserisce un sensore di monitoraggio ed uno di intervento nel sistema
2. Imposta lo stato del sistema a “Collaudo”, quindi i sensori saranno disattivati
3. Imposta lo stato del sistema ad “Attivato”, quindi si attivano i sensori e si comincia il monitoraggio
4. Mostra lo stato in tempo reale della casa domotica
5. Ripristina dal sistema tutti i sensori
6. Mostra gli allarmi dei sensori con data e ora in una tabella
7. Aggiunge una componente ad un sensore
8. Mostra i sensori ed eventualmente le sue componenti in una tabella
9. Mostra l’insieme dei sensori di monitoraggio presenti nel sistema
10. Mostra l’insieme dei sensori per l’intervento presenti nel sistema

# Design Pattern utilizzati

Per realizzare tale progetto mi sono servito di quattro pattern: Singleton per le classi che si occupano dei file, quindi per garantire un'istanza unica, Observer per il comportamento dei sensori in funzione dello stato del sistema, Memento per il reset, MVC per l'interfaccia grafica.

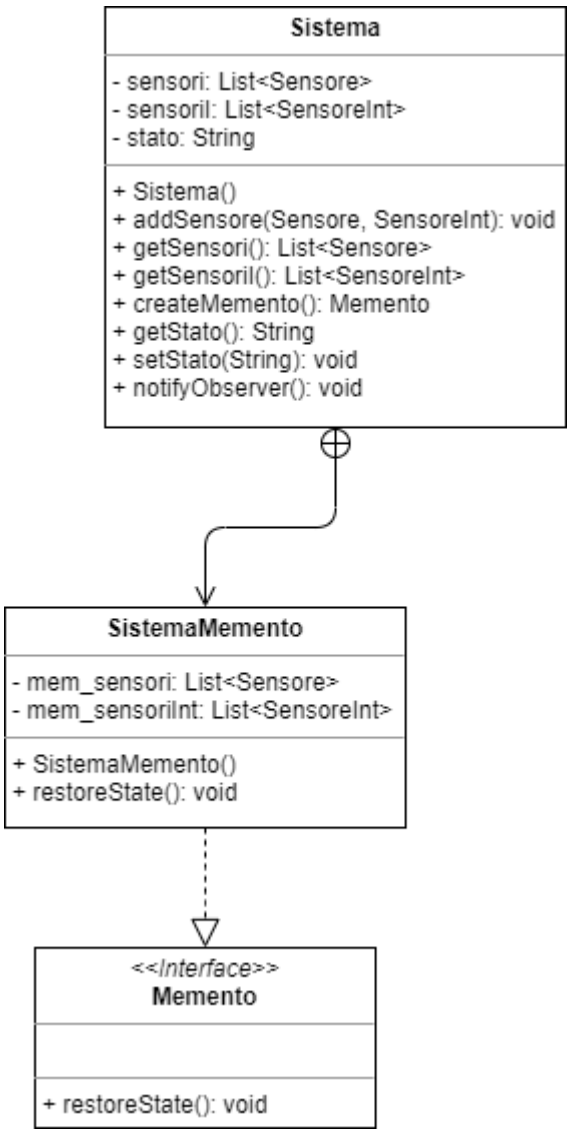
## Observer

Mi sono servito di questo pattern perché è l'ideale quando, al cambiamento di uno stato di un oggetto, segue un comportamento diverso da parte di un altro oggetto. Il caso è proprio questo, infatti, quando il sistema cambierà di stato, i sensori saranno o attivi o disattivi.



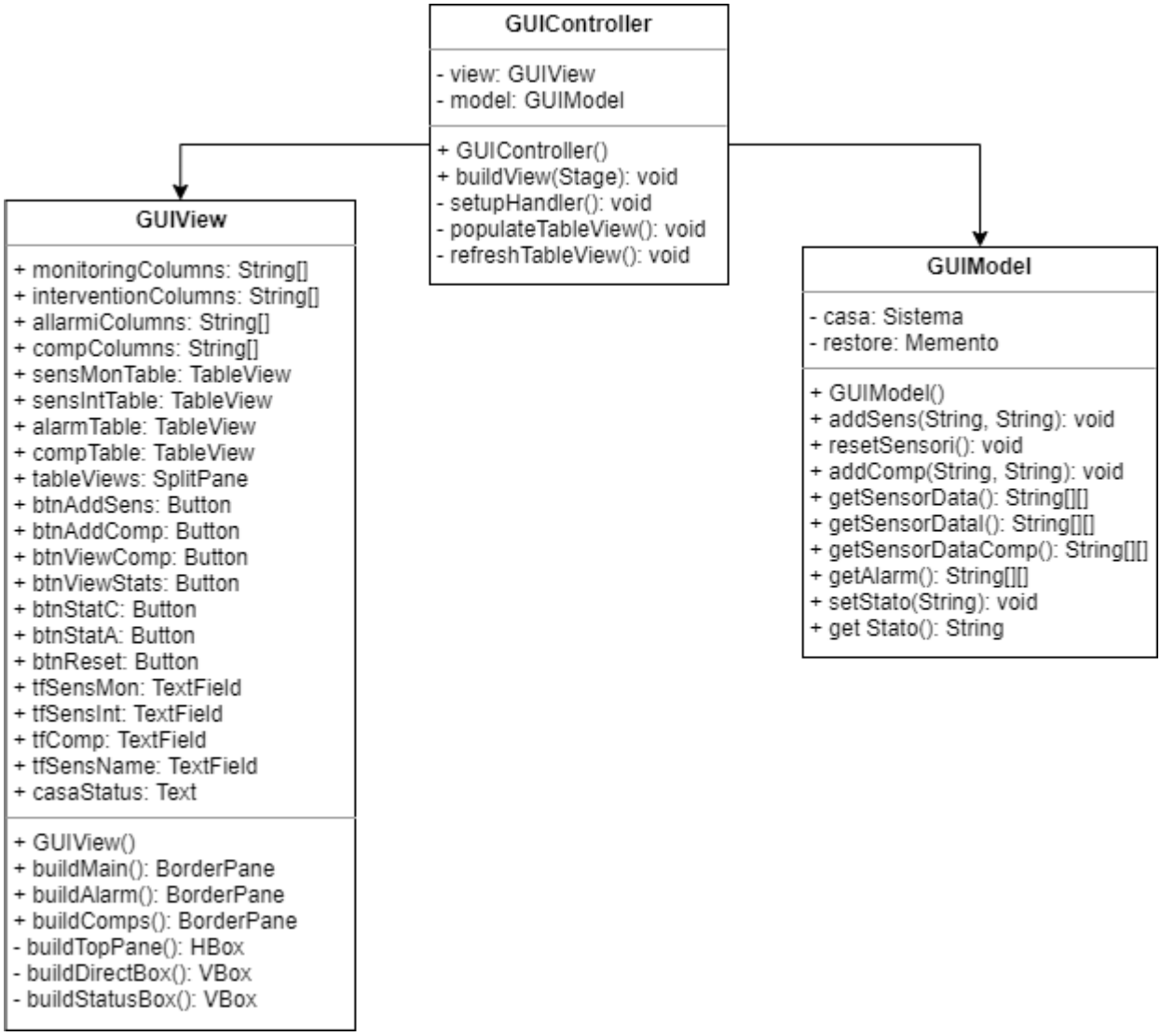
## Memento

Questo pattern è utile quando si vuole ripristinare uno stato precedente all'interno di un oggetto. In questa applicazione viene utilizzato per eseguire un reset dei sensori all'interno del sistema per ritornare allo stato in cui non ci sono sensori nel sistema.

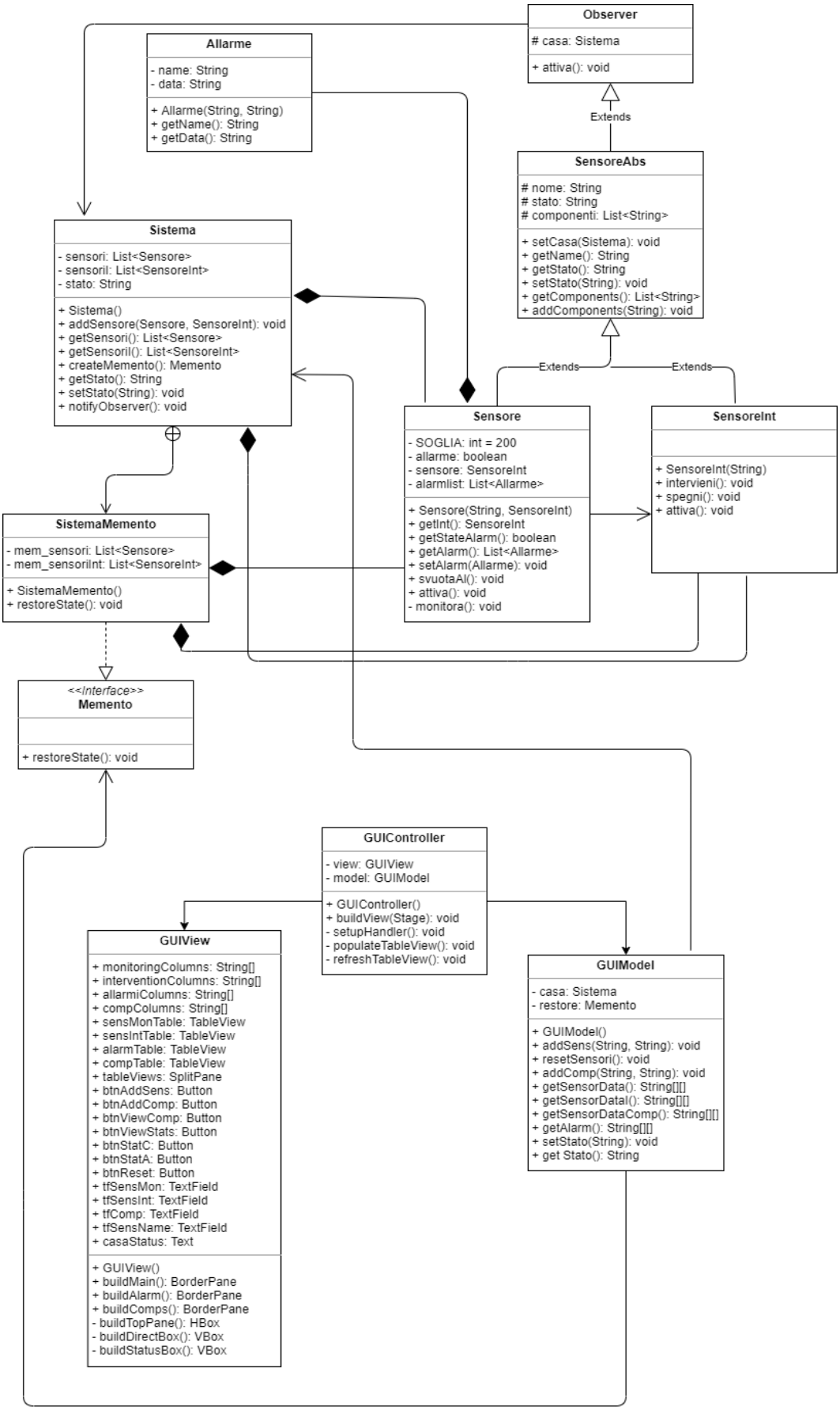


## MVC

Si utilizza il pattern MVC per rendere il codice dell'interfaccia utente più modulare e un codice più pulito.



# Class Diagram



# Dettagli implementativi

## Monitoraggio del sensore

Ogni sensore che viene aggiunto registra valori da 0 a 255, quando questo valore supera 200, la soglia impostata, farà scattare un allarme ed a sua volta si attiverà il sensore per l'intervento associato ad esso. L'allarme verrà aggiunto nella lista degli allarmi e salvato su un file. Una volta che questo valore scenderà sotto 200 l'allarme sarà spento e di conseguenza anche il sensore associato all'intervento.

```
private void monitora () throws IOException, ClassNotFoundException
	{
		int val= (int)(Math.random() * 256);

		if(val>SOGLIA)
		{
			Allarme a = new Allarme(this.getName(),new Date().toString());
			Sensore.alarmList.add(a);
			FileAllarme.getIstance().scriviTesto("allarme.txt", a);
			this.allarme=true;
			this.sensore.intervieni();
		}
		else
		{
			this.allarme=false;
			this.sensore.spegni();
		}
	}
```

## Attivazione del sensore di monitoraggio

Quando lo stato del sistema cambia ad “Attivato”, anche i sensori saranno attivi e di conseguenza parte il monitoraggio, che verrà fatto fino a quando lo stato è “Attivato” ed ogni tre secondi. Quando lo stato cambia a “Collaudo” allora tutti i sensori saranno non attivi.

```
public void attiva()
{
	Runnable runnable = () -> {
	    while(casa.getStato()=="Attivato")
	    {
	       try {
	                    monitora();
	                    try {
	                          Thread.sleep(3000);
	                    } catch (InterruptedException e) {
	                          e.printStackTrace();
	                    }
	               } catch (ClassNotFoundException | IOException e) {
	                    e.printStackTrace();
	               }
	    }
	};
	Thread t = new Thread(runnable);
	if(casa.getStato()=="Attivato")
	{
	       this.stato="Attivo";
	       t.start();
	}
	else
	{
	       this.stato="Non Attivo";
	       this.allarme=false;
	       t.interrupt();
	}

}
```