

SAPIENZA Università di Roma

A.A. 2008-2009

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

Esercitazioni di Progettazione del Software
(Canale A-L)

Diagrammi delle Attività

Fabio Patrizi

Requisiti

Si vuole realizzare un'applicazione per gestire le riparazioni di autovetture in un'officina. Di ciascun veicolo, interessano: la targa (una stringa) ed il proprietario (nome e cognome memorizzati in una stessa stringa). Ciascuna autovettura è coinvolta in almeno una riparazione ed ogni riparazione coinvolge esattamente un'autovettura. Per quanto riguarda le riparazioni, sono d'interesse: l'importo (reale), la descrizione (una stringa), il fatto che la carrozzeria sia danneggiata o meno (un booleano), il fatto che le parti meccaniche siano danneggiate o meno (booleano), e la data di accettazione della vettura.

L'attività di riparazione di ogni vettura si svolge come segue. Innanzitutto, quando l'automobile arriva in officina, viene aperta la pratica: un operatore compila un formulario con i dati del veicolo e del proprietario, i dati dell'autovettura vengono registrati ed una riparazione ad essa relativa viene creata, con la data corrente come data di accettazione. Dopodiché, la riparazione è eseguita in maniera iterativa, seguendo i passi elencati sotto.

1. Un meccanico effettuata una valutazione dei danni, ispezionando il veicolo ed inserendo in un formulario la descrizione delle riparazioni necessarie, oltre ad indicare se la carrozzeria o la meccanica (o entrambe) siano interessate da qualche riparazione. Tali dati sono poi inseriti nel sistema, aggiornando la riparazione relativa alla vettura.
2. I ricambi necessari alla riparazione vengono acquistati. Il loro costo complessivo, aumentato del 20% per la manodopera, costituisce l'importo della riparazione. L'acquisto dei ricambi e l'aggiornamento dell'importo vengono effettuati automaticamente dal sistema (simulare il processo, ad es., restituendo un valore randomico del costo dei ricambi).

3. Due meccanici, uno specializzato in carrozzerie ed uno in meccanica, eseguono due mansioni contemporanee: se necessario, intervengono sulla parte di loro competenza e compilano un formulario con i dati relativi alla loro attività. In particolare, indicano se la parte di loro competenza sia ancora danneggiata o meno. il sistema viene quindi aggiornato con i dati inseriti. Si osservi che i due meccanici riparano ed inseriscono i dati indipendentemente l'uno dall'altro, eventualmente in contemporanea.
4. Se la carrozzeria o la meccanica sono ancora danneggiate, viene effettuata una nuova iterazione, a partire dal punto 1., altrimenti la vettura è riparata e può essere consegnata al cliente.

Svolgere le fasi di analisi e di progetto, e realizzare la specifica sopra descritta, usando il codice pubblicato, comprendente il framework per la gestione delle attività visto a lezione ed una libreria per le attività di I/O.

Diagramma UML delle Classi

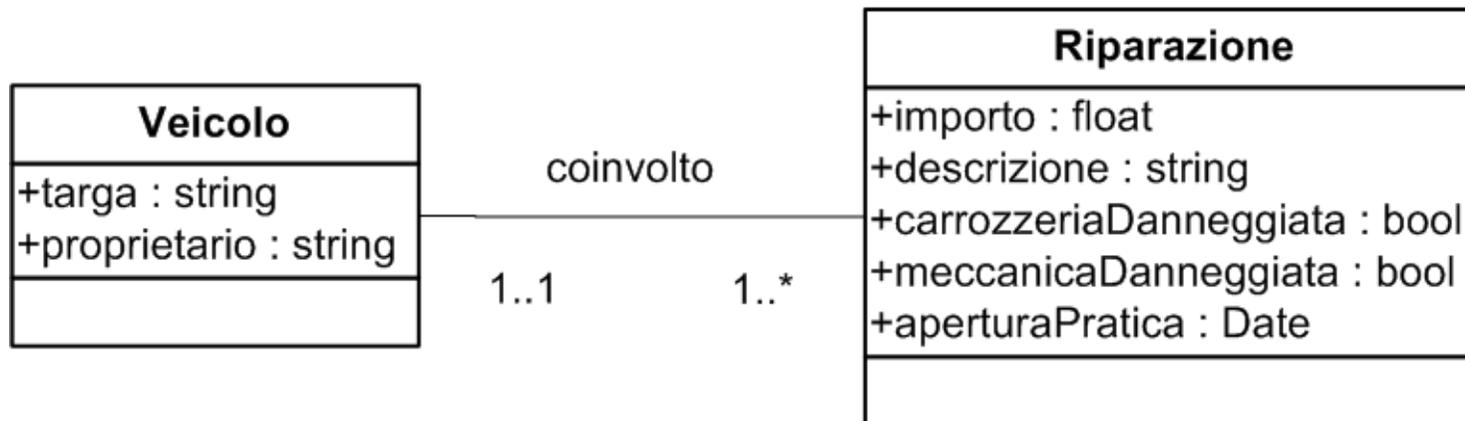
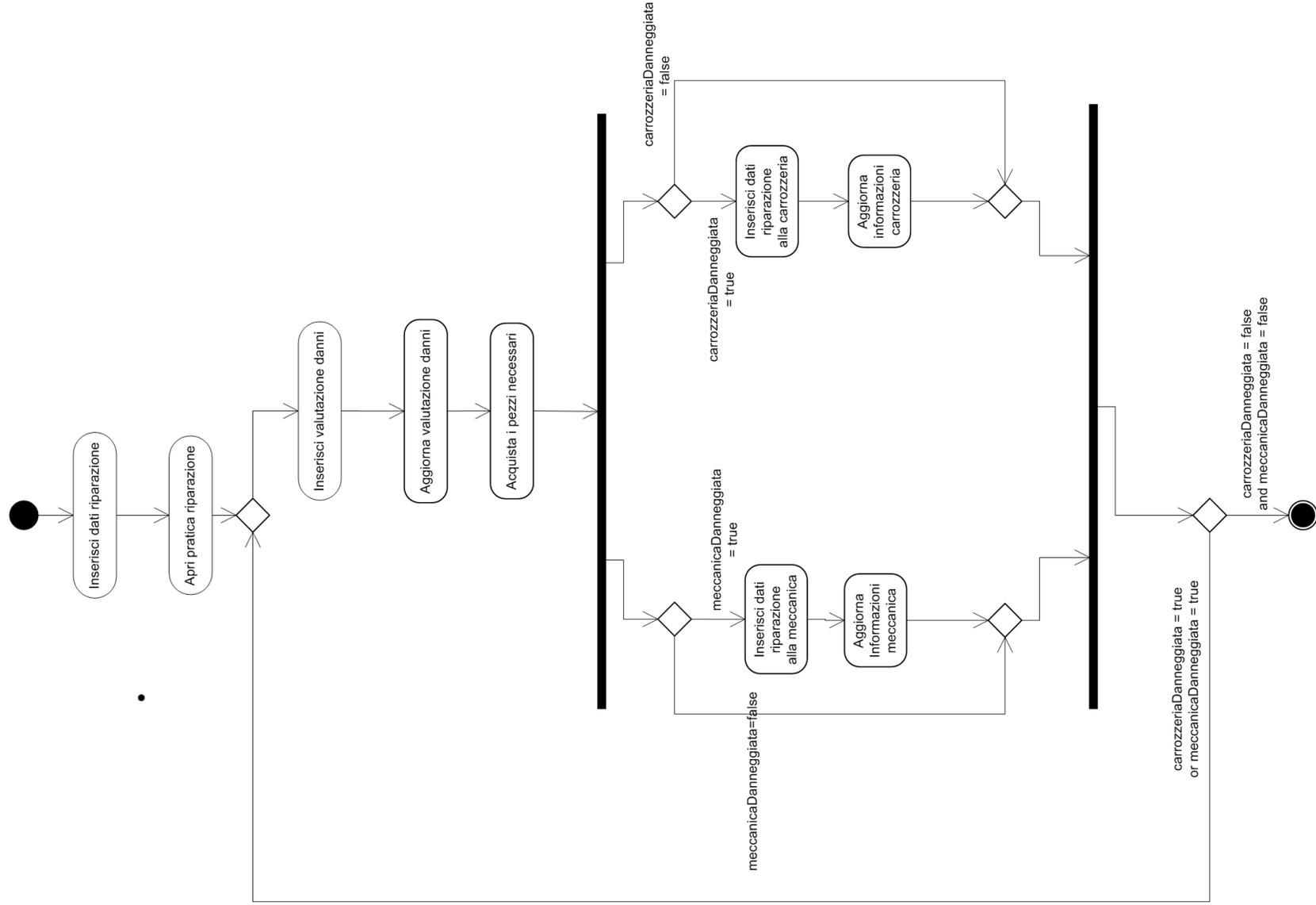


Diagramma delle Attività



Specifica dell'Attività Principale

Si compone delle seguenti definizioni, fornite nel seguito:

- Definizione della segnatura delle operazioni di input/output
- Definizione delle variabili dell'attività
- Specifica delle sottoattività

Definizione delle Operazioni Input/Output

InizioSpecificaOperazioniIO AttivitaPrincipale

operazione: leggiDatiAccettazione

-- se veicolo dato in input al processo, questa operazione è inutile, perché l'unica cosa da fare è creare
-- una nuova riparazione (con campi qualsiasi) e linkarla al veicolo (v. attività AperturaPratica)

in: --

out: autovettura: Veicolo

operazione: leggiValutazioneDanni

in: riparazione: Riparazione

out: pratica: RecordPratica

operazione: leggiRiparazioniMeccanica

in: riparazione: Riparazione

out: pratica: RecordPratica

operazione: leggiRiparazioniCarrozzeria

in: riparazione: Riparazione

out: pratica: RecordPratica

FineSpecifica

Definizione delle Variabili dell'Attività Principale

InizioSpecificaVariabiliAttività AttivitaPrincipale

var: autovettura: Veicolo

var: riparazione: Riparazione

var: datiPratica: RecordPratica

FineSpecifica

Specifica delle Attività Atomiche

InizioSpecificaAttività InserisciDatiAccettazione

```
-- Se vettura presa da processo, questa attività non serve
-- Legge da input i dati relativi ad una vettura
pre: --
post: ^leggiDatiAccettazione and autovettura = leggiDatiAccettazione.out.autovettura
```

FineSpecifica

Per semplicità, assumiamo di avere a disposizione una funzione dataCorrente che restituisce la data corrente

InizioSpecificaAttività AperturaPratica

Crea una pratica associata alla vettura corrente ed inserisce il link tra la pratica e la vettura

```
pre: --
post: Veicolo.allInstances() = Veicolo@pre.allInstances()->including(autovettura) and
      nuovaRiparazione.oclIsNew() and Riparazione.allInstances() =
      Riparazione@pre.allInstances()->including(nuovaRiparazione) and
      ^dataCorrente and nuovaRiparazione.aperturaPratica = dataCorrente.out.data and
      nuovoLink.oclIsNew() and coinvolto = coinvolto@pre.including(nuovoLink) and
      nuovoLink.Veicolo = autovettura and nuovoLink.Riparazione = nuovaRiparazione and
      riparazione = nuovaRiparazione
```

FineSpecifica

InizioSpecificaAttività InserisciValutazioneDanni

-- legge da input i dati relativi alla valutazione dei danni

```
pre: --
post: leggiValutazioneDanni.in = riparazione and ^leggiValutazioneDanni and
      datiPratica.descrizione = leggiValutazioneDanni.out.riparazione.descrizione and
      datiPratica.carrozzeriaDanneggiata = leggiValutazioneDanni.out.riparazione.carrozzeriaDanneggiata and
      datiPratica.meccanicaDanneggiata = leggiValutazioneDanni.out.riparazione.meccanicaDanneggiata
```

FineSpecifica

Specifica delle Attività Atomiche (2)

InizioSpecificaAttività AggiornaDanni

```
-- aggiorna la riparazione corrente con i dati letti da input
pre: --
post: riparazione.descrizione = datiPratica.descrizione and
post: riparazione.carrozzeriaDanneggiata = datiPratica.carrozzeriaDanneggiata and
post: riparazione.meccanicaDanneggiata = datiPratica.meccanicaDanneggiata
```

FineSpecifica

InizioSpecificaAttività AcquistaRicambi

```
-- Simula l'acquisto dei ricambi ed aggiorna conseguentemente l'importo della riparazione
pre: --
post: riparazione.importo = RANDOM
```

FineSpecifica

InizioSpecificaAttività InserisciDatiRiparazioneMeccanica

```
-- Legge da inputi i dati relativi alla riparazione della meccanica
pre: --
post: leggiRiparazioniMeccanica.in = riparazione and ^leggiRiparazioniMeccanica and
      datiPratica.meccanicaDanneggiata = leggiRiparazioniMeccanica.out.pratica.meccanicaDanneggiata
```

FineSpecifica

Specifica delle Attività Atomiche (3)

InizioSpecificaAttività AggiornaMeccanica

```
-- aggiorna la riparazione corrente con i dati della riparazione alla meccanica  
pre: --  
post: riparazione.meccanicaDanneggiata = datiPratica.meccanicaDanneggiata
```

FineSpecifica

InizioSpecificaAttività InserisciDatiRiparazioneCarrozzeria

```
-- Legge da inputi i dati relativi alla riparazione della carrozzeria  
pre: --  
post: leggiRiparazioniCarrozzeria.in = riparazione and ^leggiRiparazioniMeccanica and  
      datiPratica.carrozzeriaDanneggiata = leggiRiparazioniMeccanica.out.pratica.carrozzeriaDanneggiata
```

FineSpecifica

InizioSpecificaAttività AggiornaCarrozzeria

```
-- aggiorna la riparazione corrente con i dati della riparazione alla carrozzeria  
pre: --  
post: riparazione.carrozzeriaDanneggiata = datiPratica.carrozzeriaDanneggiata
```

FineSpecifica

Progetto

Lasciato per esercizio agli studenti

Realizzazione

Vedere codice pubblicato