SAPIENZA Università di Roma
Facoltà di Ingegneria - Corso di Laurea in Ingegneria Informatica
Corso di Progettazione del Software A.A. 2008/2009

Prova al calcolatore (riveduta) del 5 giugno 2009

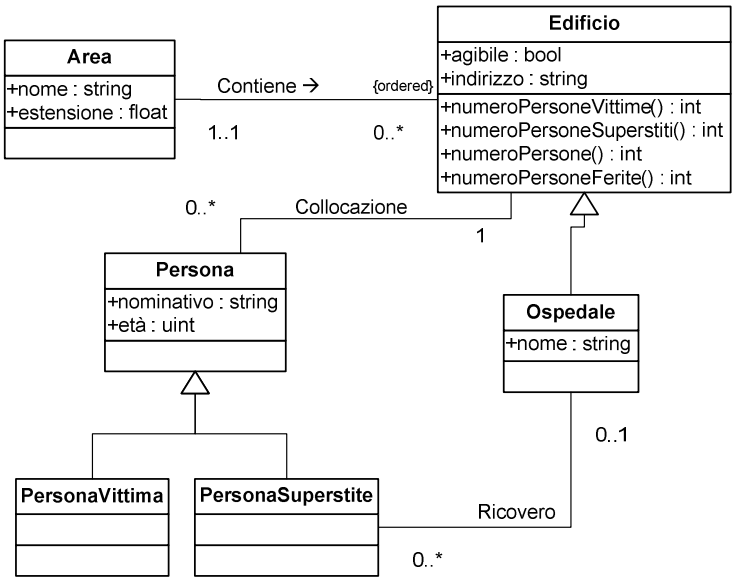
L'applicazione da progettare riguarda la gestione dei terremoti che possono verificarsi in una data zona. L'applicazione ha lo scopo di eseguire un insieme di attività, in un dato ordine, con le seguenti finalità:

- dare il primo supporto alle popolazioni colpite,
- salvare le persone intrappolate negli edifici,
- dirigere le persone verso ospedali più vicini,
- effettuare una prima analisi sull'agibilità degli edifici.

La zona da supervisionare è divisa in aree, ciascuna contenente un insieme di edifici, alcuni dei quali sono ospedali. Di ciascun edificio, in ciascuna area, viene fatto un sopralluogo, per valutarne l'agibilità e determinare il numero di persone che vi sono all'interno. Le persone rimaste negli edifici sono suddivise in superstiti e vittime. Un supertite può essere ferito, nel qual caso viene trasportato in un ospedale. Una persona è considerata ferita se è stata ricoverata in qualche ospedale.

Il processo di supervisione prevede che un operatore della Protezione Civile inserisca le informazioni sulle aree colpite, che vengono memorizzate nel sistema. Successivamente, viene gestita l'emergenza nelle diverse aree. Per velocizzare le operazioni, le aree possono essere gestite in parallelo da diversi operatori.
Inizialmente, l'operatore addetto, non appena è in grado di gestire una certa area, inserisce la lista degli edifici da controllare. Successivamente, viene effettuato il sopralluogo di ogni edificio, valutandone l'agibilità, determinando i relativi superstiti e vittime ed, eventualmente, decidendo presso quali ospedali ricoverare i feriti.
La Protezione Civile è, inoltre, interessata a calcolare alcune statistiche, in particolare: il rapporto vittime/superstiti ed il rapporto feriti/persone (opportunamnete realizzate all'interno di un'attività).

Il diagramma delle classi è raffigurato in Figura 1, insieme alla descrizione della responsabilità sulle associazioni.



Associazione	Classe	Resp.
Contiene	Area	SI
	Edificio	SI
Collocazione	Edificio	SI
	Persona	SI
Ricovero	PersonaSuperstite	SI
	Ospedale	NO

Figura 1: Diagramma delle Classi e relative Responsabilità

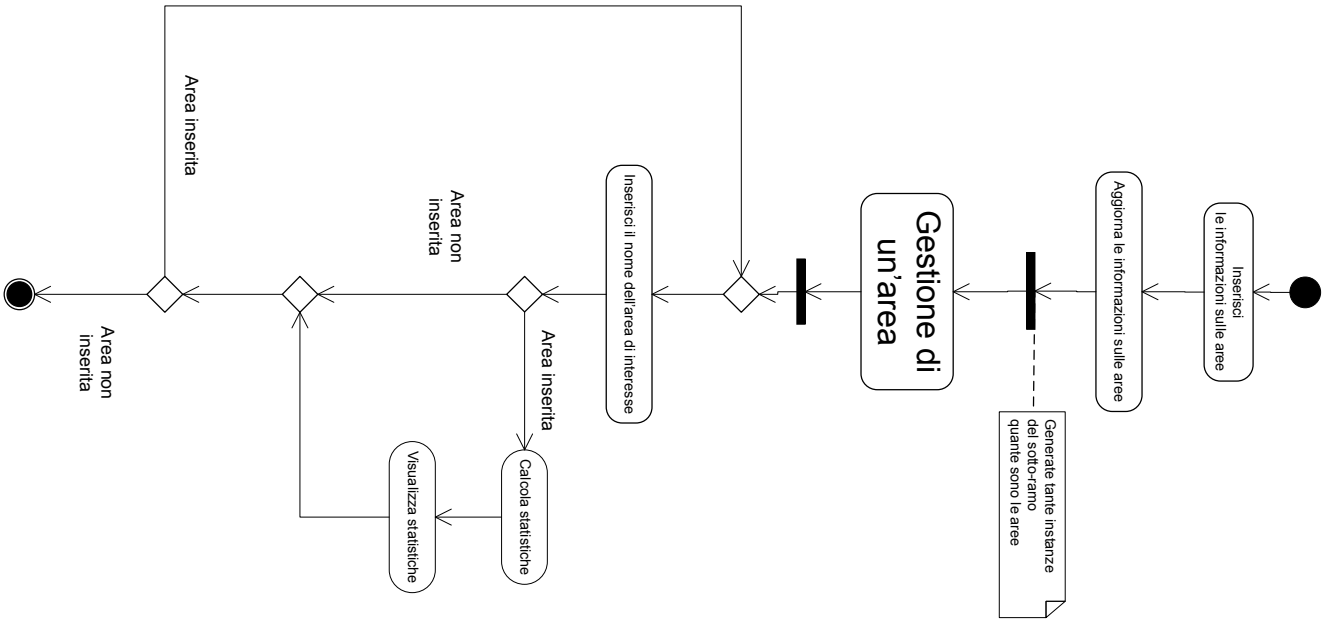


Figura 2: Diagramma dell'attività principale

Tutte le istanze delle classi del diagramma UML vengono inserite in memoria attraverso i metodi statici della classe **emergenza.Modello**. Tale classe fornisce anche i metodi necessari a reperire le istanze aggiunte in precedenza oppure tutte quelle di una data classe. In particolare, la classe assume che non possano esistere due aree con lo stesso nome, due edifici con lo stesso indirizzo, due persone con lo stesso nominativo oppure due ospedali con lo stesso nome. Maggiori dettagli sulla classe Modello sono disponibili come commenti nel codice stesso.

I metodi dichiarati nella classe **emergenza.Edificio** svolgono le seguenti operazioni:

- `numeroPersoneVittime()`, restituisce il numero di vittime nell'edificio
- `numeroPersoneSuperstiti()`, restituisce il numero di superstiti nell'edificio
- `numeroPersoneFerite()`, restituisce il numero di persone ferite estratte dall'edificio

Il diagramma delle attività è rappresentato in Figura 2. L'attività **Gestione di un'area** è composta ed il corrispondente diagramma delle attività è rappresentato in Figura 3.

Le attività utilizzano i seguenti tipi di dato:

- **RecordArea** che definisce due proprietà:
  - **nome: String**
  - **estensione: float**
- **RecordPersona** che definisce due proprietà
  - **nominativo: String**
  - **eta: int**
  - **vivo: boolean**
  - **ospedale: nome**
    - La proprietà **ospedale** può valere anche *null* se la persona in questione è una vittima oppure è superstite ma non ferita. Ovviamente non può accadere che esista una persona per cui `vivo=false` e `ospedale!=null`

A seguire, viene fornita la specifica delle attività. Tutte le classi che realizzano le varie attività sono definite nel package **processo**. Le attività atomiche (I/O e Task) sono codificate in classi aventi le stesso nome usato nella specifica.

Attività di I/O

InizioSpecificaAttivitàAtomica inserisciInfoAree
inserisciInfoAree () : (List<RecordArea>)
pre: --
post: dà all'utente la possibilità di inserire le informazioni relative alle varie aree. result è la lista dei RecordArea contenenti le informazioni inserite.
FineSpecifica

InizioSpecificaAttivitàAtomica inserisciEdifici
inserisciEdifici (nomeArea : String) : (edifIO : List<String>)
pre: --
post: dà all'utente la possibilità di inserire le informazioni relative agli edifici presenti nell'area "nomeArea". result è la lista degli indirizzi degli edifici.
FineSpecifica

InizioSpecificaAttivitàAtomica leggiAbilita
leggiAbilita (indirizzo : String) : (agibile : boolean)
pre: --
post: dà all'utente la possibilità di specificare se l'edificio all'indirizzo "indirizzo" è agibile. result è true se e solo se l'edificio è specificato agibile.
FineSpecifica

InizioSpecificaAttivitàAtomica leggiQuestionario
leggiQuestionario (indirizzo : String) : (recordList: List<RecordPersona>)
pre: --
post: dà all'utente la possibilità di inserire le informazioni relative alle persone all'interno dell'edificio all'indirizzo "indirizzo". result è la lista (di RecordPersona) delle informazioni inserite
FineSpecifica

InizioSpecificaAttivitàAtomica mostraInfoSalienti
mostraInfoSalienti (recordArea: RecordArea) : ()
pre: --
post: Mostra le informazioni relative all'area "recordArea"
FineSpecifica

Task

InizioSpecificaAttivitàAtomica aggiornaInfoAree
aggiornaInfoAree (areeIO: List<RecordArea>) : ()
pre: --
post: per ogni RecordArea presente in areeIO, viene creato un oggetto di classe Area contenente le informazioni all'interno del record.
FineSpecifica

InizioSpecificaAttivitàAtomica registraInfoEdifici
registraInfoEdifici (edifIO: List<String>, area: RecordArea):()
pre: --
post: per ogni indirizzo, rappresentato da una stringa, contenuto in edifIO, crea un edificio con indirizzo contento nella stringa e lo inserisce nell'area "area"
Fine

InizioSpecificaAttivitàAtomica aggiornaAgibilita
aggiornaAgibilita (indirizzo: String, agibilita:boolean)
pre: --
post: imposta ad "agibilità" il valore dell'agibilità dell'edificio situato all'indirizzo "indirizzo"

InizioSpecificaAttivitàAtomica aggiornaInfoEdifici
aggiornaInfoEdifici (indirizzo: String ,quest:RecordPersona)

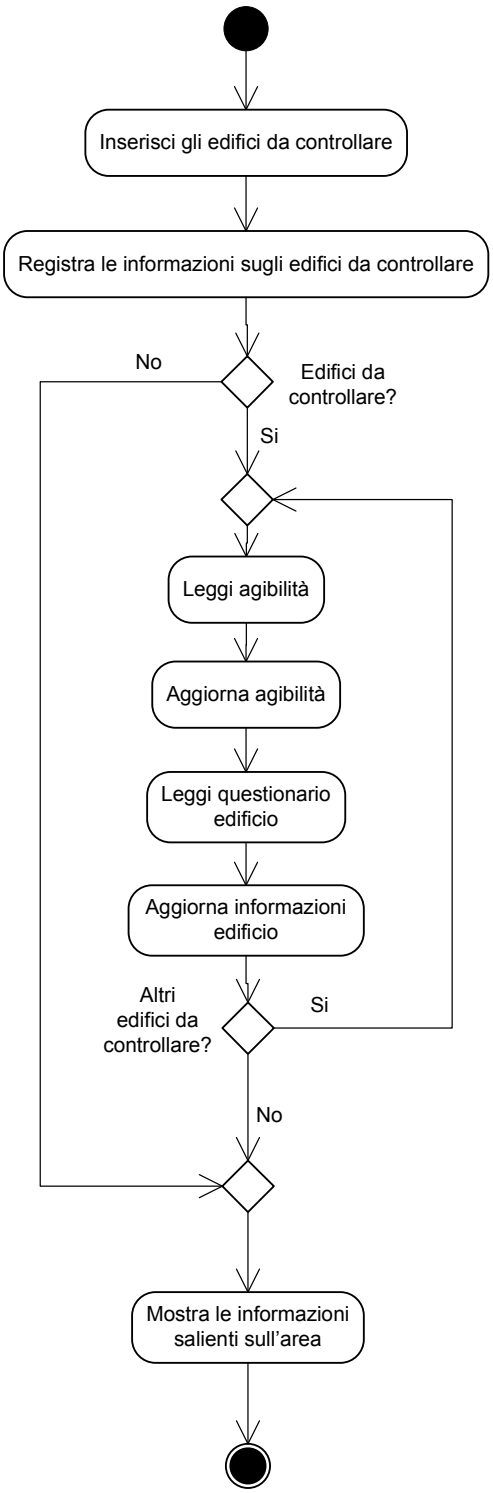
```
pre: --
post: associa ogni persona ad un edificio e, se vivo e ferito, associa tale persona all'ospedale
```

Attività composte

Realizzare la specifica per esercizio.

Il diagramma in Figura 2 è realizzato nella classe **processo.GestioneEmergenza** ed il sotto-ramo è codificato nella classe **processo.GestioneArea**. Quest'ultimo definisce un costruttore che prende come input un oggetto **RecordArea** che identifica l'area di interesse.

- **Al fine di modificare il codice fornito per implementare le funzionalità, si suggerisce di intervenire esclusivamente sulle seguenti classi:**
  - **gui.PrincipaleListener**
  - **gui.Principale**
  - **emergenza.PersonaSuperstite**
  - **emergenza.Edificio**
  - **emergenza.ManagerContiene**
  - **emergenza.TipoLinkRicovero**
  - **processo.GestioneArea**
  - **processo.CalcolaStatistica**
  - **processo.AggiornaInfoAree**
  - **processo.CalcolaStatistica**


Inserisci gli edifici da controllare
Registra le informazioni sugli edifici da controllare
No
Edifici da controllare?
Si
Leggi agibilità
Aggiorna agibilità
Leggi questionario edificio
Aggiorna informazioni edificio
Altri edifici da controllare?
Si
No
Mostra le informazioni salienti sull'area