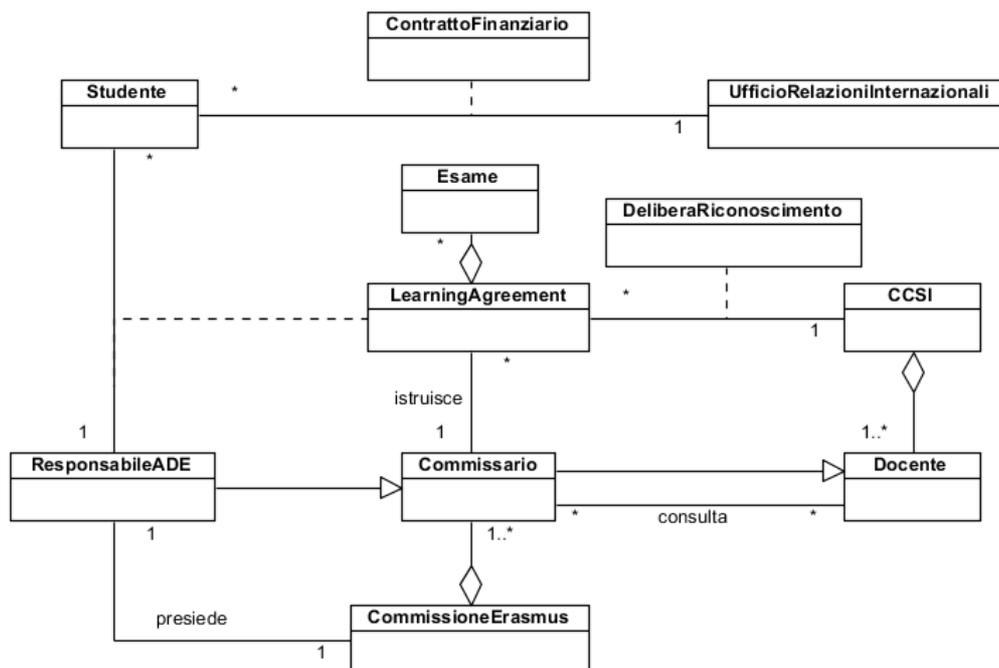


La prova si svolge a libri chiusi (non è permessa la consultazione di materiale didattico).

Si consideri il caso di studio Gestione Erasmus e in particolare la stipula dei contratti che riguardano gli studenti in uscita.

Il seguente diagramma delle classi modella la parte del dominio relativa all'istruzione e alla stipula dei contratti che riguardano gli studenti in uscita:



Nel rispondere alla prossima domanda si ricordi che l'istanza di una classe associazione è un oggetto collegato a due oggetti, istanze delle classi agli estremi dell'associazione.

**Domanda 1.** Analisi del dominio.

Dare un diagramma degli oggetti che comprenda, gli Studenti s1 e s2, i Docenti d1 e d2 (d1 è anche commissario e ResponsabileADE), gli Esami e1, e2e almeno un oggetto per le altre classi che compaiono nel diagramma dato.

\*\*\*\*\*

I membri della commissione Erasmus valutano la corrispondenza tra i corsi offerti all'estero e quelli di Pisa in base ai dati loro disponibili. Se necessario, per risolvere questioni che possono sorgere nel valutare un'equivalenza, consultano il docente del corso pisano. In tal caso, le interazioni tra il commissario e il docente sono mediate dal sistema GestioneErasmus, che le gestisce appoggiandosi a un sistema esterno di posta: i docenti consultati non accedono direttamente al sistema GestioneErasmus, ma comunicano con esso via posta elettronica. Sono stati individuati tre casi d'uso:

**Prima Valutazione:** Il commissario decide sulla corrispondenza, o, se necessario, individua un docente a cui chiede una consulenza.

**Precondizione:** Esiste una coppia di esami per cui deve essere valutata la corrispondenza.

**Postcondizione:** La coppia è dichiarata equivalente o no, oppure è stata richiesta una consulenza in merito.

**AcquisizioneParere:** La mattina il sistema acquisisce i pareri inviati dai docenti, in risposta alle richieste ricevute.

**Precondizione:** E' stata richiesta una consulenza.

**Postcondizione:** La risposta alla richiesta di consulenza è stata registrata.

**FinalizzazioneConsulenza:** Il commissario decide sulla base della consulenza ricevuta.

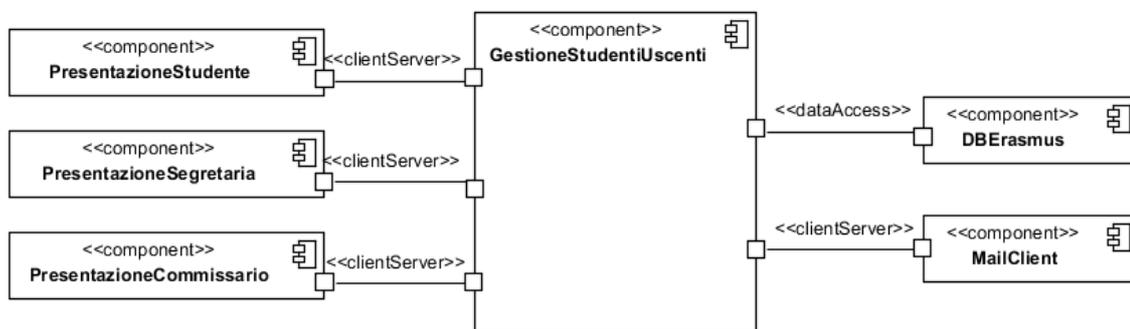
**Precondizione:** La risposta alla richiesta di consulenza è stata registrata.

**Postcondizione:** La coppia di esami è dichiarata equivalente o no.

**Domanda 2.** Analisi dei requisiti: Completare la narrativa dei casi d'uso AcquisizioneParere e FinalizzazioneConsulenza considerando almeno una sequenza alternativa per ciascuno.

\*\*\*\*\*

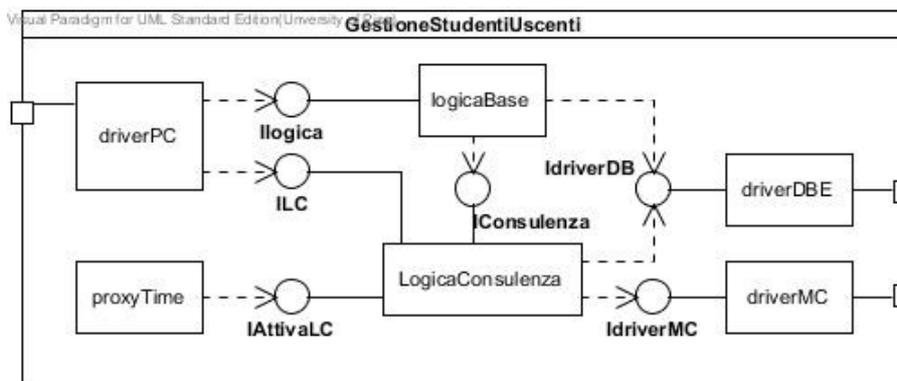
Si è deciso di realizzare il sistema su tre livelli logici: Presentazione, Business Logic (GestioneStudentiUscenti) e un terzo livello che comprende il data-base (DBErasmus) e un cliente di posta, come documentato dalla seguente vista C&C, che comprende le componenti di interesse per i caso d'uso legati alla valutazione di equivalenza tra corsi.



**Domanda 3.** Realizzazione dei casi d'uso. Tenendo conto componenti introdotte qui sopra, dare un diagramma di sequenza che mostri lo scambio di messaggi attori-componenti e componenti-componenti necessari per realizzare il caso d'uso AcquisizioneParere.

\*\*\*\*\*

Si è deciso di seguire un modello di ciclo di vita incrementale, realizzando inizialmente solo parte del caso d'uso PrimaValutazione, senza prevedere cioè la possibilità di richiedere una consulenza. Ciò ha portato al seguente progetto di dettaglio:



Nel primo incremento si realizza solo logicaBase e le parti rilevanti dei due driver che essa utilizza, e LogicaConsulenza è uno stub che genera un messaggio d'errore. Nel secondo incremento si completa la componente, realizzando prima il caso d'uso AcquisizioneParere e poi FinalizzazioneConsulenza.

**Domanda 4.** Progetto di dettaglio: Dare un diagramma delle classi che definisca, in prima approssimazione, le interfacce (e le classi che esse utilizzano) necessarie per realizzare il caso d'uso AcquisizioneParere.

\*\*\*\*\*

Per facilitare l'individuazione delle corrispondenze tra gli esami di Pisa e quelli delle sedi estere è in corso di allestimento un DB che registra le corrispondenze già approvate. Nel frattempo si vuol testare il codice che lo utilizzerà. In particolare, la classe Verifica implementa l'interfaccia IVerifica che offre il metodo `verificaCorrispondenze`, che, data la lista degli esami che uno studente intende sostenere all'estero, restituisce due liste: la prima raccoglie quelli per cui esiste già una corrispondenza, la seconda quelli da valutare ex-novo.

Il metodo `verificaCorrispondenze`, per determinare il suo risultato, invoca il metodo `check`, offerto dall'interfaccia IDriverDB, che dice se un esame di Pisa ha una corrispondenza all'estero (per semplicità assumiamo che la richiesta avvenga in uno stato in cui l'università estera è nota).

Per fare le prove, è stato allestito l'ambiente di test del metodo `verificaCorrispondenze` descritto dal seguente diagramma di struttura composita:



Lo stub che implementa l'interfaccia IDriverDB simula un database in cui compaiono gli esami di codice A1, A2. Gli altri esami hanno codici simili: A3 ecc.

Assumendo che ListaEsami sia un tipo per cui è definito un iteratore, e date le definizioni:

per la classe Esito

```

/* aggiunge es alla lista degli esami
 * con corrispondenza, in this
 */
void addCorrispondente(Esame es){
    ...}
/* aggiunge es alla lista degli esami
 * senza corrispondenza, in this
 */
void addNonCorrispondente(Esame es){
    ...}

```

per Iterator su ListaEsami

```

/* dice se ci sono ancora esami
 * da considerare
 */
public boolean hasNext(){
    ...}
/* restituisce un esame e aggiorna
 * di conseguenza lo stato dell'
 * iteratore
 */
public Esame next(){
    ...}

```

il metodo `verificaCorrispondenze` è stato realizzato come segue, dove `myDB` è istanziato allo stub descritto sopra:

```

public Esito verificaCorrispondenze(ListaEsami l) {
    Iterator i = l.iterator();
    Esito o = new Esito();
    while (i.hasNext()) {
        Esame e = i.next();
        if (myDB.check(e)) {
            o.addCorrispondente(e);
        } else {
            o.addNonCorrispondente(e);
        }
    }
    return o;
};
}

```

**Domanda 5.** Verifica: dare l'insieme minimo di casi di test a scatola aperta per `verificaCorrispondenze` che soddisfi il criterio di copertura delle decisioni.

Si rappresentino le liste secondo lo schema `[]`, `[A1]`, `[A1, A2]`, e le coppie come `<U,D>` e si giustifichi la risposta.