

Esercitazioni di Progettazione del Software
A.A. 2013/2014

Prova al calcolatore – 30 maggio 2014

Requisiti

Si vuole realizzare un'applicazione per la gestione di condomini.

Un condominio è caratterizzato dal nome e dall'indirizzo, ed include almeno due immobili. Di ogni immobile interessano i metri quadri, l'interno, il piano e la quota millesimale.¹ Gli immobili che costituiscono un condominio si dividono in abitazioni (di cui interessa memorizzare il numero di vani), uffici, e negozi (di cui interessa il nome).

Ogni immobile ha almeno un proprietario, con la relativa quota percentuale di possesso. In caso di singolo proprietario, la quota di possesso è pari a 100; in presenza di più proprietari, la somma delle quote di possesso deve essere pari a 100. Di ogni proprietario interessano il nome, il cognome, il codice fiscale, ed eventuali recapiti telefonici; un proprietario può possedere più immobili.

Un condominio sostiene ogni anno delle spese, distinte in spese ordinarie (es. luce, pulizia scale) o straordinarie (es. rifacimento facciata). Di ogni spesa sostenuta interessano il codice con cui è stata registrata, l'importo, l'anno di riferimento, e la descrizione. In caso di spese straordinarie, è di interesse conoscere la data in cui tale spesa è stata approvata.

In Figura 1 è mostrato il diagramma delle classi corrispondente al dominio.

Tra le diverse funzionalità a disposizione degli amministratori, si vuole realizzare la seguente attività principale di analisi e ripartizione spese. L'attività prende in input un insieme di condomini e si svolge come segue:

- il sistema mostra la lista dei condomini e l'amministratore seleziona un condominio di interesse;
- per il condominio selezionato, il sistema mostra gli anni per cui sono disponibili informazioni sulle spese sostenute, e l'amministratore seleziona un anno di interesse;

¹esprime in millesimi il valore di ciascuna unità immobiliare, in rapporto al valore dell'intero edificio, che è posto pari al numero 1000.

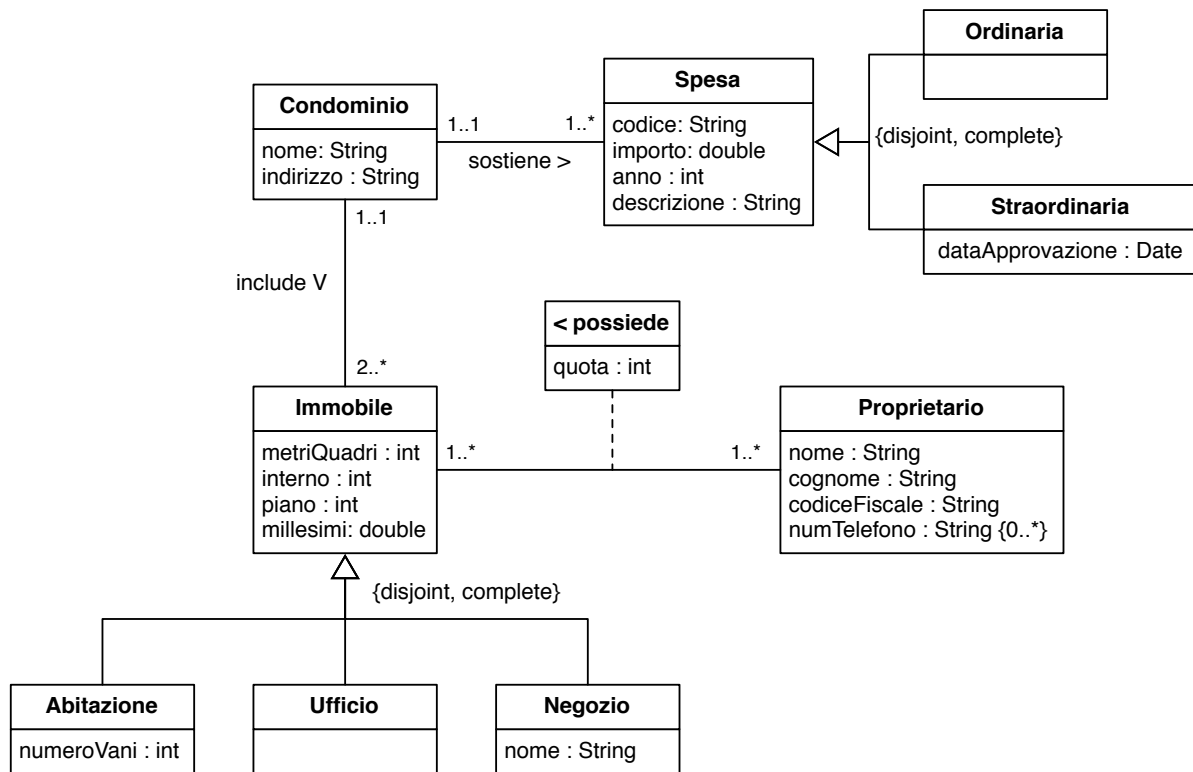


Figura 1: Diagramma UML delle classi

- sulla base del condominio selezionato e dell'anno di riferimento, vengono eseguite concorrentemente due sottoattività che calcolano rispettivamente l'importo totale delle spese ordinarie e l'importo totale delle spese straordinarie sostenute dal condominio nell'anno di riferimento;
- una volta che tali sottoattività sono state completate, gli importi calcolati vengono visualizzati;
- se le spese straordinarie superano le spese ordinarie, viene visualizzato un messaggio di notifica per l'amministratore;
- si procede poi con le seguenti sottoattività concorrenti:
 1. si calcola la spesa totale sostenuta dal condominio (come somma tra le spese ordinarie e straordinarie) e tale spesa viene poi ripartita tra gli immobili sulla base dei millesimi di competenza;
 2. si determina l'insieme dei proprietari degli immobili inclusi nel condominio che posseggono immobili anche in un condominio diverso da quello in considerazione, e tale insieme viene visualizzato;
- viene visualizzata poi una schermata riassuntiva con la ripartizione di spesa, mostrando per ogni immobile la quota di competenza.

In Figura 2 è riportato il diagramma delle attività corrispondente.

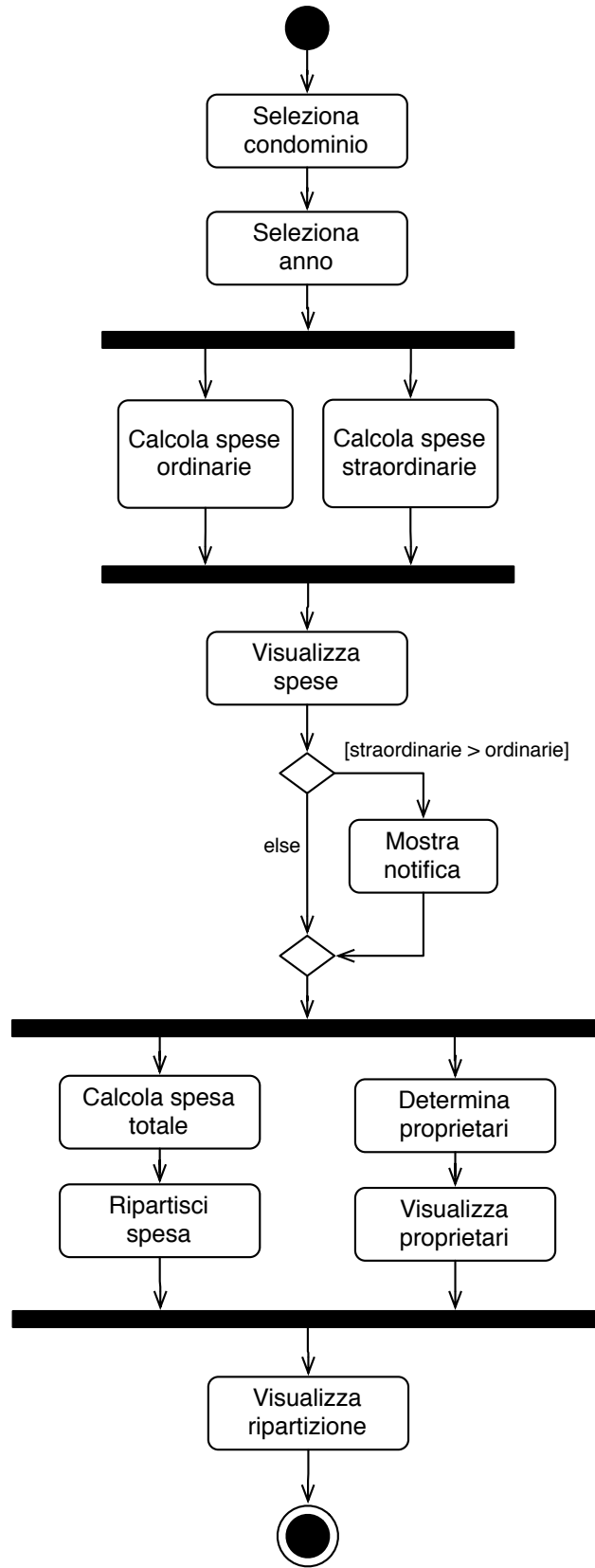


Figura 2: Diagramma delle attività

La prova consiste nel completare o modificare il codice fornito insieme al testo, in modo da soddisfare i requisiti sopra riportati. Seguendo le indicazioni riportate nei commenti al codice², si chiede di intervenire sulle seguenti classi:

- Condominio (package `app.dominio`)
- ManagerSostiene (package `app.dominio`)
- CalcolaSpeseStraordinarie (package `app.attivita.atomiche`)
- RipartisciSpesa (package `app.attivita.atomiche`)
- AttivitaSottoramo2_2 (package `app.attivita.complesse`)

Tempo a disposizione: **3 ore**.

Gli elaborati non accettati dal compilatore saranno considerati insufficienti.

Per facilitare la comprensione del codice e lo svolgimento della prova, nel seguito sono riportati i documenti di specifica risultanti dalle fasi di analisi e di progetto.

²le porzioni di codice su cui intervenire sono identificate dal commento `/* DA COMPLETARE A CURA DELLO STUDENTE */`

Analisi

Attività di I/O

InizioSpecificaAttivitàAtomica SelezionaCondominio

SelezionaCondominio (sc:Set<Condominio>):(Condominio)

pre: --

post: Mostra l'insieme *sc* dei condomini e consente all'amministratore di selezionarne uno.
result è il condominio selezionato.

FineSpecifica

InizioSpecificaAttivitàAtomica SelezionaAnno

SelezionaAnno (c:Condominio):(Integer)

pre: --

post: Visualizza tutti gli anni in cui il condominio *c* ha effettuato almeno una spesa, e consente all'amministratore di selezionarne uno.
result è l'anno selezionato.

FineSpecifica

InizioSpecificaAttivitàAtomica VisualizzaSpese

VisualizzaSpese (so:Double, ss:Double):()

pre: --

post: Visualizza gli importi *so* e *ss* corrispondenti alle spese ordinarie e straordinarie totali.

FineSpecifica

InizioSpecificaAttivitàAtomica MostraNotifica

MostraNotifica ():()

pre: --

post: Mostra un messaggio che informa l'amministratore che le spese straordinarie superano le spese ordinarie.

FineSpecifica

InizioSpecificaAttivitàAtomica VisualizzaProprietari

VisualizzaProprietari (sp:Set<Proprietario>):()

pre: --

post: Visualizza l'insieme *sp* di proprietari che possiedono immobili anche in altri condomini.

FineSpecifica

InizioSpecificaAttivitàAtomica VisualizzaRipartizione

VisualizzaRipartizione (rip:Map<Immobile,Double>):()

pre: --

post: Visualizza le ripartizioni di spesa rappresentate nella struttura dati *rip*.

FineSpecifica

Attività Atomiche

InizioSpecificaAttivitàAtomica CalcolaSpeseOrdinarie

CalcolaSpeseOrdinarie (c:Condominio, anno:Integer) : (Double)

pre: --

post: Calcola l'importo totale delle spese ordinarie sostenute dal condominio **c** nell'anno **anno**.

In particolare, sia S_o l'insieme delle spese ordinarie s legate al condominio **c** da un link di tipo *Sostiene* e tali che $s.anno == anno$. **result** è pari a

$$\sum_{s \in S_o} s.importo$$

FineSpecifica

InizioSpecificaAttivitàAtomica CalcolaSpeseStraordinarie

CalcolaSpeseStraordinarie (c:Condominio, anno:Integer) : (Double)

pre: --

post: Calcola l'importo totale delle spese straordinarie sostenute dal condominio **c** nell'anno **anno**.

In particolare, sia S_{so} l'insieme delle spese straordinarie s legate al condominio **c** da un link di tipo *Sostiene* e tali che $s.anno == anno$. **result** è pari a

$$\sum_{s \in S_{so}} s.importo$$

FineSpecifica

InizioSpecificaAttivitàAtomica CalcolaSpesaTotale

CalcolaSpeseStraordinarie (so:Double, ss:Double) : (Double)

pre: --

post: Calcola l'importo totale delle spese sostenute, come somma tra le spese ordinarie **so** e quelle straordinarie **ss**.

result è pari a **so + ss**.

FineSpecifica

InizioSpecificaAttivitàAtomica RipartisciSpesa

InizializzaSimulazione(c:Condominio, st:Double):(Map<Immobile,Double>)

pre: --

post: Ripartisce la spesa totale **st** tra gli immobili del condominio **c**, sulla base dei millesimi assegnati a ciascun immobile. In particolare, sia I l'insieme degli immobili collegati al condominio **c** da un link di tipo *Include*.

Per ogni immobile $imm \in I$, la quota di spesa di competenza è pari a

$$q_{imm} = \frac{st \times imm.millesimi}{1000}$$

result è l'insieme di coppie $\langle imm, q_{imm} \rangle$ così determinate.

FineSpecifica

InizioSpecificaAttivitàAtomica DeterminaProprietari

InizializzaSimulazione(c:Condominio):(Set<Proprietario>)

pre: --

post: Determina l'insieme **result** costituito dai proprietari che possiedono immobili del condominio **c** e possiedono anche immobili in altri condomini.

In particolare, sia I l'insieme degli immobili legati al condominio **c** da un link di tipo *Include*, e sia P l'insieme dei proprietari legati agli immobili in I da link di tipo *possiede*. L'insieme **result** $\subseteq P$ è costituito dai proprietari $prop \in P$ per i quali esiste un link di tipo *possiede* con un immobile imm coinvolto in un link di tipo *include* con un condominio c' , con $c' \neq c$ (cioè il proprietario possiede almeno un immobile che non è tra quelli inclusi nel condominio **c**).

FineSpecifica

Attività Composte

InizioSpecificaAttività AttivitaSottoramo1_1

```
AttivitaSottoramo1_1(c:Condominio, anno:int):(Double)
```

Variabili Processo:

```
condominio:Condominio -- il condominio corrente  
anno:int -- l'anno di riferimento  
speseOrdinarie:double -- l'importo totale delle spese ordinarie
```

Inizio Processo

```
CalcolaSpeseOrdinarie(condominio, anno):(speseOrdinarie);
```

Fine Processo

FineSpecifica

InizioSpecificaAttività AttivitaSottoramo2_1

```
AttivitaSottoramo2_1(c:Condominio, anno:int):(Double)
```

Variabili Processo:

```
condominio:Condominio -- il condominio corrente  
anno:int -- l'anno di riferimento  
speseStraordinarie:double -- l'importo totale delle spese ordinarie
```

Inizio Processo

```
CalcolaSpeseStraordinarie(condominio, anno):(speseStraordinarie);
```

Fine Processo

FineSpecifica

InizioSpecificaAttività AttivitaSottoramo1_2

```
AttivitaSottoramo1_2(c:Condominio, speseOrdinarie:Double, speseStraordinarie:Double):(Map<Immobile, Double>)
```

Variabili Processo:

```
condominio:Condominio -- il condominio corrente  
speseOrdinarie:Double -- le spese ordinarie  
speseStraordinarie:Double -- le spese straordinarie  
spesaTotale:Double -- la spesa totale  
result:Map<Immobile, Double> -- la ripartizione di spesa tra gli immobili
```

Inizio Processo

```
CalcolaSpesaTotale(speseOrdinarie, speseStraordinarie):(spesaTotale);  
RipartisciSpesa(condominio, spesaTotale):(result);
```

Fine Processo

FineSpecifica

InizioSpecificaAttività AttivitaSottoramo2_2

```
AttivitaSottoramo2_2(c:Condominio):()
```

Variabili Processo:

```
condominio:Condominio -- il condominio corrente  
proprietari:Set<Proprietario> -- l'insieme dei proprietari
```

```

Inizio Processo
    DeterminaProprietari(condominio):(proprietari);
    VisualizzaProprietari(proprietari):();
Fine Processo

```

FineSpecifica

InizioSpecificaAttività AttivitaPrincipale

```

AttivitaPrincipale(condomini:Set<Condominio>):()

```

Variabili Processo:

```

    insiemeCondomini:Set<Condominio> -- l'insieme dei condomini
    condominio:Condominio -- il condominio selezionato
    anno:int -- l'anno selezionato
    spesaOrdinaria:double -- il totale delle spese ordinarie
    spesaStraordinaria:double -- il totale delle spese straordinarie
    ripartizioneSpesa:Map<Immobile, Double> -- la ripartizione di spesa tra gli immobili

```

Inizio Processo:

```

    SelezionaCondominio(insiemeCondomini):(condominio);

    SelezionaAnno(condominio):(anno);

    fork {
        thread t1_1:{AttivitaSottoramo1_1(condominio, anno):(spesaOrdinaria);}
        thread t2_1:{AttivitaSottoramo2_1(condominio, anno):(spesaStraordinaria);}
    }
    join t1_1, t2_1;

    VisualizzaSpese(spesaOrdinaria, spesaStraordinaria):();

    if (spesaStraordinaria > spesaOrdinaria) {
        MostraNotifica():();
    }

    fork {
        thread t1_2:{AttivitaSottoramo1_2(condominio,spesaOrdinaria,spesaStraordinaria):(ripartizioneSpesa);}
        thread t2_2:{AttivitaSottoramo2_2(condominio):();}
    }
    join t1_2, t2_2;

    VisualizzaRipartizione(ripartizioneSpesa):();

```

FineSpecifica

Progetto

Responsabilità sulle Associazioni

R: Requisiti; O: Specifica delle Operazioni/Attività; M: Vincoli di Molteplicità

Associazione	Classe	Ha Responsabilità
Include	Condominio Immobile	SI (M,O,R) SI (M,O)
Sostiene	Condominio Spesa	SI (M,O,R) SI (M)
Possiede	Proprietario Immobile	SI (M,O) SI (M,O,R)

Strutture di Dati

Rappresentiamo le collezioni omogenee di oggetti mediante le classi `Set` ed `HashSet` del `Collection Framework` di Java.

Tabelle di Gestione delle Proprietà delle Classi UML

Riassumiamo le scelte differenti da quelle di default mediante la tabella delle proprietà immutabili.

Classe UML	Proprietà Immutabile			
Condominio	nome	indirizzo		
Immobile	metriQuadri	interno	piano	
Spesa	codice	importo	anno	descrizione
Straordinaria	dataApprovazione			
Proprietario	nome	cognome	codiceFiscale	

Altre Considerazioni

Non dobbiamo assumere una particolare sequenza di nascita degli oggetti.

Non esistono valori di default per qualche proprietà che siano validi per tutti gli oggetti.