

La prova si svolge a libri chiusi (non è permessa la consultazione di materiale didattico).

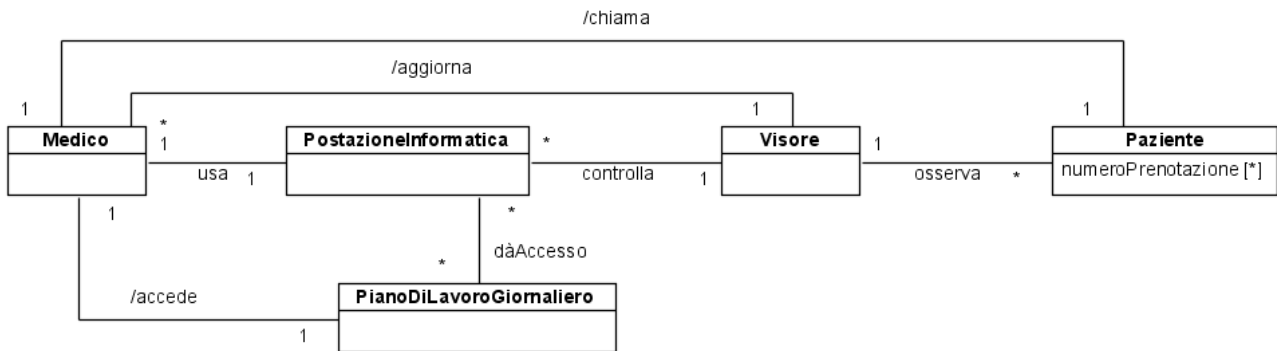
Traccia delle soluzioni

Si consideri il caso di studio Poliambulatorio.

Domanda 1. Analisi del dominio.

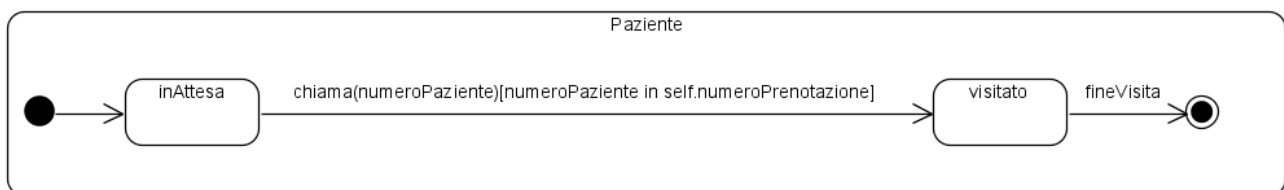
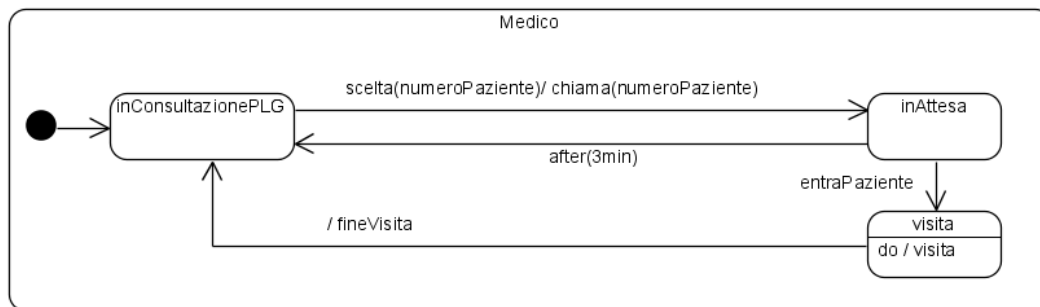
Dare un diagramma delle classi che descriva la parte di dominio rilevante per la funzionalità del sottosistema di supporto alle attività ambulatoriali che permette a un medico di chiamare i pazienti (nel seguito indicata come ChiamataPaziente). Si richiede di mettere in evidenza come le associazioni più astratte (come l’aggiornamento del visore) siano *derivabili* da quelle più concrete (come l’uso della postazione informatica che controlla il visore).

Risposta. Un possibile diagramma è il seguente.



Domanda 2. Dare due macchine a stati finiti che descrivano l’evoluzione di Medico e Paziente, rispettivamente, nel processo di chiamata del paziente. Si fissi un time-out di 3 minuti per l’attesa del paziente chiamato. Si assuma che dallo stato iniziale il medico passi direttamente in “inConsultazionePLG” e il paziente passi in “inAttesa”.

Risposta.

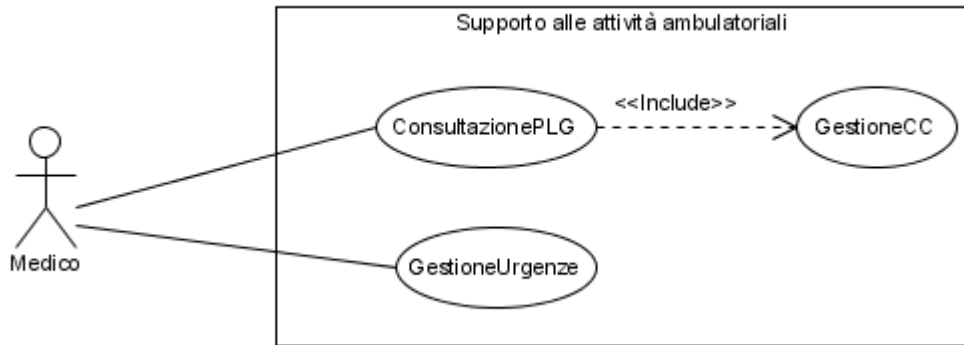


Si consideri il sottosistema **Supporto alle attività ambulatoriali**, con i casi d'uso:

ConsultazionePLG: permette al medico di consultare il Piano di Lavoro Giornaliero, chiamare il prossimo paziente in attesa e aprire la sua Cartella Clinica.

GestioneCC: permette al medico di accettare il paziente (mostra la sua anamnesi personale e familiare), modificare (o creare) la scheda di anamnesi, compilare il referto, stampare il referto.

GestioneUrgenze: permette di modificare il piano di lavoro giornaliero per inserire una visita d'urgenza. Il corrispondente diagramma è il seguente:



L'esperienza ha mostrato che la scelta fatta in precedenza, relativa alle chiamate dei pazienti, di lasciare i numeri sul visore fino a quando non debbano sparire per far posto a nuove chiamate (vedi secondo appello) si è rivelata fonte di confusione. Si vuole quindi modificare ConsultazionePLG da questo punto di vista, introducendo esplicitamente il caso d'uso ChiamataPaziente, che ha inizio dopo che il Medico ha selezionato un appuntamento nel proprio PianoDiLavoroGiornaliero, e deve garantire che i visori vengano aggiornati correttamente: il numero di prenotazione relativo al Paziente selezionato deve restare visibile finchè non viene chiamato un altro Paziente, ma non oltre.

Domanda 3. Requisiti.

- a) Dare la narrativa di ChiamataPaziente.
- b) E' opportuno introdurre una relazione tra ConsultazionePLG e ChiamataPaziente?

Risposta. a)

Nome: ChiamataPaziente.

Breve descrizione: Il Medico affigge il numero di prenotazione del prossimo Paziente da visitare.

Attore principale: Medico.

Attori secondari: Nessuno.

Precondizioni: Deve essere selezionato un appuntamento nel PianoDiLavoroGiornaliero del Medico.

Postcondizioni: Numero affisso.

Sequenza principale degli eventi:

1. Il Medico richiede l'affissione del numero di prenotazione relativo all'appuntamento selezionato.
2. Il sistema controlla se c'è un numero affisso per l'ambulatorio usato dal Medico;
3. se (sì)
 - 3.1. il sistema lo elimina
4. il sistema affigge il numero dell'appuntamento selezionato.

Situazioni eccezionali: NA.

b) E' naturale porre un <<include>> dal primo al secondo.

Si considerino le seguenti componenti del sottosistema di supporto alle attività ambulatoriali:

Componente	Responsabilità
GestionePLG	Permette al medico di gestire il Piano di Lavoro Giornaliero, attivare la chiamata dei pazienti e aprire la cartella clinica.
GestioneVisore	Gestisce le richieste di chiamata originate da più medici, e le visualizza.

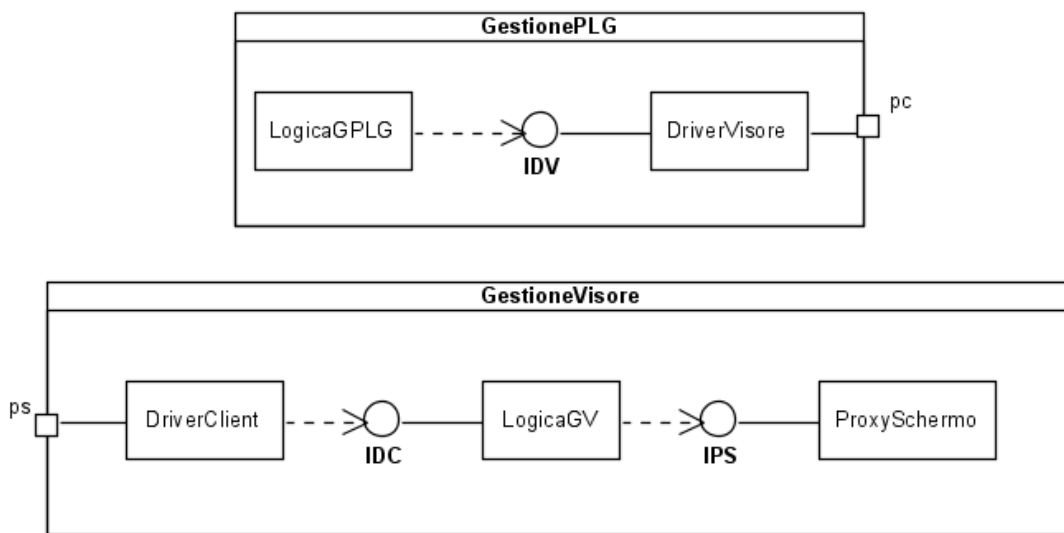
che nella vista C&C son connesse come client-server (il visore è visto come server).

Domanda 4. Progettazione di dettaglio.

Chiamando *pc* e *ps* rispettivamente i porti lato cliente e servente, dare due diagrammi di struttura composita per le due componenti.

NB Il diagramma per GestionePLG può essere incompleto, limitandosi a coprire la sola chiamata dei pazienti.

Risposta.



Si considerino le seguenti classi, che mostrano parte della realizzazione del calcolo del ticket di una visita ambulatoriale:

```

public interface IDriverDB {
    Prenotazione getPrenot(Integer numeroPrenot)
        throws PrenotazioneInesistenteEx;
    //...
}
public class Ticket {
    protected Euro importo = Euro.zero;
    //...
}
public class TicketVisitaAmbulatoriale extends Ticket {

// parte di configurazione
private static IDriverDB driverDB = null;

```

```

private static IRegoleTicket regole = null;
public static void inizializza(IDriverDB dDB, IRegoleTicket r){
    driverDB = dDB;
    regole = r;
}
// parte operativa
public TicketVisitaAmbulatoriale(Integer numPrenot,
                                   CodiceFiscale cf ) {
    Prenotazione prenot = driverDB.getPrenot(numPrenot); //1
    boolean concordanzaCF = prenot.haComeCF(cf); //2
    if (!concordanzaCF) { //3
        throw new CFdiscordantiEx(prenot, cf); //4
    }
    importo = regole.determinaImporto(prenot); //5
}
// ...
}

```

Domanda 5. (Verifica)

- Dare un insieme di casi di prova che copra `TicketVisitaAmbulatoriale` secondo il criterio dei comandi
- Dire se è necessario estenderlo per soddisfare i criteri delle decisioni e delle condizioni, motivando la risposta.

NB Si usino valori simbolici, np1, np2, cf1... e si usi l'ambiente per registrare le condizioni sullo stato del sistema durante il test.

Risposta.

a)

Input	Output	Ambiente
np1, cf1	il (valore determinato dalle regole, secondo l'oracolo)	Detta p la prenotazione restituita da <code>getPrenot(np1)</code> , il valore booleano restituito da <code>p.haComeCF(cf1)</code> è true.
np2, cf2	eccezione	Detta p la prenotazione restituita da <code>getPrenot(np2)</code> , il valore booleano restituito da <code>p.haComeCF(cf2)</code> è false.

b) L'insieme dato soddisfa anche il criterio delle decisioni, dato che la presenza di `throw` impone di seguire entrambe le vie del condizionale, e quello delle condizioni, dato che l'espressione condizionale non ha connettivi. Infatti, nel caso venga lanciata l'eccezione, il comando alla linea 5 non viene eseguito.