

1) Una società finanziaria eroga prestiti ai suoi clienti (privati o aziende). Per ogni cliente, oltre ai dati anagrafici, si memorizza la sua fascia di affidabilità, intesa come la probabilità che il prestito venga correttamente restituito. Ci sono tre fasce di affidabilità (bassa, media, alta), stimabili tramite un algoritmo a partire dai dati anagrafici e dallo stipendio del cliente. Un prestito è approvato da un agente, che opera presso una delle sedi della società, a un cliente; ha una data di richiesta, una di approvazione e una di erogazione. All'atto dell'approvazione si concorda un piano di rate per la restituzione; i piani variano per numero di rate, importo delle rate e loro periodicità. Il tasso applicato sulle rate cambia a seconda che il cliente sia un privato o un'azienda. In prossimità della scadenza di una rata, viene inviato al cliente un promemoria. Se il pagamento della rata perviene entro la scadenza, esso viene registrato. In caso contrario, si invia un sollecito al cliente e si ricalcola la sua fascia di affidabilità. Se il pagamento non perviene entro un mese dalla scadenza della rata, il cliente viene etichettato come insolvente e non gli si concedono altri prestiti.

Si modellino le specifiche sopra riportate in UML attraverso un diagramma dei casi d'uso (6/32 punti), un diagramma delle classi (9/32 punti) e un diagramma di attività per la riscossione di una rata (8/32 punti).

- 2) Si elenchino e si illustrino brevemente le fasi del ciclo di vita del software successive all'installazione (4/32 punti).
- 3) Si definisca il numero ciclomatico del grafo di controllo di una procedura e si spieghi il suo ruolo nell'ambito della misurazione del software (5/32 punti).

1) Una catena internazionale di hotel gestisce le prenotazioni in modo informatico. Collegandosi al sito della catena, un utente può scegliere uno degli hotel della catena e, indicando un periodo di soggiorno e il numero di occupanti, ottenere la lista delle possibili sistemazioni (tipo di camera e prezzo). Tutti gli hotel hanno gli stessi tipi di camere (anche se in numero diverso); il prezzo di una camera dipende, oltre che dal suo tipo, anche dal mese del soggiorno e dall'hotel. L'utente può scegliere un tipo di camera ed effettuare la prenotazione. Ciascun hotel tiene traccia di tutte le prenotazioni effettuate; inizialmente la prenotazione è in stato "provvisoria", e diventa "confermata" nel momento in cui l'utente invia i dati della sua carta di credito, previo controllo di validità della stessa effettuato da un operatore collegandosi a una banca dati esterna. Attraverso il sito, l'utente può anche cancellare una prenotazione, nel qual caso gli viene applicata una penale (a valere sulla carta di credito). Non interessa tenere traccia delle prenotazioni cancellate.

Si modellino le specifiche sopra riportate in UML attraverso un diagramma dei casi d'uso (6/32 punti), un diagramma delle classi (10/32 punti) e un diagramma degli stati per la classe *Prenotazione* (7/32 punti).

2) Si illustri il concetto di *polimorfismo* nel paradigma a oggetti, spiegandone pro ed eventuali contro (4/32 punti).

3) Si illustrino i principi fondanti dei *modelli di processo agili*, e si descrivano in particolare le attività strutturali dell'*extreme programming* (5/32 punti).

1) Le Terme Quisisana erogano diversi tipi di terapie, tra cui aerosol sulfurei, bagni di fanghi, inalazioni a getto diretto, ecc. Ogni terapia fa capo a una specialità (dermatologia, otorinolaringoiatria, fisiatra, medicina estetica). Per accedere alle terapie un paziente deve innanzitutto prenotare telefonicamente una visita specialistica presso uno degli ambulatori. A seguito della visita, il medico potrà prescrivere un ciclo di terapie (per esempio, 12 aerosol sulfurei + 12 inalazioni a getto diretto). Il paziente si recherà poi presso uno degli sportelli per pagare la visita e il ciclo di terapie, ricevendo una card magnetica che gli darà accesso alle aree in cui si praticano le terapie. In ogni area si pratica una specifica terapia. A ogni accesso, il paziente striscia la sua card su un lettore che verifica la corrispondenza tra la prescrizione e l'area cui si accede, e decrementa la disponibilità residua di accessi. Per le terapie non serve prenotazione, però il ciclo completo deve essere fruito dal paziente entro 60 giorni dal primo accesso. Tutti i dati vengono gestiti su supporto informatico, a partire dalla registrazione della prenotazione telefonica della visita. In ogni momento, l'amministratore di sistema può accedere ai dati delle prenotazioni e delle terapie erogate, nonché aggiungere o cancellare tipi di terapie.

Si modellino le specifiche sopra riportate in UML attraverso un diagramma dei casi d'uso (6/32 punti), un diagramma delle classi (10/32 punti) e un diagramma degli stati per la classe Card (7/32 punti).

2) Si illustri il concetto di *delegazione* nel paradigma a oggetti, spiegando in che modo esso trova applicazione durante la progettazione delle associazioni (5/32 punti).

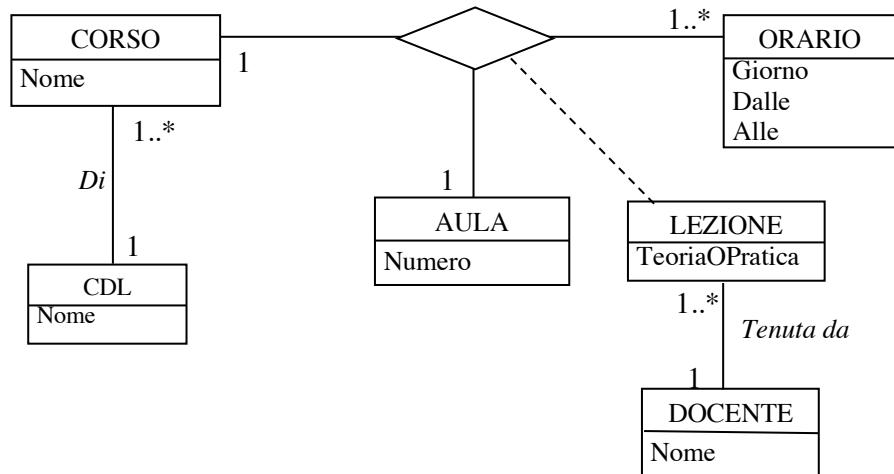
3) Si elenchino e si discutano brevemente 3 *qualità del software*, di cui almeno una esterna e una interna (4/32 punti).

1) Il sistema *AlphaCentauri* ha lo scopo di rilevare infrazioni al codice stradale, controllando semafori e strade ad accesso limitato mediante telecamere. Di ogni telecamera, identificata da un codice univoco, sono noti il modello, l'ubicazione (ovvero l'indirizzo e il posizionamento specifico, ad es. l'altezza da terra), la strada controllata, e tutte le date in cui la telecamera è stata revisionata. Tutti i nomi delle strade fanno riferimento allo stradario ufficiale del Comune. Per le telecamere dedicate al controllo degli accessi vengono in particolare specificati, per ogni giorno della settimana, i periodi (uno o più) di inizio e fine del divieto di accesso, oltre a un elenco di targhe esentate dal divieto. Ogni infrazione rilevata viene registrata memorizzando la data e l'ora di rilevamento, la targa del veicolo interessato e uno o più file di documentazione (video o foto).

Si modellino le specifiche sopra riportate in UML attraverso un diagramma delle classi (10/32 punti).

2) Dato il diagramma delle classi in figura, relativo all'orario delle lezioni per i corsi di laurea di una scuola, si effettui il progetto delle associazioni (10/32 punti) considerando un carico di lavoro così composto:

- Stampa del numero di ore di teoria e pratica svolte da un certo docente all'interno di ciascun corso in cui insegna (1 volta al giorno)
- Stampa del piano dei corsi per ogni CdL della scuola, calcolando per ciascuno il numero di ore settimanali di teoria e pratica (1 volta al mese)
- Visualizzazione dell'orario per un certo giorno e una certa aula (10 volte al giorno)



3) Si elenchino le fasi del ciclo di vita del software, illustrando in particolare il ruolo dello *studio di fattibilità* (6/32 punti).

4) Si descriva brevemente il metodo *function points* spiegandone anche il ruolo e l'utilità (6/32 punti).

1) Il sito *ConferenceManager* (*ConfMan*) permette di gestire online tutte le attività principali relative all'organizzazione scientifica di conferenze. Di ogni articolo scientifico sottomesso a una data conferenza, *ConfMan* tiene traccia degli autori (uno o più), del titolo, di un suo riassunto (*abstract*) e del file in formato PDF contenente il testo dell'articolo. Al termine del periodo in cui è possibile sottomettere articoli ha luogo la fase di *bidding*, durante la quale ciascun membro del Comitato Scientifico della conferenza può vedere titoli e abstract degli articoli sottomessi ed eventualmente indicare per ciascuno se (a) desidera revisionare quell'articolo, (b) non desidera revisionare quell'articolo, (c) non può revisionare quell'articolo a causa di un conflitto di interessi. Scaduto il termine della fase di *bidding*, i responsabili scientifici della conferenza assegnano ogni articolo a tre revisori, selezionati tra i membri del Comitato Scientifico tenendo conto delle preferenze espresse da ciascuno durante il *bidding*. Ogni revisore, nel periodo di tempo stabilito per quella conferenza, fornisce un valutazione dell'articolo costituita da un punteggio e da una revisione testuale. Al termine della fase di valutazione, il Comitato Scientifico decide quali articoli devono essere accettati e quali no, e per ogni articolo viene inviata una mail agli autori con le revisioni e il punteggio. Per i soli articoli accettati è richiesta la sottomissione, entro una data prefissata, di una versione finale che tenga conto dei commenti dei revisori. Oltre alle informazioni già citate, *ConfMan* mantiene i dati sul nome di ciascuna conferenza e sulla sua edizione (ad es. *International Conference on UML, Sixth edition*), sul periodo e sul luogo di svolgimento.

Si modellino le specifiche sopra riportate in UML attraverso un diagramma delle classi (8/32 punti), un diagramma dei casi d'uso (5/32 punti) e un diagramma degli stati che rappresenti a livello macro i diversi stati di *ConfMan*, corrispondenti alle diverse fasi del processo illustrato (7/32 punti).

2) Si descrivano brevemente almeno due *meccanismi di estendibilità* di UML (6/32 punti).

3) Si descrivano le due tecniche di *prototipazione* e se ne illustrino obiettivi e vantaggi (6/32 punti).

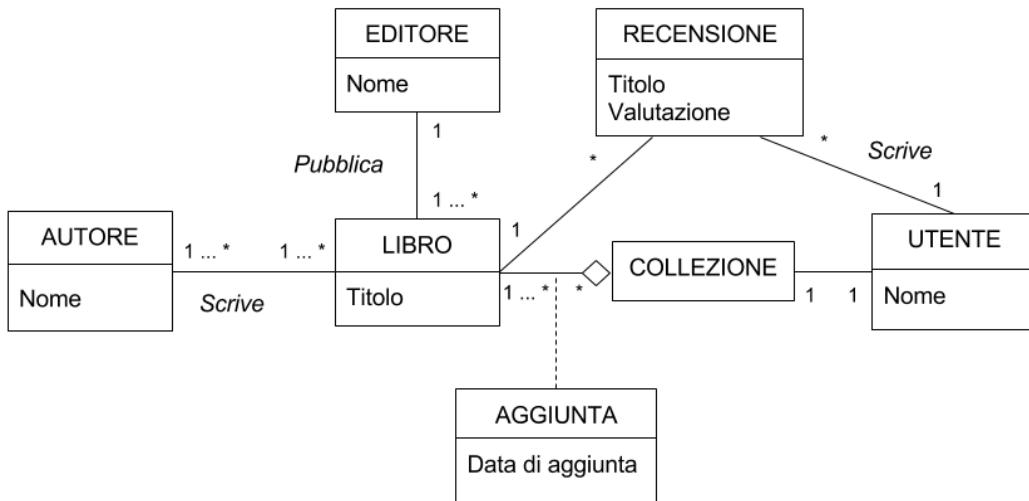
1) Una società di animazione organizza in villaggi turistici il programma settimanale di intrattenimento. Ogni serata del programma settimanale deve essere di tipo diverso (es: cabaret, ballo, etc) per evitare che i turisti assistano allo stesso tipo di spettacolo nell'arco della settimana. Le serate hanno un orario di inizio, uno di fine e un luogo del villaggio dove vengono tenute. I clienti del villaggio turistico possono segnalare la loro partecipazione per ricevere un promemoria inviato tramite SMS trenta minuti prima dell'inizio della serata. Un team di animatori viene assegnato a un villaggio per l'intera settimana di animazione, ogni singola serata potrebbe però essere organizzata solo da una parte di questo team. I turisti che partecipano alle serate possono dare un parere sulla serata e/o su ogni singolo animatore che ne ha fatto parte.

Modellare anche la carriera dell'animatore. Ogni candidato animatore riceve inizialmente una proposta da parte di un agente della società, se questa viene accettata entrerà a far parte della scuderia di tale agente. Solo a questo punto potrà ricevere richieste di partecipazione per l'animazione in villaggi turistici, con indicazione di luogo, data e durata. Una volta terminato il periodo in villaggio dovrà attendere la conferma in ruolo da parte dell'agente, in caso negativo uscirà dalla scuderia. L'animatore potrà anche rifiutare una chiamata in un villaggio mandando una motivazione che l'agente giudicherà per decidere se mantenere i rapporti con l'animatore o escluderlo dalla propria scuderia.

Si modellino le specifiche sopra riportate in UML attraverso un diagramma delle classi (12/32 punti).

2) Dato il diagramma delle classi in figura, relativo ad un sito di gestione di collezioni di libri, si effettui il progetto delle associazioni (8/32 punti) considerando un carico di lavoro così composto:

- Visualizzazione dell'elenco degli editori i cui libri hanno una media di valutazioni superiore ad una data soglia (50 volte al mese).
- Visualizzazione degli utenti che hanno scritto recensioni su almeno uno dei libri di un dato autore (1 volta a settimana).
- Visualizzazione, per ogni autore, del libro che è presente in più collezioni (100 volte al giorno).



3) Si descrivano brevemente le caratteristiche dei diversi tipi di manutenzione del software (6/32 punti).

4) Si illustrino i principi fondanti del *model-driven development* spiegando in particolare il ruolo svolto dai diversi modelli (6/32 punti).

1) Gli organizzatori della gara cicloturistica Nove Colli vogliono creare un software informativo per la gestione della corsa. Per ogni edizione della Nove Colli si vogliono mantenere la data di svolgimento, la data di apertura e chiusura delle iscrizioni, una descrizione dei percorsi (uno di 130 km e uno di 200 km) e il numero totale dei partecipanti. La descrizione dei percorsi è costituita dall'elenco dei punti di controllo e relative distanze (es. Cesena 20 km, Mercato Saraceno 32 km, Colle Barbotto 15km, ...). I corridori al momento dell'iscrizione forniscono le proprie generalità (nome, cognome, ...), il gruppo ciclistico al quale sono affiliati, la categoria di appartenenza (es. cicloamatore, elite, senior) e il tipo di percorso prescelto (130 km o 200 km). Inoltre ogni corridore deve presentare un certificato medico agonistico valido alla data della corsa. Le generalità dei corridori vengono controllate dagli organizzatori mentre il certificato medico viene verificato dal medico sportivo. In caso di non conformità la richiesta di iscrizione viene rifiutata; se invece i controlli hanno esito positivo il corridore deve effettuare il versamento della quota di partecipazione alla gara il cui corretto versamento viene verificato alla scadenza delle iscrizioni. Se anche il versamento viene effettuato correttamente l'organizzazione registra il corridore e gli assegna un numero di maglia (univoco per ogni edizione). La gestione della corsa prevede la registrazione dei tempi in tutti i punti di controllo e della posizione di classifica assoluta e di categoria di tutti gli iscritti al fine di poter stampare le classifiche singole e per gruppi sportivi di ciascuna gara e di ciascuna categoria. Viene inoltre stilata una particolare classifica per il migliore scalatore realizzata in base al tempo di scalata del "Barbotto".

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma delle classi (*10/32 punti*)
- 1.2) un diagramma dei casi d'uso (*3/32 punti*)
- 1.3) un diagramma delle attività per il caso d'uso "Iscrizione" (*7/32 punti*)

2) Si spieghi cos'è il metodo *CoCoMo* e come può essere utilizzato (*5/32 punti*).

3) Si illustri il concetto di *incapsulamento* nel paradigma a oggetti, spiegandone pro ed eventuali contro (*7/32 punti*).

1) Dato il seguente dominio applicativo:

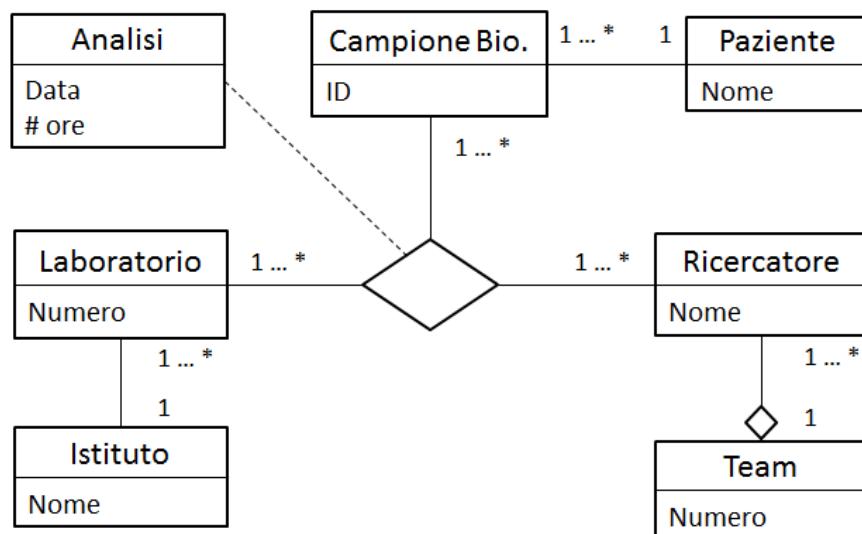
Si vuole realizzare un sistema software per la gestione di una galleria d'arte. La galleria è costituita da una serie di sale per ciascuna delle quali si memorizzano un numero univoco, il nome e la dimensione. La galleria organizza delle mostre delle quali si memorizzano il titolo, la data di inizio, la data di fine e il costo del biglietto. In ogni mostra vengono esposti diversi quadri e uno stesso quadro può comparire in più mostre. Per ogni quadro si memorizzano l'autore, l'anno di realizzazione, una descrizione e la sala nella quale viene esposto (che può cambiare in mostre diverse). Per ogni mostra si memorizza inoltre un itinerario consigliato, contraddistinto da un codice e costituito da una lista ordinata di sale da visitare. Per ogni mostra la galleria organizza delle visite guidate delle quali si memorizzano l'orario di inizio e la durata. Le visite guidate si ripetono nei vari giorni della mostra (una stessa visita può ripetersi in giorni diversi) e per ogni visita effettuata si vuole memorizzare il numero di partecipanti. Il software deve permettere la ricerca di informazioni su quadri e mostre, la visualizzazione della classifica delle mostre per numero di visitatori e il calcolo dell'incasso totale di ciascuna mostra.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma delle classi (*14/32 punti*)
- 1.2) un diagramma dei casi d'uso (*3/32 punti*)

2) Dato il diagramma delle classi in figura, relativo a laboratori di analisi genetica, si effettui il progetto delle associazioni (*8/32 punti*) considerando un carico di lavoro così composto:

- 1) Per ogni team, mostrare il laboratorio più utilizzato (1 volta al mese).
- 2) Elenco delle analisi effettuate su un dato paziente (10 volte al giorno).
- 3) Per ogni ricercatore, totale ore di utilizzo di un certo laboratorio (1 volta al giorno).



3) Si illustrino le diverse tipologie di *tecniche di verifica del software dinamiche* (*7/32 punti*).

1) Dato il seguente dominio applicativo:

Una casa d'aste memorizza informazioni relative alle opere d'arte e agli oggetti di antiquariato in vendita.

Ogni oggetto d'antiquariato ha un nome, una datazione, una descrizione, un prezzo di base e una foto. Ogni opera d'arte ha un nome, un autore, una datazione, una descrizione, una tipologia (pittura, scultura, ...), un prezzo base e una foto. Le aste sono relative ad un singolo oggetto o a un gruppo di oggetti, hanno un prezzo base, un prezzo di riserva, un minimo rilancio, una data di inizio e una di fine asta.

Ogni asta inizia con la messa in mostra, da parte dello staff, dell'opera in vendita. In seguito il banditore presenta l'opera e, al contempo, gli offerenti ne eseguono una prima valutazione. Dopo che il banditore ha dichiarato il prezzo base, inizia il rilancio delle offerte da parte degli offerenti che vengono man mano registrate dal banditore. Allo scadere del tempo d'asta massimo, se almeno un'offerta è stata presentata, viene assegnato il vincitore. Successivamente lo staff posiziona, se presente, il prossimo articolo da mettere all'asta (che ha preparato durante lo svolgimento della fase di rilancio).

Per ogni asta vengono memorizzati tutti i rilanci degli utenti registrati e, se l'asta termina con una vendita, si registra il prezzo, la tipologia di pagamento, la data di ricezione del pagamento, i dati dell'acquirente e i dati relativi alla spedizione (indirizzo e risultato). Allo scadere del tempo prestabilito l'asta viene chiusa e il migliore offerente ha tre giorni per effettuare il pagamento che garantisce l'acquisto; in caso contrario l'oggetto può essere rimesso all'asta. Ogni cliente deve registrarsi (login, password, nome, cognome e e-mail) per effettuare le offerte, ma i suoi dati fiscali (luogo e data di nascita, indirizzo e codice fiscale) vengono registrati solo in conseguenza di un'effettiva vendita.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*7/32 punti*)
- 1.2) un diagramma delle classi (*10/32 punti*)
- 1.3) un diagramma di attività relativo allo svolgimento di un'asta (*3° paragrafo*) (*8/32 punti*)

2) Si illustrino le diverse tipologie di *tecniche di verifica del software statiche* (*7/32 punti*).

1) E' dato il seguente dominio applicativo:

Si vuole realizzare un sistema informativo per la gestione di uno studio medico dentistico. L'obiettivo è quello di mantenere traccia dei pazienti in cura presso lo studio, degli appuntamenti e degli interventi effettuati.

Il sistema deve gestire gli appuntamenti con i pazienti; in particolare devono essere memorizzati il giorno e l'ora dell'appuntamento, gli interventi effettuati e il loro costo totale (dato dalla somma dei singoli interventi).

Gli interventi possono essere di diverso tipo (controllo, otturazione, estrazione, inserimento di capsule, pulizia, ecc...). Di ciascun intervento si memorizzano la durata, il medico responsabile e eventualmente gli assistenti che vi hanno preso parte. Di ciascun paziente, medico e assistente si mantengono i dati anagrafici. Il costo di ogni intervento è riportato in un tariffario che contiene i tipi di intervento possibili e le relative tariffe.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Si modelli tramite diagramma degli stati il seguente scenario relativo ad un sistema antintrusione. Inizialmente il sistema è inattivo, per poterlo attivare è necessario inserire correttamente il codice composto da quattro cifre. Se il codice inserito è errato, è possibile digitarlo nuovamente, con un limite massimo di tre tentativi. Se per tre volte di seguito viene inserito un codice errato, il sistema va in stato di blocco e vi rimane fino a quando non viene sbloccato manualmente da tecnici. Se invece il codice viene inserito correttamente, il sistema attende 1 minuto prima di attivarsi. In caso di rilevamento di intrusione il sistema andrà in allarme. Per disattivare il sistema è necessario inserire un apposito codice [in questo caso non è necessario modellare il processo di inserimento del codice dettagliatamente]. (*8/32 punti*)

2) Si illustri il concetto di *ereditarietà* nel paradigma a oggetti, spiegandone vantaggi ed eventuali svantaggi (*5/32 punti*).

3) Si elenchino e si illustrino brevemente tutte le fasi del *ciclo di vita del software* (*7/32 punti*).

1) Dato il seguente dominio applicativo:

Si vuole sviluppare un sistema informativo per la gestione delle sagre di paese e delle recensioni su tali sagre compilate dagli utenti web (di cui sono schedati i dati anagrafici). Per ogni sagra, oltre alle informazioni usuali (nome, data inizio, durata), sono mantenuti il nominativo e i dati di un referente del comitato organizzativo, il sito web e l'indirizzo e-mail. Relativamente al comune ospitante si mantengono nome, provincia, cap, istruzioni per raggiungerlo; ciascun comune può ospitare più di una sagra, ma in diversi periodi dell'anno.

Ad ogni sagra è associato un calendario con date, orari, nomi e descrizioni degli eventi. Tra i possibili eventi vi sono i concerti dal vivo per i quali sono memorizzate informazioni relative alle band che si esibiscono (nome del gruppo, genere musicale, target).

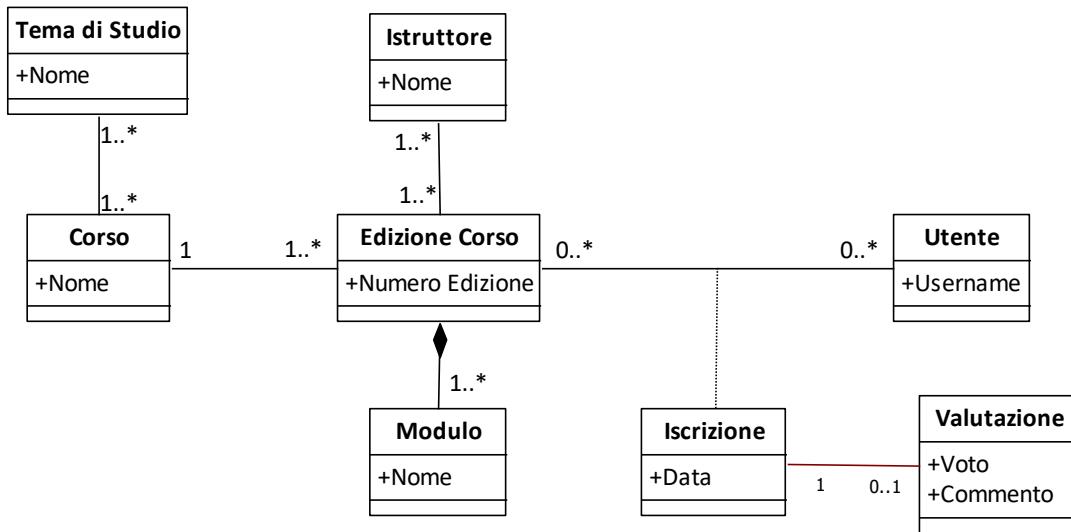
Le recensioni forniscono un commento sintetico sulle sagre e una valutazione in base ad alcuni "parametri valutativi" (Musica, Animazione, Servizio, RapportoQualità/Prezzo, ecc): a ogni parametro è associata una scala di valori numerici. Sono memorizzate sia i voti e il commento dei singoli recensori che i punteggi finali (ottenuti dalla media dei voti dei diversi recensori) e una valutazione globale di ciascuna sagra.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Dato il diagramma delle classi in figura, relativo ad un sistema di gestione di corsi di studio online, si effettui il progetto delle associazioni (*8/32 punti*) considerando un carico di lavoro così composto:

- 1) Per ogni corso, contare il numero di iscritti ad ogni edizione (1 volta al giorno).
- 2) Calcolare la media delle valutazioni per tutti i corsi tenuti da un dato istruttore (10 volte al mese).
- 3) Elenicare gli istruttori che hanno tenuto almeno un'edizione di almeno un corso con un dato tema di studio (1 volta all'anno).



3) Si illustri in termini generali il concetto di *dipendenza* in UML e si diano almeno tre esempi di stereotipi di dipendenza (*6/32 punti*).

4) Si illustrino i principi fondamentali del *test-driven development* (*6/32 punti*).

1) Dato il seguente dominio applicativo:

Il premio Strega è un prestigioso premio letterario che viene assegnato ogni anno a un libro di narrativa in prosa di autore italiano, pubblicato tra il 1 maggio dell'anno precedente e il 30 aprile dell'anno in corso. Il premio viene assegnato da un giuria di oltre 400 persone dette "Amici della Domenica".

Per ciascuna edizione si memorizzano l'anno, l'eventuale premio in denaro previsto e il vincitore. Ogni anno partecipano alla gara diversi libri dei quali si registrano il titolo, gli autori (con relativi dati anagrafici e una breve biografia), la data di pubblicazione, la casa editrice (ragioneSociale e sedeLegale) e il genere. Ogni libro, per essere qualificato a partecipare alla gara, deve essere presentato da due "Amici della Domenica". Le presentazioni di libri, per le quali si memorizza anche un breve giudizio scritto, devono pervenire entro un termine variabile di anno in anno, ma sempre tra l'1 e il 15 di maggio. Per ogni edizione del premio bisognerà dunque memorizzare anche tale termine. Ogni "amico" può presentare al massimo un libro per ogni edizione.

L'assegnazione del premio avviene tramite due votazioni distinte. Nella prima, tutti i libri qualificati sono sottoposti a votazione da parte degli "amici"; i 5 libri che ottengono il maggior numero di voti diventano finalisti e passano alla fase finale. La fase finale, che termina con la consegna del premio al vincitore, consiste in un'ulteriore votazione nella quale gli "amici" scelgono il libro vincente tra i 5 finalisti. Per ciascun libro si vuole mantenere il dettaglio dei voti degli "amici" ottenuti nelle due votazioni. Nel caso in cui nella votazione finale risulti una parità di voti per il 1° posto, il premio è assegnato ex-aequo.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)
- 2) Si modelli il diagramma degli stati relativo alla classe *Libro*, dalla pubblicazione fino all'eventuale esito del concorso (*8/32 punti*).
- 3) Si illustri il concetto di *powertyping* in UML e se ne dia almeno un esempio (*6/32 punti*).
- 4) Si definiscano le seguenti qualità del software: *efficienza, portabilità, interoperabilità* (*6/32 punti*).

1) Dato il seguente dominio applicativo:

Si vuole realizzare un sistema automatico per gestire un'agenzia di taxi.

Ogni taxi ha un numero che lo identifica, una marca, modello, un numero di posti e un conducente (di cui sono mantenuti tutti i dati anagrafici). All'interno di ogni taxi è installato un dispositivo che permette di comunicare al sistema, in qualsiasi momento, la posizione e lo stato (occupato, libero o fuori servizio) del taxi. Per i taxi liberi, la posizione è memorizzata sotto forma di numero di zona (le zone corrispondono ai quartieri della città e determinano le tariffe).

Il sistema deve prevedere la possibilità di prenotare una corsa indicando il giorno, l'ora, l'indirizzo di partenza, in numero di posti e il nome del passeggero richiedente. Se nella prenotazione viene indicato anche l'indirizzo di arrivo il sistema è in grado di fornire al richiedente un preventivo della tariffa (che dipende dalla fascia oraria, dalla zona di partenza e da quella di arrivo). Alla prenotazione sarà poi associato in automatico un taxi libero in zona.

Tutte le corse effettuate devono essere memorizzate nel sistema, con indicazione del taxi, data, orario e indirizzo di partenza, orario e indirizzo di arrivo, Km percorsi, costo della corsa. Il sistema deve prevedere la gestione dei dati contabili di ciascun taxi, con stampe degli incassi totali giornalieri per taxi.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*4/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Si modelli tramite diagramma di attività il seguente scenario riguardante la partecipazione a corsi di formazione e la relativa richiesta di rimborso spese.

Nell'azienda Consulting&Stuff s.r.l., i dipendenti possono partecipare a corsi di formazione (non obbligatori ed organizzati da terzi) per mantenere aggiornate le proprie conoscenze lavorative. Prima di poter partecipare ad un corso, il dipendente interessato deve compilare e inviare al reparto amministrazione l'apposito modulo di autorizzazione e richiesta rimborso (si tratta di un unico modulo). Il reparto amministrazione valuta la richiesta e comunica l'esito della valutazione: autorizzato con rimborso, autorizzato senza rimborso, oppure non autorizzato. Una volta ricevuto l'esito della valutazione, il dipendente, se autorizzato, deve: iscriversi al corso di formazione, eseguire il pagamento e, alle opportune date, partecipare al corso. Dopo il corso, se autorizzato con rimborso, il dipendente deve compilare ed inviare all'amministrazione un documento in cui è elencato il dettaglio spese. Una volta ricevuto il documento di dettaglio, l'amministrazione procede con il versamento del rimborso. (*8/32 punti*)

3) Si spieghi cos'è un'interfaccia in UML e come può essere messa in relazione con le classi che la implementano e quelle che la usano (*6/32 punti*).

4) Si elenchino e si definiscano almeno due qualità del software relative al processo (*5/32 punti*).

1) Dato il seguente dominio applicativo:

La FererVolley vuole realizzare il sistema informativo per la gestione dei campionati italiani assoluti di beach volley maschile che si svolgono con frequenza annuale.

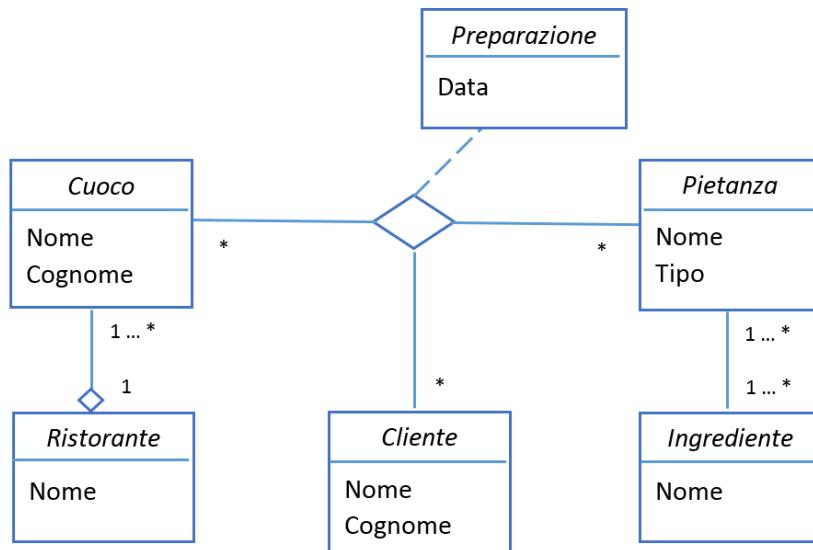
Il campionato è sempre composto da 10 tappe (ciascuna corrispondente ad un singolo torneo) che si svolgono in diverse località italiane. Ciascuna tappa è caratterizzata da una data d'inizio, data di fine, nome della località, e montepremi. A ciascuna tappa si possono iscrivere fino a 48 coppie di giocatori che entrano a far parte del torneo. Di ciascun giocatore sono memorizzati i dati anagrafici, il numero di tesserino FIPAV e il recapito; è inoltre interessante tener traccia del punteggio totale acquisito nelle varie tappe, il montepremi vinto e la posizione in classifica. Ciascun giocatore può cambiare compagno nei diversi tornei, ma non durante le partite di uno stesso torneo. Di tutte le partite di ciascun torneo viene registrato un numero progressivo, il livello (quarti, semifinale, ...), la durata, le coppie sfidanti e il risultato dei set (es. 21-14, 19-21, 15-13). A seguito di ogni torneo ovviamente la posizione di ciascun giocatore può variare in base al punteggio e al montepremi acquisito.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Dato il diagramma delle classi in figura, effettuare il progetto delle associazioni (*8/32 punti*) tenendo conto del seguente carico lavoro:

1. Visualizzare tutte le pietanze preparate da tutti i cuochi di un dato ristorante (1 volta al giorno).
2. Visualizzare l'elenco di tutte le preparazioni effettuate in un dato intervallo temporale (1 volta alla settimana).
3. Visualizzare tutte le pietanze che utilizzano un dato ingrediente (1 volta al mese).



3) Spiegare il doppio significato e utilizzo della relazione di *raffinamento/realizzazione* in UML (*6/32 punti*).

4) Si illustrino i tre criteri principali per il *testing in the small* commentandone l'efficacia (*6/32 punti*).

1) Dato il seguente dominio applicativo:

La lavanderia Arcobaleno vuole realizzare un sistema software per la gestione dei clienti, dei capi da lavare e del tariffario dei servizi offerti.

La lavanderia offre i servizi di lavaggio, in acqua o a secco, e stiratura. Il prezzo dei servizi offerti dipende dalla tipologia di capo da trattare (es. pantalone, gonna, giacca, ...). Per alcune tipologie di capi (es. tappeti, tende) il prezzo dei servizi dipende anche dalla fascia di peso. Le fasce, descritte da un peso minimo e un peso massimo (es. da 3 a 5 kg), devono essere registrate nel sistema.

I clienti della lavanderia sono memorizzati in un archivio; per ciascuno si registrano un codice identificativo, il cognome, il nome, l'indirizzo e un recapito telefonico. Ogni volta che un cliente porta dei capi in lavanderia, nel sistema devono essere memorizzati la data, il costo totale e la data di restituzione prevista.

Inoltre per ogni capo da trattare, identificato da un codice univoco, si memorizzano una descrizione, l'eventuale peso e i servizi richiesti. Naturalmente per uno stesso capo possono essere richiesti più servizi (es. lavaggio a secco e stiratura).

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Nel contesto fornito dall'esercizio 1, modellare tramite diagramma delle attività (*8/32 punti*) il seguente scenario in cui viene descritta la richiesta di servizi da parte di un cliente. Dopo aver ricevuto i capi e i dettagli relativi ai servizi richiesti (lavaggio a secco, stiratura, etc.), l'addetto al front office passa il tutto al personale designato e controlla se i dati relativi al cliente sono già presenti in archivio. In caso negativo, il cliente dovrà fornire i propri dati che in seguito verranno registrati. Nel frattempo, il personale addetto ai servizi si occuperà di eseguire le richieste ricevute. In particolare avvierà il processo di lavaggio e allo scadere del tempo necessario effettuerà la stiratura (notare che il lavaggio o la stiratura potrebbero non essere richiesti). Una volta terminate le precedenti operazioni verrà inviata una notifica al cliente, il quale dovrà poi ritirare i propri capi ed effettuare il dovuto pagamento.

3) Si illustri il concetto di *delegazione* nel paradigma a oggetti, spiegando in che modo esso trova applicazione durante la progettazione delle associazioni (*6/32 punti*).

4) Si spieghi la differenza tra un software *corretto* e un software *robusto* (*6/32 punti*)

1) Dato il seguente dominio applicativo:

Un circolo nautico vuole realizzare un software per la gestione delle anagrafiche dei soci, dei corsi e delle imbarcazioni. Tutti i soci del circolo hanno un tesserino in cui è registrato il codice identificativo, i dati anagrafici (nome, cognome e data di nascita) e la data di annessione al circolo. Ogni anno vengono rinnovate le cariche sociali (presidente, segretario, consigliere, ...) che devono essere registrate (socio in carica, anno). Alcune cariche (es. consigliere) possono essere rivestite da più soci nello stesso anno ma un singolo socio però non può rivestire più cariche contemporaneamente. Ogni socio può possedere una o più imbarcazioni (del resto ciascuna imbarcazione può avere più proprietari, tutti soci, per ciascuno dei quali viene registrata una quota). I dati registrati per ciascuna imbarcazione sono: matricola, marca e modello, nome, valore, data del varo, dimensioni (larghezza e lunghezza), alimentazione e potenza del motore se è una barca a motore. Il circolo organizza inoltre dei corsi aperti a tutti (non solo soci) per i quali viene registrata la data di inizio e fine, il numero massimo di partecipanti, il tipo (es. "corso di vela base"), la quota di partecipazione e dati anagrafici degli iscritti.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Nel contesto fornito dall'esercizio 1, modellare il diagramma degli stati (*8/32 punti*) relativo ai *corsi*. I soci del circolo possono proporre in qualsiasi momento un corso da realizzare in un determinato periodo. La prima domenica di ogni mese i soci si riuniscono per discutere ed eventualmente approvare le proposte di corsi raccolte. In particolare, per poter approvare un corso è necessario il parere positivo della maggioranza dei soci. Se il corso viene approvato, il socio proponente ha il compito di finalizzarne l'organizzazione. In caso si verifichino problemi tali per cui il corso non può più essere svolto (per esempio nessun istruttore risulta disponibile), allora questo verrà accantonato per poter essere eventualmente rivalutato e riorganizzato in un secondo momento. Dopo che il corso è stato approvato e organizzato è necessario raccogliere un numero sufficiente di iscritti prima delle date effettive del corso. In caso non vengano raccolte abbastanza iscrizioni, il corso sarà temporaneamente sospeso per poter essere riproposto (senza rivalutazione) in futuro. Se invece il numero di iscrizioni risulta sufficiente, il corso verrà realizzato nelle date concordate.

3) Si definisca il *numero ciclomatico* del grafo di controllo di una procedura, si enunci il *teorema di Mills* e si porti un esempio di calcolo (*6/32 punti*).

4) Si spieghi cos'è un *package* e si illustrino struttura e utilizzo dei *diagrammi dei package* (*6/32 punti*).

1) Dato il seguente dominio applicativo:

Si vuole realizzare un software per la gestione degli interventi effettuati da un'azienda idraulica. Il sistema dovrà gestire il magazzino, i dipendenti, le prenotazioni degli interventi richiesti dai clienti, il tariffario, l'emissione di ricevute fiscali e i pagamenti.

I clienti, dei quali si memorizzano nominativo, indirizzo, numero di telefono e codice fiscale, richiedono telefonicamente un intervento. Al momento della prenotazione, il cliente deve specificare la descrizione dell'intervento richiesto. Per ogni prenotazione si memorizzano inoltre la data e l'ora della richiesta, il cliente e la priorità (urgenza dell'intervento).

Ogni intervento può essere effettuato da uno o più dipendenti e può prevedere l'impiego di materiale presente in magazzino. Per ciascun dipendente si registrano i consueti dati anagrafici e il livello (es. apprendista, operaio specializzato, ecc.). A ciascun livello corrisponde un costo orario di manodopera. Il costo totale di ogni intervento è calcolato in base ai materiali utilizzati, al costo di manodopera dei dipendenti intervenuti e al numero di ore impiegate. All'intervento può essere aggiunto un sovrapprezzo determinato dal livello di priorità. Il sistema deve gestire il materiale presente in magazzino, memorizzando per ciascun articolo un codice identificativo, il nome, la categoria, la quantità disponibile, il prezzo di vendita e il livello di riordino.

A ogni intervento è associata l'emissione di una ricevuta fiscale contraddistinta da un numero progressivo e dall'anno di emissione. Al momento dell'emissione della ricevuta, l'azienda può decidere di applicare una percentuale di sconto sull'importo totale. Per ogni ricevuta emessa bisogna tenere traccia dell'eventuale pagamento.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Modellare tramite *diagramma di attività* (*8/32 punti*) il seguente scenario in cui viene descritta la sottomissione e la gestione di un reclamo assicurativo relativo ad un sinistro stradale. Per prima cosa il cliente della compagnia assicurativa deve compilare e sottomettere un modulo di richiesta in cui vengono descritti i dettagli del sinistro. In seguito l'ufficio assicurativo prende in carico la richiesta e ne verifica la correttezza. L'esito della verifica viene poi notificato al cliente che, in caso di errori di compilazione dovrà effettuare una nuova sottomissione. Se invece il reclamo risulta formalmente corretto, l'ufficio provvede, da un lato a valutare la copertura assicurativa del cliente, dall'altro a stabilire le colpe del sinistro. Nel frattempo un perito ha il compito di valutare l'entità dei danni. Al termine di queste operazioni l'ufficio effettua una proposta di risarcimento che il cliente valuterà se accettare o meno. In caso di accettazione il reclamo viene marcato come risolto e viene effettuato il versamento come stabilito nella proposta. In caso di rifiuto il cliente deve comunicare le proprie rimostranze all'ufficio assicurativo che poi valuterà se è possibile o meno fare una nuova proposta (ci possono essere più iterazioni di proposta e rimostranze). Nel caso in cui l'ufficio non voglia fare ulteriori proposte il reclamo viene marcato come irrisolto.

3) Si spieghi cos'è un *template* in UML e in che modo si rappresenta (*6/32 punti*).

4) Si illustri il ruolo dello *studio di fattibilità* all'interno del ciclo di vita del software, precisandone obiettivi e risultati (*6/32 punti*).

1) Dato il seguente dominio applicativo:

La redazione della trasmissione “Il grande fratello” vuole realizzare un software per la memorizzazione di tutti i dati e gli eventi relativi alla permanenza dei concorrenti nella casa. Si vogliono mantenere le informazioni personali relative ai concorrenti (nome, cognome, data di nascita, segno zodiacale, città di provenienza, foto, professione, hobby, motivazioni) e quelle relative alla loro permanenza (data di ingresso, eventuale data di uscita). Ogni settimana il gruppo di concorrenti ha un budget che deve spendere per l’acquisto degli alimenti; una percentuale di questo budget può essere scommessa nella prova settimanale. Il risultato di questa prova (rappresentata da nome e descrizione) può quindi aumentare o diminuire il budget. Al termine di ogni prova vengono anche scelti i concorrenti migliori, che riceveranno un premio, e i peggiori, che dovranno scontare una punizione. Infine le nomination: ogni settimana vengono fatte le nomination dove ciascun concorrente nomina due suoi compagni; i concorrenti (2 o più) che hanno avuto il maggior numero di voti sono candidati all’eliminazione. I telespettatori decidono tramite televoto chi eliminare; il software memorizza il numero di voti per ogni nominato.

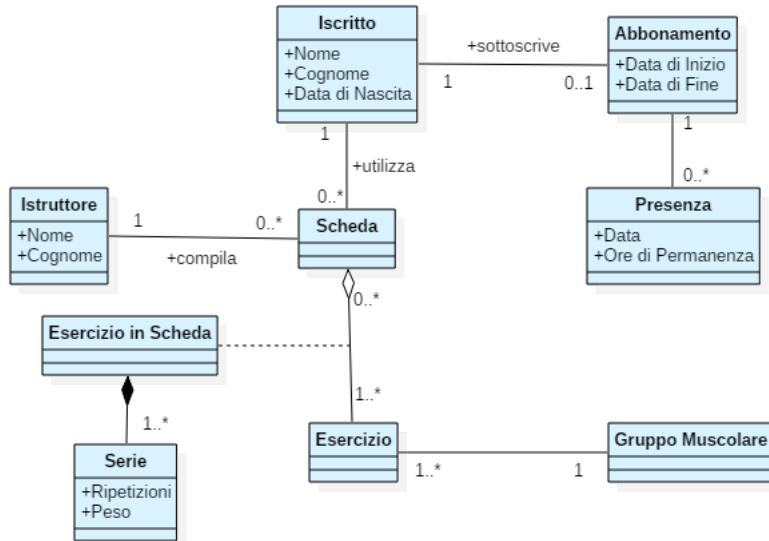
Si modellino le specifiche sopra riportate in UML attraverso:

1.1) un diagramma dei casi d’uso (3/32 punti)

1.2) un diagramma delle classi (9/32 punti)

2) Dato il diagramma delle classi in figura relativo alla gestione di una palestra, svolgere il progetto delle associazioni (8/32 punti) tenendo conto del seguente carico di lavoro:

- Calcolare le ore totali di permanenza per ogni iscritto (una volta alla settimana).
- Dato un iscritto, elencare le schede che utilizza e gli istruttori che le hanno compilate (due volte alla settimana).
- Elencare i cinque esercizi prescritti più di frequente per un dato gruppo muscolare (una volta ogni sei mesi).



3) Si spieghi cos’è uno *stereotipo* in UML e si diano tre esempi di stereotipi predefiniti (6/32 punti).

4) Si spieghi il significato del principio di *anticipazione dei cambiamenti* nell’ingegneria del software (6/32 punti).

1) Dato il seguente dominio applicativo:

Si vuole realizzare il sistema informativo per la gestione del reparto di ostetricia dell'ospedale di Cesena.

Il sistema deve gestire i dati del personale dipendente, delle partorienti e dei neonati. Il personale dipendente si divide in medici, assistenti e ostetriche. Per ciascun dipendente si memorizzano codice fiscale, nome, cognome, indirizzo e uno o più recapiti telefonici. Il sistema deve permettere inoltre di registrare i turni settimanali di ciascun dipendente, memorizzando giorno della settimana, ora di inizio e ora di fine del turno. Oltre ai turni è necessario memorizzare anche i giorni e gli orari in cui il personale è reperibile, ovvero non lavorano ma possono essere chiamati in reparto per delle urgenze. Turni e reperibilità del personale possono cambiare di settimana in settimana.

Per ciascuna partoriente si memorizzano i consueti dati anagrafici (codice fiscale, nome, cognome, indirizzo e recapito telefonico), eventuali patologie rilevanti e allergie a particolari sostanze.

Il sistema deve inoltre memorizzare i dati di ciascun parto, ovvero la data, l'orario di inizio e quello di fine, la sala in cui si svolge e i dati della partoriente. Si tenga conto del fatto che il reparto dispone di un numero limitato di sale e che non possono svolgersi contemporaneamente due parti nella stessa sala. I parti possono essere naturali o prevedere un taglio cesareo. I parti naturali richiedono la presenza di un'ostetrica e di un'assistente, mentre per i parti con taglio cesareo è indispensabile anche la presenza di un medico.

In ciascun parto nascono uno o più bambini. Per ciascun neonato, identificato da un codice univoco, si memorizzano il peso, l'altezza e eventuali note che descrivono i problemi che possono essere insorti durante il parto.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Si modelli tramite *diagramma degli stati* il seguente scenario relativo al funzionamento di un ascensore. Inizialmente l'ascensore rimane in attesa (con le porte chiuse) di ricevere prenotazioni di chiamata. Dopo aver ricevuto una prenotazione l'ascensore aggiorna il percorso da svolgere e si mette in movimento. Nuove prenotazioni possono essere effettuate in qualsiasi momento e devono essere incorporate nel percorso per poi essere eventualmente soddisfatte (eccetto al verificarsi di guasti). L'ascensore si ferma ogni volta che arriva ad un piano per il quale esiste la corrispettiva prenotazione; dopo aver aperto le porte, attende 30 secondi prima di richiuderle ed eventualmente rimettersi in movimento. In qualsiasi momento si possono verificare guasti tali per cui l'ascensore è costretto a bloccarsi fino a quando non viene riparato. Dopo ogni blocco vengono azzerate le prenotazioni in coda, viene segnalato agli utenti la non disponibilità (tramite apposita luce lampeggiante) e infine, dopo che il guasto è stato risolto, l'ascensore si muove per tornare alla posizione di default, cioè il piano terra (*8/32 punti*).

3) Si definiscano le seguenti qualità del software: *riusabilità, trasparenza, correttezza* (*6/32 punti*).

4) Si spieghi cos'è un *nodo* in un diagramma di deployment, specificando i due modi standard in cui un nodo può essere stereotipato (*6/32 punti*).

1) E' dato il seguente dominio applicativo:

Un gestore di telefonia mobile vuole realizzare un software per la gestione dei suoi clienti, dei piani tariffari e del traffico telefonico. I piani tariffari sono di tipo abbonamento o carta prepagata, hanno un nome, un codice, una data di inizio validità e, se non più attivi, una data di fine. Per ogni piano si memorizzano, inoltre, una descrizione e i costi del piano oltre alla lista delle opzioni attivabili. Ogni opzione ha una descrizione, un costo di attivazione, una lista di piani per i quali è ammessa e un periodo di validità (anche infinito). Di tutti i clienti sono mantenuti i dati anagrafici (CF, nome e cognome). Un cliente può possedere una o più SIM card; ciascuna SIM card è identificata da un numero telefonico, è intestata a un cliente, ha un piano telefonico attuale ed eventualmente delle opzioni, un credito residuo (se si tratta di abbonamento) e il totale del credito consumato (utile per selezionare i clienti più assidui). Infine sono registrate tutte le telefonate in cui il chiamante è un numero della compagnia: per ciascuna chiamata vengono registrati la durata, il costo e il numero telefonico del destinatario.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*10/32 punti*)

2) Si modelli tramite *diagramma degli stati* il funzionamento di un metronomo digitale. Il metronomo è uno strumento usato in musica per misurare il tempo e scandire la ritmica. Inizialmente il metronomo è spento e può essere acceso tenendo premuto l'apposito pulsante per tre secondi. Una volta acceso, il metronomo emette un suono (*beat*) con una frequenza detta *Beats Per Minute* (BPM) il cui valore viene visualizzato tramite apposito display; più alti sono i BPM e minore è il periodo di tempo tra un beat e l'altro. Il primo beat di ogni quattro viene enfatizzato emettendo un suono diverso dagli altri. Il valore dei BPM di default (ovvero alla prima accensione) è pari a ottanta ma può essere aumentato o diminuito dall'utente tramite appositi comandi in ogni momento in cui il metronomo è acceso. Anche in caso di spegnimento il valore corrente dei BPM deve essere mantenuto in modo da essere utilizzato all'accensione successiva. Nel caso in cui la batteria che alimenta il metronomo si esaurisca, quest'ultimo cessa di funzionare fino a quando essa non viene sostituita (*7/32 punti*).

3) Si definisca la qualità del software *riusabilità* e si spieghi in quali modi essa è incoraggiata dal paradigma a oggetti (*6/32 punti*).

4) Si illustrino brevemente le caratteristiche fondamentali di *unified process* (*6/32*).

1) E' dato il seguente dominio applicativo:

L'ente governativo americano NASA conduce diversi programmi di ricerca volti alla raccolta ed elaborazione di dati, ad esempio sulla composizione chimica dell'atmosfera e i campi gravitazionali di diversi pianeti. Le varie attività sono organizzate in missioni, ognuna delle quali è caratterizzata da un nome e da una data di inizio. Per raccogliere dati, ogni missione si avvale di una o più sonde, le quali possono essere mobili (per es. satelliti) oppure stazionarie (per es. stazioni terrestri). Ogni sonda è descritta da un nome (per es. *Hubble*) e, nel caso sia stazionaria, dalla sua posizione. Su ogni sonda sono montati diversi sensori (identificati da un apposito codice alfanumerico), ognuno dei quali esegue delle rilevazioni che, insieme all'istante in cui sono state effettuate, vengono inviate ai laboratori terrestri. Le rilevazioni possono essere immagini oppure statistiche varie (per es. concentrazione di CO₂ nell'atmosfera). Ogni rilevamento viene memorizzato per poter essere poi analizzato. In particolare, ogni utente (per es. uno scienziato) può produrre un'analisi applicando una o più tecniche di processing (clustering, outlier detection, etc.) a una o più delle rilevazioni disponibili. Infine, per filtrare le immagini raccolte vengono occasionalmente utilizzate tecniche di *crowdsourcing* in cui gli utenti analizzano manualmente alcune immagini e assegnano loro un punteggio per indicarne la qualità (nitidezza, centratura, etc.)

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)
- 2) Si modelli tramite *diagramma di sequenza* (*8/32 punti*) il seguente scenario relativo al funzionamento del servizio web per l'analisi dei dati atmosferici raccolti dai laboratori dell'ente governativo americano NASA. Un utente che vuole accedere ai dati raccolti deve in primo luogo effettuare il login inviando *username* e *password* al servizio web, il quale dopo aver controllato la validità delle credenziali risponde con il relativo esito (*accesso consentito* o *accesso negato*). In caso di esito positivo, l'utente può interrogare il servizio inviando l'apposita richiesta con i relativi parametri (*latitudine*, *longitudine* e *istante temporale*). Ricevuta la richiesta dall'utente, il servizio crea un apposito processo per gestire le operazioni necessarie e restituisce all'utente l'identificativo del processo. In qualsiasi momento durante l'elaborazione l'utente può decidere di teminare il processo relativo alla sua richiesta. Per terminare una richiesta l'utente deve inviare l'opportuno comando (*kill <ID processo>*) al servizio. Nel caso in cui non venga terminato, il processo di elaborazione restituisce il risultato della richiesta utente al servizio principale che a sua volta provvederà a comunicarlo all'utente.
- 3) Si descrivano le attività strutturali dell'*extreme programming*, illustrando in particolare il ruolo delle *user story* (*6/32 punti*).
- 4) Si elenchino e si commentino brevemente le fasi del *ciclo di vita del software* (*6/32 punti*).

1) E' dato il seguente dominio applicativo:

Un coro amatoriale mantiene un database per la gestione del repertorio, dei concerti e delle quote di iscrizione. Il coro è costituito da 5 settori (tenori, baritoni, bassi, contralti e soprani), ciascuno dei quali include un numero variabile di coristi; tra i coristi di ogni settore, uno svolge il ruolo di capo-settore. Per ogni corista, oltre ai dati anagrafici, si memorizza la data di ingresso nel coro (i dati dei coristi usciti dal coro vengono cancellati). Ogni corista paga una quota fissa di iscrizione annuale, e si vuole tenere traccia di quali coristi hanno già pagato la loro quota per ciascun anno. Il repertorio del coro include più di 100 brani; per ciascuno si memorizzano, oltre al titolo e agli autori, il pdf della partitura e il genere (spiritual, gospel o worksong). Il coro propone alcuni concerti standard di diversa durata (per esempio, "Viaggio nel Negro Spiritual"), ciascuno comprendente una specifica selezione di brani. Si vuole tenere traccia di tutti i concerti a calendario (anche quelli degli anni passati), sia di tipo standard sia personalizzati; ogni concerto è descritto da data, luogo, eventualmente tipo standard di appartenenza, scaletta effettiva dei brani (che può differire marginalmente da quella standard), ed elenco dei coristi partecipanti. La gestione del repertorio e del calendario concerti viene effettuata dal direttore del coro, quella delle quote di iscrizione dal cassiere (uno dei coristi a rotazione), quella dei coristi membri di ciascun settore dal corrispondente capo-settore.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (*3/32 punti*)
- 1.2) un diagramma delle classi (*9/32 punti*)

2) Si modelli tramite diagramma di attività il seguente processo di stipula di polizze di assicurazione medica (*8/32 punti*).

Il processo inizia con la ricezione di una richiesta di assicurazione: un impiegato registra i dati relativi alla richiesta ricevuta, che vengono poi passati ad un esaminatore incaricato di valutare la richiesta. L'esaminatore può richiedere consulenza medica o commerciale; se è necessaria una consulenza medica la pratica viene passata a uno specialista, che può decidere immediatamente sul caso oppure visitare il cliente. In caso invece di richiesta di supporto dal punto di vista commerciale, i dati vengono passati a un advisor che comunica un parere. Dopo aver ricevuto le eventuali consulenze richieste, inizia la fase di decisione. Nel caso in cui la decisione venga presa da un impiegato di livello "junior", la richiesta viene controllata anche da un impiegato "senior". Se la richiesta viene accolta, i dati vengono registrati e la polizza viene preparata; in caso invece di rifiuto, il cliente viene informato tramite lettera. In ogni caso, il processo termina con l'archiviazione della pratica.

3) Si illustrino le differenze tra *analisi orientata agli oggetti, agli stati e alle funzioni*, facendo anche esempi di diagrammi UML che ricadono in ciascuna categoria (*6/32 punti*).**4) Si illustri il principio di *modularità* commentandone i benefici sullo sviluppo del software (*6/32 punti*).**

1) Dato il seguente dominio applicativo:

Tramite l'applicazione online GiastIt è possibile ordinare pietanze dai numerosi ristoranti convenzionati, ciascuno appartenente a una categoria (per esempio, *ristorante cinese*). Per ogni ristorante è possibile visualizzare informazioni quali *nome*, *indirizzo* e *orari di apertura*. Ogni ristorante mette inoltre a disposizione un elenco delle pietanze (*nome*, *descrizione* e *prezzo unitario*) che è possibile ordinare. Per facilitarne la consultazione, tutte le pietanze vengono organizzate in categorie (es. antipasti, carne, etc.); una pietanza fa riferimento ad una ed una sola categoria. I clienti registrati possono effettuare ordini direttamente online e possono scegliere tra l'opzione *asporto* o *a domicilio*. Ad ogni ordine è associato l'orario per cui dovrà essere pronto; per ognuna delle pietanze richieste è specificata la quantità. Nel caso di ordini a domicilio si vuole inoltre conoscere l'indirizzo di consegna e il fattorino incaricato (ogni fattorino fa riferimento ad uno ed un solo ristorante).

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)
- 2) Si modelli tramite *diagramma di attività* (8/32 punti) il seguente scenario relativo all'ordinazione di pietanze tramite l'applicativo descritto nel primo esercizio. Dopo aver consultato il menu, il cliente aggiunge una ad una tutte le pietanze desiderate utilizzando l'apposita interfaccia messa a disposizione dall'applicazione. Quando tutte le portate sono state aggiunte il cliente seleziona la modalità di ritiro dell'ordine che preferisce (asporto o a domicilio) e procede alla conferma dell'ordine tramite pagamento. Una volta ricevuto l'ordine il ristorante attende il momento giusto (in modo tale che le pietanze non si raffreddino) prima di iniziare a preparare le pietanze richieste e, una volta pronte, in caso di ordine a domicilio vengono passate al fattorino incaricato che provvede a consegnarle al cliente. In caso l'ordine sia da asporto il cliente si reca al ristorante per ritirare le pietanze richieste (notare che il cliente può recarsi al ristorante mentre le pietanze sono ancora in preparazione).
- 3) Si elenchino e si discutano brevemente 3 *qualità del software* (6/32 punti).
- 4) Si descriva brevemente il metodo *function points* spiegandone il ruolo e l'utilità (6/32 punti).

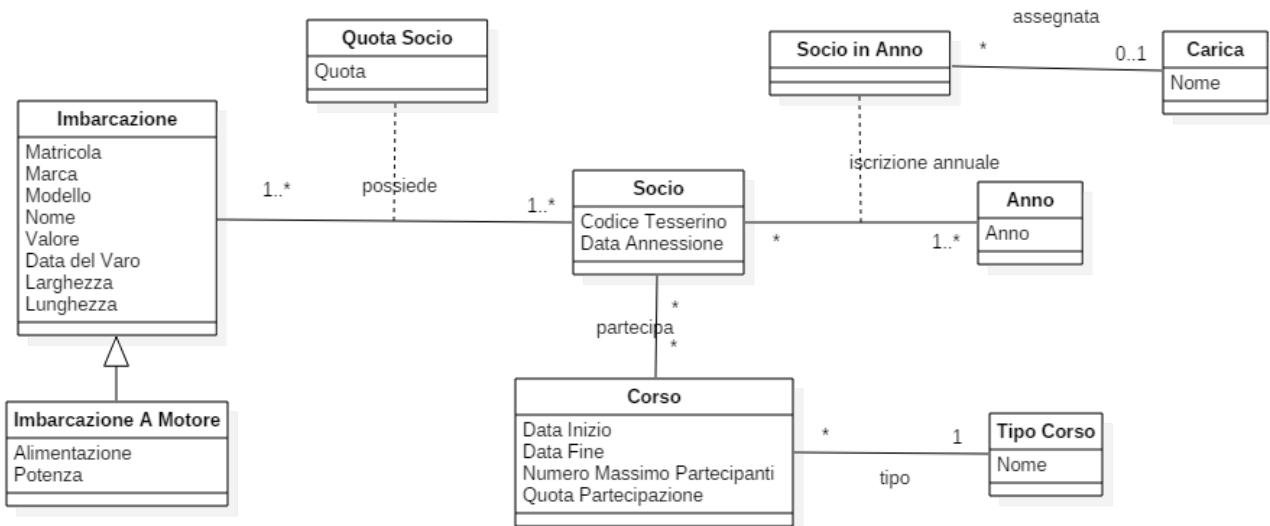
1) Il sito kaffe.it permette agli utenti registrati di ordinare alcuni tipi di prodotto (caffè o tisane) per macchine da caffè; il caffè (la cui intensità può essere bassa, media o alta) può essere fornito in capsule o in cialde, mentre le tisane (contenenti una serie di erbe) sono disponibili solo in cialde. Ogni prodotto è compatibile con diversi modelli di macchinette e può essere contraddistinto da un aroma particolare (per esempio, vaniglia o cocco). Il sito non mette in vendita i singoli esemplari dei prodotti, bensì delle confezioni che contengono una determinata quantità di un prodotto ad un determinato prezzo. Gli utenti (di cui si conoscono le generalità e le credenziali di login) possono effettuare ordini in cui includere una o più unità di almeno una confezione; per ogni ordine vengono memorizzati la data e l'importo totale delle confezioni acquistate. Il sito offre inoltre agli utenti la possibilità di fornire delle recensioni sulle confezioni acquistate, comprensive di un testo libero e di una valutazione numerica da 1 (scarso) a 5 (ottimo). Le recensioni più utili possono essere infine segnalate dagli utenti attraverso l'apposizione di un "like".

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Dato il diagramma delle classi in figura, effettuare il progetto delle associazioni (8/32 punti) tenendo conto del seguente carico di lavoro:

- a) Visualizzare il valore totale delle imbarcazioni possedute da un dato socio (1 volta al giorno).
- b) Per ogni tipo di corso visualizzare il numero medio di partecipanti e il numero medio di anni di appartenenza al circolo dei soci che hanno partecipato a corsi di quel tipo (1 volta alla settimana).
- c) Data la matricola di un'imbarcazione, verificare se ad almeno uno dei soci a cui appartiene è mai stata assegnata una carica (1 volta all'anno).



3) Si elenchino e si descrivano brevemente i sei criteri di usabilità di un'interfaccia utente (6/32 punti).

4) Si illustrino le caratteristiche comuni ai *modelli prescrittivi* di produzione del software; si citi poi almeno un esempio di modello prescrittivo e uno di modello non prescrittivo (6/32 punti).

1) L'agenzia immobiliare *DanceCasa* necessita di un software per gestire la compravendita di immobili a Cesena. Di ogni immobile si conosce l'indirizzo, l'importo richiesto, il numero di vani ed i metri quadrati commerciali, oltre alle generalità del venditore. Nel caso di appartamenti, si conosce anche il numero dell'appartamento all'interno del condominio, il piano, l'eventuale presenza dell'ascensore e le spese condominiali. Gli immobili sono suddivisi in diverse aree geografiche della città, ciascuna delle quali è gestita da almeno un agente immobiliare. Gli agenti si occupano di gestire le visite dei clienti nei diversi immobili. Un cliente può visitare diversi immobili (anche più volte), ma non può visitare lo stesso immobile con agenti diversi; inoltre, si vogliono memorizzare le date di ogni singola visita. Il software deve gestire anche le proposte di acquisto di un immobile presentate da un cliente, indicando l'importo che è stato offerto e l'eventuale accettazione della proposta da parte del venditore. Se la proposta viene rifiutata, il cliente è libero di effettuare una nuova proposta; se la proposta viene accettata, l'agente che ha seguito il cliente durante le visite riceve una percentuale sull'importo della vendita.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Si modelli tramite *diagramma degli stati* (8/32 punti) il seguente scenario relativo ad un'applicazione per lo streaming di musica.

Dopo che l'applicazione è stata aperta, per poter iniziare lo streaming è necessario selezionare una delle playlist presenti in catalogo; i brani della playlist selezionata vengono poi riprodotti in ordine. Durante la riproduzione dei brani l'utente può mettere in pausa (e riprendere in seguito l'ascolto), selezionare un'altra playlist o, se presente, passare al brano successivo o precedente tramite apposite funzioni next e previous. L'applicazione è fruibile gratuitamente ma sono previsti intermezzi pubblicitari tra la riproduzione di un brano e l'altro. Nello specifico, un spot pubblicitario viene riprodotto ogni n brani, dove n è un valore random compreso tra 2 e 20; si noti che n viene inizializzato all'accensione e successivamente rivalorizzato dopo ogni riproduzione di pubblicità. Ai fini della gestione degli intermezzi pubblicitari si contano come brani riprodotti anche tutti quei brani che sono stati saltati tramite funzioni next e previous. Si fa notare infine che la pubblicità non può né essere saltata, né essere messa in pausa (l'applicazione può comunque essere chiusa in ogni momento).

3) Si illustrino le diverse tipologie di *tecniche di verifica del software statiche* (6/32 punti).

4) Si descrivano brevemente le caratteristiche dei diversi tipi di *manutenzione del software* (6/32 punti).

1) Il sistema web-based *BigHut* per il versioning del codice mette a disposizione dei suoi utenti un servizio di hosting e gestione di repository di codice. Per accedere ai vari servizi, un utente deve effettuare il login tramite username e password. Dopo il login, un utente può creare un nuovo progetto e, al suo interno, inserire più repository; sia progetti che repository sono definiti da nome e data di creazione. Per ciascun repository, l'utente creatore del progetto corrispondente può concedere ad altri utenti dei permessi, per esempio lettura e modifica (utilizzare un generico campo testuale per descriverli tutti). Ogni repository è formato da una sequenza di commit (ognuno definito da un ID e una data) che coinvolge uno o più file sorgente (ognuno definito da un path). Ogni commit è effettuato da un utente (tra quelli che contribuiscono a quel repository) e può esprimere diverse operazioni, una per ogni file sorgente coinvolto. Un'operazione può essere di tre tipi: aggiunta, modifica (descritta da un generico campo testuale) e rimozione. Esempio: con un singolo commit è possibile aggiungere un nuovo file sorgente e modificarne un altro. Infine, un utente può aprire una issue (definita da un ID e una descrizione) relativa a uno specifico repository per segnalare eventuali bug, proposte di miglioramento, ecc. Lo stato iniziale di una issue è "aperto", ma può in seguito diventare "chiuso" (per esempio, perché il bug viene risolto).

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Modellare tramite *diagramma di attività UML* (8/32 punti) il seguente scenario, in cui viene descritta la sottomissione di una proposta di acquisto di un immobile da parte di un soggetto acquirente a un soggetto venditore e il seguente iter burocratico.

Inizialmente l'acquirente formula una proposta di acquisto relativa un immobile di interesse, che viene valutata dal rispettivo venditore. Il venditore può decidere se accettare la proposta, rifiutarla o se formulare una controproposta da sottoporre all'acquirente. In quest'ultimo caso, l'acquirente può decidere a sua volta se accettare la controproposta, rifiutarla o se ripartire con una nuova proposta. Una volta che acquirente e venditore hanno trovato un accordo, la proposta viene presa in gestione da un notaio, il quale effettua tutte le verifiche necessarie e, in caso di accertamenti positivi, prepara la documentazione della compravendita. A questo punto, acquirente e venditore possono procedere con la compravendita, firmando (entrambi) la documentazione e pagando (l'acquirente) gli importi relativi alla compravendita e alle spese notarili. Il notaio procede alla finalizzazione delle pratiche non appena tutte le attività precedenti sono state eseguite.

3) Qual è il significato delle relazioni di *raffinamento* e *realizzazione* in UML? (6/32 punti).

4) Nell'ambito delle interfacce utente, si spieghino brevemente le differenze tra i modelli *multi-window*, *multi-document* e *multi-paned* (6/32 punti).

1) Ogni anno, durante il mese di agosto, a Edimburgo si svolge il festival Fringe, nel quale si radunano artisti e visitatori da tutto il mondo. Durante il Fringe, molti pub, teatri ed edifici pubblici mettono a disposizione i loro locali per ospitare vari spettacoli. Ogni venue (descritta da un nome ed una posizione geografica) mette a disposizione degli slot temporali, definiti da una data/ora iniziale ed una data/ora finale, per le performance dei vari spettacoli (ognuno definito da un nome ed una descrizione). Ovviamente, in ogni slot di una data venue può essere eseguita una sola performance. Ogni spettacolo appartiene inoltre ad una determinata categoria ed è organizzato ed eseguito da un uno o più artisti; ogni artista è descritto da un nome d'arte e dalla sua nazionalità. Per poter assistere ad una performance, un visitatore (descritto da nome, cognome e nazionalità) deve acquistare il corrispettivo biglietto. L'acquisto può essere effettuato sia sul posto, oppure online via prenotazione. Nel secondo caso viene anche assegnato un numero di posto riservato all'acquirente. Ogni biglietto, acquistato tramite prenotazione o meno, espone la data/ora di acquisto, il prezzo e un codice che il visitatore può utilizzare (accedendo al sito del festival) per rilasciare una recensione sulla performance; la recensione è costituita da una valutazione (da 1 a 5 stelle) e da un commento testuale.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Modellare tramite *diagramma di sequenza UML* (8/32 punti) il seguente scenario, in cui viene descritta la sottomissione di una tesi di laurea da parte di uno studente.

Una volta collegato al sistema di gestione della carriera universitaria (si supponga che il login sia già stato effettuato), lo studente invia al server il file PDF della propria tesi di laurea. Prima di inserirla nel database, il server avvia un'istanza del software antiplagio per verificare l'originalità della tesi. Se il controllo non riscontra irregolarità, la tesi viene inviata al DBMS e l'utente viene notificato dell'avvenuta sottomissione. Qualora venga invece riscontrato un plagio, il server invia al Servizio Antiplagio dell'Università una debita segnalazione (inclusiva dei dati dello studente e del documento di tesi) e notifica lo studente dell'avvenuta segnalazione. In entrambi i casi, il server distrugge l'istanza del software a controllo terminato.

3) Nell'ambito della misurazione del software, si elenchino i principali fattori che ne determinano il costo (6/32 punti).

4) Si elenchino i *meccanismi di astrazione* utilizzati durante il processo di analisi (6/32 punti).

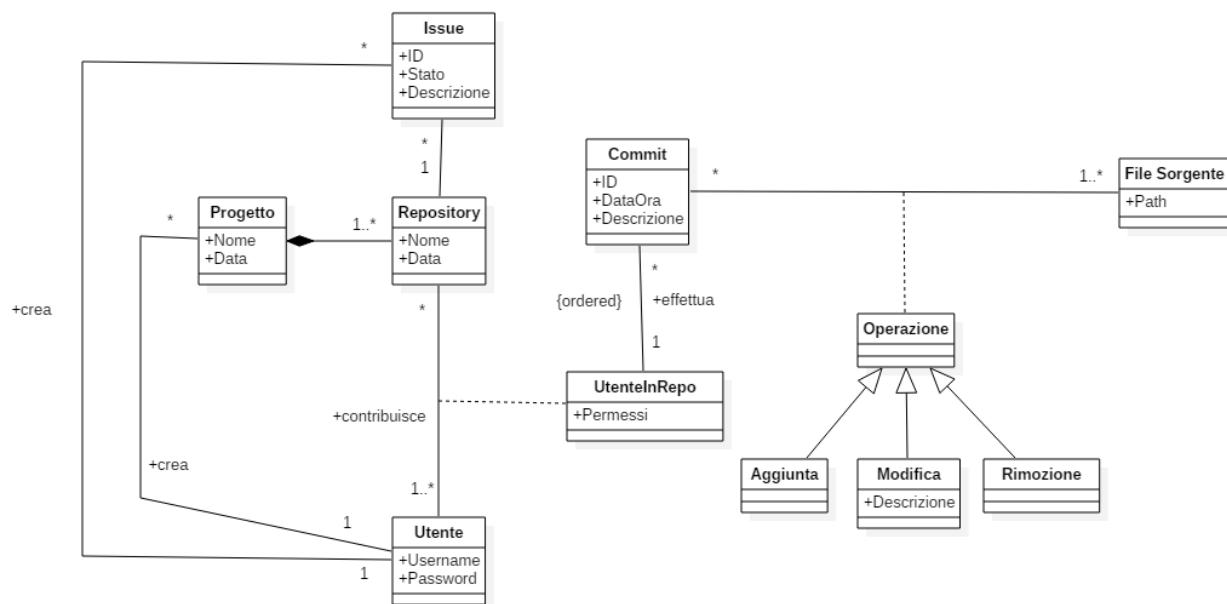
1) L'applicazione "Pastiyas" consente alle persone di effettuare dei trasferimenti di denaro attraverso il proprio smartphone; i trasferimenti possono essere effettuati sia verso altre persone, sia verso aziende. Le persone che intendono usare l'applicazione devono comunicare i propri dati (nome, cognome, codice fiscale e documento), mentre le aziende devono comunicare un nome ed una partita IVA; in entrambi i casi va inoltre comunicato l'IBAN di un conto corrente. Per ogni trasferimento di denaro si tiene traccia della data e dell'orario, dell'importo trasferito e di un'eventuale causale. Un trasferimento può avvenire tra due persone, oppure da una persona ad un'azienda; in quest'ultimo caso, l'applicazione applica una commissione sul trasferimento. Ogni 15 giorni, l'applicazione verifica i trasferimenti che sono stati fatti da ciascuna persona/azienda ed esegue un congauglio sul conto corrente della stessa (ossia calcola l'importo totale che la persona/azienda deve pagare - o ricevere - ed esegue la transazione bancaria); per ogni conguaglio si memorizzano l'importo (che può essere positivo o negativo) e la data. Per incentivare la diffusione, l'applicazione consente alle persone di invitarne delle altre; quando una persona si registra tramite invito, viene assegnato ad entrambi (invitato ed invitante) un bonus di denaro, il cui importo può essere variabile.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
1.2) un diagramma delle classi (9/32 punti)

2) Dato il diagramma delle classi in figura, effettuare il *progetto delle associazioni* (8/32 punti) tenendo conto del seguente carico di lavoro:

- a) Visualizzare lo storico di tutte le operazioni effettuate da un dato utente su un dato file sorgente (1 volta al giorno).
 - b) Visualizzare il numero di commit eseguiti da un dato utente in certo intervallo temporale (2 volte alla settimana).
 - c) Per ogni progetto e ogni repository in esso contenuto, visualizzare l'elenco delle issue associate (1 volta al mese).



3) Si illustrino i principi fondanti dei *modelli di processo agili*, e si descrivano in particolare le attività strutturali dell'*extreme programming* (6/32 punti).

4) Si illustri il concetto di *incapsulamento* nel paradigma a oggetti, spiegandone pro ed eventuali contro (6/32 punti).

1) L'applicazione spoti5 consente ai suoi utenti di ascoltare musica online e di organizzare le canzoni preferite in playlist. Gli utenti (di cui si conoscono username, password) si dividono in premium e standard. Ogni utente può seguire/essere seguito da altri utenti e può possedere più playlist (di cui spoti5 registra titolo e data di creazione). Le playlist possono essere seguite da più utenti e hanno una duplice natura: collaborativa (a cui possono partecipare attivamente più utenti) e non collaborativa (pubblica o privata). Ad ogni playlist può essere aggiunto un numero arbitrario di canzoni (descritte dal titolo, dal genere e dal nome dell'artista/band). Si vuole tenere traccia del fatto che una canzone possa essere aggiunta alla playlist a seguito di un suggerimento da parte di spoti5. Infine, l'applicazione memorizza lo storico degli ascolti di ogni utente.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Modellare tramite *diagramma di attività UML* (8/32 punti) il seguente scenario, in cui viene descritta la sottomissione di un articolo scientifico all'organizzatore di una conferenza.

Previa autenticazione presso il sito della conferenza, l'autore invia l'articolo che vuole sottomettere. In prima istanza, l'organizzatore verifica che l'argomento dell'articolo sia coerente con l'ambito della conferenza. In caso negativo, l'articolo viene semplicemente rifiutato; in caso positivo, l'articolo entra nel processo di revisione. L'organizzatore individua quindi un revisore qualificato da incaricare per la valutazione dell'articolo. Ricevuta notifica, il revisore decide se accettare o meno l'incarico (in caso contrario, l'organizzatore deve individuare un nuovo revisore). Il revisore che ha accettato l'incarico invia la revisione all'organizzatore, il quale decide se accettare o rifiutare l'articolo. In caso di accettazione, l'autore viene notificato e conclude l'iter pagando la tassa di iscrizione alla conferenza. Nel diagramma si modelli anche lo stato dell'articolo attraverso il processo.

3) Si illustri il concetto di *polimorfismo* nel paradigma a oggetti, spiegandone pro ed eventuali contro (6/32 punti).

4) Si illustri in termini generali il concetto di *dipendenza* in UML e si diano almeno tre esempi di stereotipi di dipendenza (6/32 punti).

1) Il sistema FarmER4.0 gestisce le anagrafiche relative ai terreni utilizzati a scopo agricolo o di allevamento a livello regionale, insieme a dati statistici raccolti in tempo reale attraverso sensori in loco. Al sistema possono registrarsi sia persone fisiche (invia nome, cognome e codice fiscale) che giuridiche (invia ragione sociale e partita iva); ciascuna persona può possedere più terreni, di cui si memorizzano la geometria georeferenziata e l'estensione in ettari. Ogni terreno è composto da particelle, le quali coprono una determinata percentuale del terreno. Ogni particella è destinata ad un uso specifico, che può essere agricolo (nel qual caso si memorizza il tipo coltura), di allevamento (nel qual caso si memorizzano le specie animali presenti), o altro. Persone fisiche e giuridiche possono inoltre dotarsi di uno o più sensori (di cui esistono diverse tipologie e modelli), da collocare in un determinato terreno (un terreno non può essere dotato di più sensori dello stesso tipo). Per ogni sensore collocato, si vogliono conoscere le coordinate esatte e le misurazioni rilevate ogni giorno con cadenza oraria.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Modellare tramite *diagramma di sequenza UML* (8/32 punti) il seguente scenario che modella in maniera astratta la modalità di interazione client server in una specifica applicazione multi-tier.

Il diagramma descrive l'esecuzione di codice server-side a partire dal click di un bottone in una pagina web. Il browser (client) reagisce alla pressione del bottone invocando la funzione click() da cui è inviata una richiesta asincrona al web server. Il web server esegue quindi il parsing dei parametri: in caso di fallimento restituisce l'errore al browser, il quale invoca la funzione error(); in caso di successo, il web server istanzia un oggetto di classe Foo su cui invoca il metodo bar(parameters). Al termine dell'esecuzione il web server distrugge l'oggetto creato, procede con il preparazione del risultato e lo restituisce al browser che invoca la funzione success().

3) Si spieghi (indicando anche alcuni esempi) la differenza tra le fonti di costo ed i fattori di costo nell'ambito della misurazione del software (6/32 punti).

4) Nell'ambito della progettazione di interfacce utente (GUI), si spieghi cosa si intende per *project standard* riportando alcuni esempi (6/32 punti).

1) La società Areh gestisce i contratti relativi ai consumi energetici nel territorio emiliano-romagnolo. Un contratto (o utenza) viene stipulato da un cliente, del quale si conoscono dati anagrafici, indirizzo di fatturazione ed eventuale IBAN su cui addebitare automaticamente le bollette. L'utenza (di cui si memorizzano il codice, l'indirizzo effettivo, il numero di residenti nell'indirizzo e la data di attivazione) comprende un insieme di bollette, che vengono solitamente emesse con cadenza bimestrale. Ogni bolletta indica la data di inizio, quella di fine, l'importo totale, lo status (pagata/non pagata) e la data di scadenza del pagamento; se il cliente tarda nel pagamento, viene emessa una mora (di cui si memorizzano l'importo e lo status). Le bollette riportano anche gli importi di dettaglio, i quali sono limitati ad un'unica voce nel caso in cui la bolletta riguardi i rifuti, mentre si scompongo in importi per acqua, luce e gas nel caso in cui la bolletta riguardi l'energia; a questi ultimi può essere aggiunto anche l'importo del canone tv. Ad ogni bolletta fanno riferimento un insieme di rilevamenti eseguiti all'indirizzo dell'utente per verificare i consumi effettivi. Di ogni rilevamento si conoscono la data di esecuzione, la tipologia (eseguito dal cliente/eseguito dall'addetto/stimato) ed i valori rilevati sui contatori. Se il rilevamento è stato eseguito da un addetto, si memorizzano nome, cognome e tesserino di quest'ultimo. I valori rilevati dipendono dal tipo di energia: nel caso di acqua e gas si tratta di un unico valore, mentre nel caso della luce si tratta di tre valori distinti (fascia1, fascia2 e fascia3).

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un diagramma dei casi d'uso (3/32 punti)
- 1.2) un diagramma delle classi (9/32 punti)

2) Si modelli tramite diagramma degli stati UML (8/32 punti) il seguente scenario che modella il funzionamento di uno scheduler.

Lo scheduler ha il compito di eseguire dei task accodati in due liste (Q_1 e Q_2) in base alla priorità: i task appartenenti a Q_1 sono eseguiti sempre prima dei task presenti in Q_2 . Una volta acceso, lo scheduler attende nuovi task da eseguire (idle). Ricevuti uno o più task, lo scheduler entra nell'articolata fase di esecuzione. In prima istanza, lo scheduler verifica la presenza di task in Q_1 . Se ci sono task (Q_1 non è vuota), lo scheduler preleva il task più corto (T_1), esegue un'operazione di "context switch" ed esegue T_1 . Altrimenti lo scheduler verifica la presenza di task in Q_2 . Se ci sono task (Q_2 non è vuota), lo scheduler preleva il primo elemento della lista (T_2), esegue un'operazione di "context switch" e procede con l'esecuzione di T_2 , altrimenti termina la fase di esecuzione e ritorna in attesa di nuovi task. L'esecuzione di T_1 richiede di: ottenere il lock sulle risorse necessarie, eseguire il task, e rilasciare i lock precedentemente ottenuti. L'esecuzione di T_2 prevede gli stessi passi di T_1 , tuttavia presenta due differenze sostanziali: se in questa fase lo scheduler riceve un segnale "preempt" (cioè lo scheduler riceve un task T_1 durante l'esecuzione di T_2), lo scheduler sospende T_2 , lo inserisce in testa a Q_2 , esegue un "context switch" e procede con l'esecuzione del nuovo task T_1 ; se il tempo di esecuzione ha raggiunto la soglia "timeslot", lo scheduler accoda T_2 in coda a Q_2 e procede con la selezione del prossimo elemento in Q_2 . In qualsiasi momento lo scheduler può essere terminato con segnale "kill".

3) (6/32 punti) Si spieghi (riportando anche un esempio) il concetto di polimorfismo, descrivendone i vantaggi in abbinamento all'istanziamento dinamico (*late binding*).

4) (6/32 punti) Nell'ambito della produzione del software, si descrivano le caratteristiche dei modelli evolutivi e delle tecniche di prototipazione adottabili.

1) Si vuole rappresentare un sistema per la gestione elettronica delle votazioni. Le votazioni sono contraddistinte da una descrizione (es: *votazione per la Camera dei Deputati*, oppure *per il Senato*) e da una tipologia (es: *politiche, amministrative*) e sono raggruppate in tornate, identificate da una data. A una votazione si presentano più partiti, contraddistinti da un nome e da una sede. Esistono diverse sezioni elettorali, di cui si memorizzano l'indirizzo, il comune ed il numero; in occasione di una votazione, un ufficio elettorale si insedia presso ciascuna sezione. L'ufficio (a cui è assegnato un ID univoco) è composto da un presidente (di cui si memorizza l'ID di registrazione presso l'Albo dei Presidenti), scelto automaticamente dal sistema, e da un segretario, scelto e registrato dal presidente. Eventualmente possono prendervi parte anche dei rappresentanti di lista, i quali fanno riferimento ad un determinato partito; la registrazione dei rappresentanti di lista all'ufficio deve essere gestita dal segretario. Si consideri che una persona possa ricoprire uno solo dei ruoli sopra menzionati. Per ciascuna votazione, gli elettori (di cui si conoscono nome, cognome, credenziali, carta d'identità e numero della tessera elettorale) accedono al sistema ed esprimono un unico voto, rivolto ad uno dei partiti che si sono presentati (oppure rivolto a nessuno); i voti registrati devono tenere traccia anche dell'orario, dell'ufficio elettorale presso cui il voto è stato espresso e di eventuali dichiarazioni lasciate dall'elettore (le quali vengono registrate dal segretario). Alla fine della votazione, il conteggio dei voti dell'ufficio viene azionato e validato dal relativo presidente.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con *diagramma delle attività* UML (8/32 punti) il funzionamento di Oracolo, applicazione i cui giocatori si sfidano per indovinare un token segreto. L'applicazione è composta da una componente visuale (GUI) e da un oracolo, il cui compito è quello di dispensare saggi consigli per indirizzare i “guess” dei giocatori. Quando un giocatore inizia la partita, l'oracolo carica i consigli precedentemente dispensati e ne demanda la visualizzazione alla GUI. I consigli sono visualizzati per 30s. Allo scadere dei 30s il giocatore ha un breve lasso temporale (timeout) per inserire il guess. Se il giocatore non inserisce il guess in tempo, la GUI mostra il messaggio “Lost!” e la partita termina. Se il guess è stato inserito in tempo, la GUI visualizza il guess inserito e l'oracolo lo controlla. Se il guess è corretto, l'oracolo genera un nuovo segreto e rimuove i consigli precedentemente dispensati. Se il guess è errato, l'oracolo genera un nuovo consiglio e lo accoda a quelli esistenti. Una volta mostrato e verificato il guess, la GUI visualizza l'esito e la partita termina.

3) Si elenchino le fasi del ciclo di vita del software, illustrando in particolare il ruolo dello *studio di fattibilità* (6/32 punti).

4) Si descriva brevemente il metodo *function points* spiegandone anche il ruolo e l'utilità (6/32 punti).

1) Si vuole rappresentare un sistema per la gestione dei biglietti in un cinema multisala. Il cinema è composto da diverse sale (di cui si memorizzano la numerazione e la capienza), ciascuna delle quali contiene diversi posti a sedere (per ciascuno dei quali si memorizzano la numerazione e l'eventuale esclusività per portatori di handicap). I film (contraddistinti da un nome, una durata ed un eventuale limitazione ad un pubblico adulto) possono essere proiettati in più giorni e in più sale contemporaneamente. Previa registrazione sul portale web del cinema (fornendo nome, cognome, username e password), i clienti possono effettuare prenotazioni (non vincolanti e annullabili in qualunque momento) per uno o più posti di una determinata proiezione; i biglietti possono essere poi ritirati presso casse automatiche inserendo il codice della prenotazione ed effettuando il pagamento. Si noti che un biglietto viene emesso per ogni posto. Se la prenotazione non viene riscossa entro 30 minuti dall'inizio della proiezione, i posti possono essere venduti ad altre persone tramite acquisto diretto presso le casse automatiche. Per le prenotazioni si vogliono memorizzare lo stato ("attiva", "annullata" o "scaduta"), la data e l'ora di prenotazione, mentre per i biglietti si vogliono memorizzare il prezzo, la data e l'ora di acquisto e l'eventuale prenotazione di riferimento. In fase di acquisto presso le casse è possibile selezionare una delle riduzioni disponibili, le quali danno diritto ad uno sconto sul prezzo base.

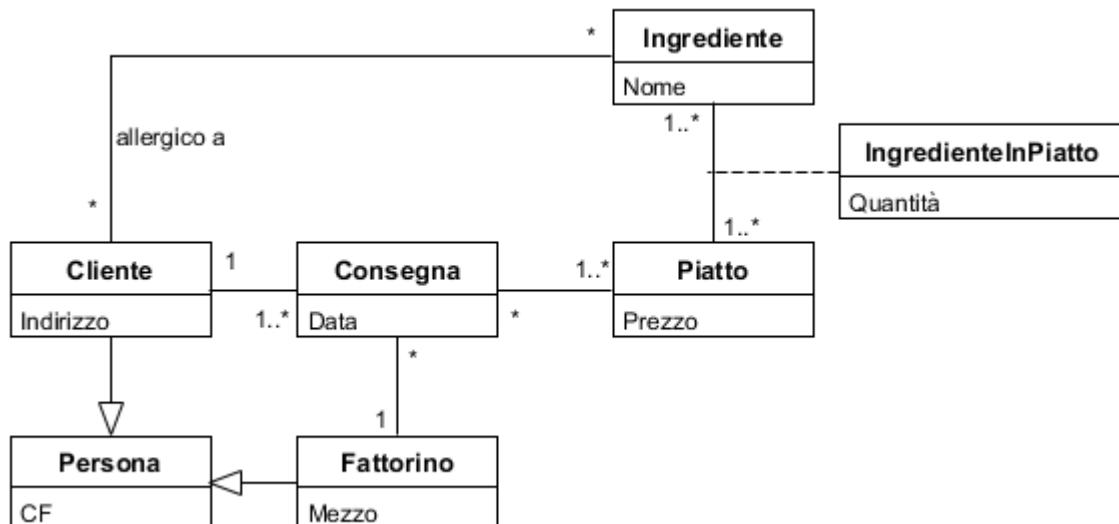
Si modellino le specifiche sopra riportate in UML attraverso:

1.1) un *diagramma dei casi d'uso* (3/32 punti)

1.2) un *diagramma delle classi* (9/32 punti)

2) Dato il diagramma delle classi in figura, che rappresenta le consegne effettuate dalla società "Food*", effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro (8/32 punti):

- Visualizzare tutti gli ingredienti a cui un dato cliente è allergico (per ogni consegna).
- Visualizzare l'elenco di tutte le consegne effettuate in un dato intervallo temporale (1 volta alla settimana);
- Visualizzare i piatti preferiti di ogni cliente (1 volta al mese)



3) Si spieghi in cosa consiste il *principio di modularità* nella progettazione del software, indicandone anche i benefici, le linee guida e quale impatto ha nella definizione dell'interfaccia (6/32 punti).

4) Si descrivano i modelli Multi-Window, Multi-Document e Multi-Paned per la progettazione di un'interfaccia grafica evidenziandone gli ambiti di applicazione (6/32 punti).

1) Si vuole modellare un sistema di monitoraggio dei calciatori per la Lega Calcio, utilizzato per raccogliere statistiche di movimento attraverso dispositivi GPS. Il sistema funziona per diverse competizioni (*Serie A*, *Serie B*, ecc.), ciascuna delle quali è composta da diverse partite; di ogni partita si conoscono la data e l'ora, il luogo, l'eventuale esito (inserito dagli amministratori del sistema al termine della partita) e le due squadre che si affrontano (una delle quali gioca “in casa”, l'altra “in trasferta”). Di ogni calciatore si conoscono nome, cognome, età, altezza, nazionalità e squadra di appartenenza; per semplicità, si assume che un calciatore non possa cambiare squadra. A ciascun calciatore viene assegnato un dispositivo GPS (che può cambiare di partita in partita e di cui si conosce il modello), il quale raccoglie informazioni sulla posizione in campo del giocatore: con cadenza di un secondo, il dispositivo invia al sistema la latitudine e la longitudine della posizione, insieme ad un timestamp e ad un indice di accuratezza del segnale GPS; i modelli più recenti inviano anche il conteggio dei passi rilevati. Al termine della partita, gli amministratori (previa login) lanciano delle elaborazioni statistiche sui dati raccolti per ottenere un dato riassuntivo per ciascun giocatore, il quale contiene il numero di chilometri percorsi, il tempo trascorso correndo e la zona del campo principalmente coperta. Le credenziali degli amministratori sono memorizzate nel sistema; un amministratore copre le partite di una sola competizione, ma una competizione è gestita da almeno due amministratori.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con un diagramma a stati il funzionamento dell'applicazione B00m!, gioco alla moda per smartphone in cui il giocatore deve disinnescare un ordigno senza farlo esplodere. All'avvio della partita l'ordigno è innescato. Attenzione: quando l'ordigno è innescato lo scadere del timeout e le vibrazioni dello smartphone lo fanno esplodere! Il giocatore deve indovinare una password di 3 caratteri per disinnescare l'ordigno. A ogni inserimento di un carattere corretto il gioco emette un beep ed è possibile procedere con l'inserimento del carattere successivo. Se l'inserimento è errato, i caratteri inseriti sono resettati e il giocatore deve ricominciare dall'inserimento del primo carattere (ma la password non cambia). Se il terzo carattere inserito è corretto allora il timeout si ferma e l'ordigno è disinnescato. Una volta disinnescato l'ordigno, l'utente ha 5 secondi per neutralizzarlo. Se l'ordigno è neutralizzato la partita termina con vittoria. In caso contrario, allo scadere dei 5 secondi, l'ordigno entra nuovamente in stato di innescos, il timeout è ripristinato e la password è resettata. Il giocatore deve quindi ricominciare con l'inserimento della nuova password. In ogni momento il gioco può essere sospeso e ripreso nell'esatto punto in cui l'utente l'ha sospeso (8/32 punti).

3) Si descrivano brevemente i *meccanismi di estendibilità* di UML (6/32 punti).

4) Si illustrino i tre criteri principali per il *testing in the small* commentandone l'efficacia (6/32 punti).

1) Si vuole modellare tramite diagramma delle classi il funzionamento di HOST, un famoso gioco multiplayer. Per prendere parte al gioco, ogni giocatore si autentica con username e password. Effettuato l'accesso, al giocatore appare una schermata di eroi di cui si conoscono la classe (guerriero, specialista, etc.) e l'universo di appartenenza. Il gioco è organizzato in match a cui prendono parte 10 eroi. Un giocatore può prendere parte ad un match con un singolo eroe. Tra tutti gli eroi disponibili, il giocatore seleziona un eroe in suo possesso (alcuni eroi sono disponibili a tutti, altri devono essere comprati) per prendere parte al match (di cui si conoscono la durata e la mappa in cui si svolge il match). Ad ogni match l'utente assegna due tipi di elementi costume al proprio eroe (selezionabili tra quelli disponibili per il proprio eroe): la cavalcatura e la voce. Di tali elementi si conosce il nome, il tema (halloween, natalizio, estivo, ...) e il costo. In particolare, una cavalcatura può essere assegnata a più eroi e una voce può essere assegnata a un solo eroe. Al termine di ogni match, per ogni eroe si registrano e visualizzano i punti esperienza ricevuti, l'esito del match (vittoria/sconfitta) e optionalmente un award.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con un diagramma di attività la casistica in cui un hotel vada in overbooking sul sito *prenotando.com*. La procedura viene avviata dal manager dell'hotel, il quale richiede allo staff del sito di cancellare una determinata prenotazione a causa di overbooking. Ricevuta la richiesta, lo staff contatta lo sfortunato cliente proponendogli una soluzione alternativa. Il cliente, una volta presa visione dell'offerta, può decidere se accettarla o meno. In caso di rifiuto, lo staff del sito cerca un'alternativa e sottopone al cliente una nuova proposta; il processo è iterativo e si interrompe quando il cliente accetta una proposta o quando lo staff non riesce a trovare nuove alternative. In quest'ultimo caso, la procedura viene conclusa dallo staff, il quale comunica al manager dell'hotel di non essere riuscito a cancellare la prenotazione (sarà poi cura del manager cercare un modo di risolvere la situazione). Qualora il cliente accetti una proposta alternativa, lo staff cancella la prenotazione e, allo stesso tempo, il manager provvede a rimborsare il cliente di quanto già pagato. Una volta effettuato il soggiorno presso la struttura alternativa, il cliente invierà infine la fattura del soggiorno allo staff del sito; in caso la spesa risulti maggiore rispetto a quella inizialmente prevista, lo staff richiedere al manager dell'hotel di rimborsare al cliente anche le spese in eccesso (8/32 punti).

3) Illustrare, con l'aiuto di un esempio, il concetto di *associazione qualificata* in UML (6/32 punti).

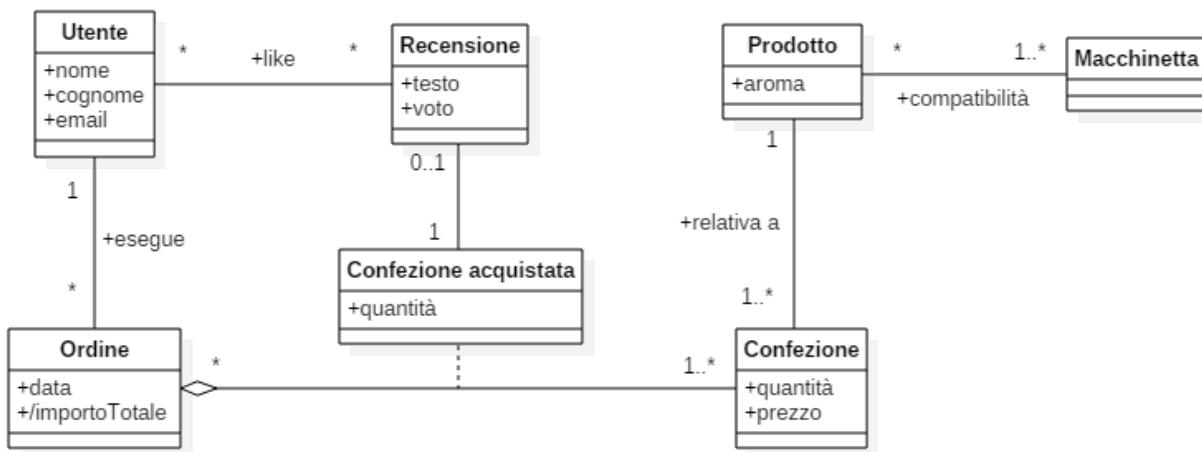
4) Si illustri il *principio di encapsulamento* e se ne descrivano i vantaggi (6/32 punti).

1) ChittyChat è un'applicazione che consente agli utenti di scambiare messaggi e pubblicare storie online. I messaggi possono essere scambiati tra singoli utenti o all'interno di gruppi di utenti. Di ogni utente si registrano l'username, il numero di telefono, la foto profilo e la descrizione. Di ogni gruppo si registrano il nome e la descrizione. Un utente può creare e partecipare ad un numero arbitrario di gruppi di cui può essere amministratore. Quando un utente crea un gruppo ne diventa amministratore. L'amministratore del gruppo può aggiungere a sua volta altri amministratori tra gli utenti del gruppo. Ogni utente può inviare messaggi a un singolo utente o all'interno di un gruppo. Dei messaggi inviati si conosce il testo, il timestamp (quando il messaggio è stato inviato), e un allegato multimediale opzionale. Un messaggio può essere scritto in risposta ad un messaggio precedente. I messaggi inviati all'interno di gruppi possono essere annotati come importanti. ChittyChat assegna ad ogni coppia di utenti singoli che scambia messaggi un token segreto utilizzato per criptare i messaggi scambiati (ciò non accade per i messaggi inviati all'interno di gruppi). Infine, ogni utente può eseguire l'upload di storie di cui si registrano l'allegato multimediale, la durata, e opzionalmente il testo e le menzioni di altri utenti.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Dato il diagramma delle classi in figura, il quale rappresenta la gestione degli ordini da parte di un rivenditore di caffè, effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro:



- Visualizzare gli ultimi prodotti acquistati da un determinato utente (più volte al giorno).
- Visualizzare la media dei voti ricevuti da un dato prodotto (1 volta alla settimana).
- Visualizzare i prodotti che hanno ricevuto meno ordini (1 volta al mese).

3) Nell'ambito della progettazione delle interfacce utente, si descrivono brevemente gli aspetti che riguardano le tre *strutture di riferimento*, i *project standard* ed i criteri relativi alla *comunicazione visiva* (6/32 punti).

4) Si illustrino le caratteristiche dei *modelli agili* e si descrivano brevemente le attività strutturali dell'Extreme programming (6/32 punti).

1) Il sito marry.me consente agli utenti registrati di programmare matrimoni. Ogni utente può creare uno o più eventi (contraddistinti da una data) ed eventualmente associarli all'utente del partner. I due utenti degli sposi possono quindi creare la propria lista di invitati, indicando l'eventuale conferma e con la possibilità di indicare un'email per inviare un promemoria all'invitato. Gli invitati possono essere organizzati in tavoli (indicando il nome di questi ultimi) e possono essere associati ad un menù, il quale ha una descrizione, una tipologia (es: "normale", "celiaco", "artista") ed un prezzo per persona. Il sito mette inoltre a disposizione dei servizi (associati ad un utente fornitore) che gli sposi possono cercare ed acquistare; per ogni servizio acquistato si memorizzano l'importo speso ed il punteggio (da 1 a 5) eventualmente assegnato a matrimonio concluso. I servizi hanno un nome, una descrizione ed un'eventuale "stella" che evidenzia i servizi più meritevoli; la stella viene assegnata in base a due criteri (un numero minimo di punteggi assegnati ed un minimo punteggio medio) che variano in base alla tipologia di servizio. Tra le tipologie disponibili sono presenti i fioristi (di cui si memorizza l'orario di apertura), i fotografi, le location (di cui si memorizza la tariffa oraria) ed il catering. I menù possono essere associati ad un catering o ad una location.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con diagramma degli stati il funzionamento di una macchinetta per l'erogazione di caffè dotata di touch screen e prodotta dall'azienda Kaffee. Ad accensione avvenuta, la macchinetta entra in attesa e suona una leggiadra melodia classica. Quando un utente preme il touch screen, la macchinetta interrompe la musica ed entra in selezione. In questa fase, all'utente è mostrata la lista dei caffè disponibili. Tra i possibili tipi di caffè selezionabili, il "coding coffee" è un particolare caffè che non richiede l'aggiunta di zucchero. Se il caffè selezionato non è di tipo coding coffee, all'utente è mostrata la schermata di selezione zucchero. Una volta scelta la quantità di zucchero, la macchinetta visualizza l'importo del caffè e attende che l'utente completi il pagamento. Se l'utente seleziona il coding coffee, la macchinetta salta la selezione dello zucchero, visualizza l'importo del caffè e attende che l'utente inserisca l'importo. Se l'importo inserito è maggiore di quello richiesto, la macchinetta restituisce il resto. Saldato l'importo, la fase di selezione si conclude e la macchinetta eroga il caffè. Durante l'erogazione la macchinetta mostra lo stato dell'erogazione. Al termine dell'erogazione, la macchinetta emette un beep e ritorna in stato di attesa. La fase di selezione è gestita con un timer, se l'utente non completa la selezione prima dello scadere del timer, la macchinetta ritorna in stato di attesa. Infine, la macchinetta è spesso soggetta a guasti. In qualsiasi momento delle due fasi la macchinetta può auto-resettarsi. In tal caso la macchinetta entra in stato di guasto, mostra un messaggio di errore, e attende che un operatore dell'azienda Kaffee esegua la manutenzione. A manutenzione avvenuta, la macchinetta torna in stato di attesa. (8/32 punti).

3) Si indichino i criteri generali dei *metodi d'analisi* e si illustrino le differenze tra i diversi metodi (6/32 punti).

4) Si illustri il concetto di *polimorfismo* nel paradigma a oggetti, anche attraverso un esempio (6/32 punti).

1) Si vuole modellare la banca dati che gestisce gli articoli pubblicati da ricercatori, al fine di automatizzare la gestione della bibliografia (cioè l'elenco delle pubblicazioni per ricercatore) e il calcolo dell'indice di produttività dei singoli ricercatori (indice basato sul numero di citazioni per articolo). Ogni articolo è descritto da un titolo, da una lista di parole chiave e da un sommario, cita almeno un altro articolo e può essere citato da altri articoli. Un articolo è scritto da almeno un autore di cui si conoscono email e istituto di affiliazione. Per ogni articolo si memorizzano l'ordine degli autori (primo, secondo, terzo, etc.) e un flag che rappresenta il "corresponding author", ossia l'autore cui indirizzare eventuale corrispondenza relativa all'articolo (ci possono essere più corresponding author per ogni articolo). Ogni articolo è pubblicato da un ente di pubblicazione (di cui si conoscono il nome e l'editor) che può essere di tipo "Rivista" o "Conferenza"; in quest'ultimo caso, si conosce anche il luogo in cui la conferenza si svolge. Un articolo può essere pubblicato una sola volta e della pubblicazione si conoscono l'anno e il fattore di impatto (numerico) dell'ente di pubblicazione. Per poter essere pubblicato, un articolo deve essere valutato da 3 revisori (che sono a loro volta ricercatori). Nel caso cui l'ente di pubblicazione sia di tipo Conferenza, è obbligatorio per almeno uno degli autori dell'articolo partecipare alla conferenza. Tra gli autori partecipanti alla conferenza, si memorizza quello che presenterà oralmente l'articolo pubblicato. Infine, ogni anno l'ente di pubblicazione premia l'articolo (pubblicato dallo stesso ente di pubblicazione 10 anni prima) con il maggior numero di citazioni.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare tramite un diagramma di attività l'acquisto di prodotti tramite l'applicazione PO\$, un servizio di gestione di carte prepagate. L'acquisto inizia con la selezione del prodotto da parte del cliente. Una volta selezionato il prodotto, PO\$ attende l'inserimento del PIN dell'utente e, nel frattempo, mostra il messaggio "wait". Una volta inserito il PIN, PO\$ si interfaccia con la banca presso cui la carta è registrata, la quale verifica che il PIN inserito sia corretto. Se il PIN inserito non è corretto, PO\$ richiede nuovamente il PIN all'utente ripetendo la procedura di inserimento. L'utente ha 3 tentativi per inserire un PIN corretto. Al terzo inserimento errato, la banca blocca la carta, PO\$ segnala l'errore di "transazione fallita" all'utente e conclude il pagamento. Se il PIN inserito è corretto, la banca verifica il credito disponibile sulla carta. Se il credito non è sufficiente, PO\$ segnala l'errore di "transazione fallita" all'utente e conclude il pagamento. Se il credito è sufficiente, la banca aggiorna il credito residuo sulla carta, e, infine, PO\$ notifica all'utente la conclusione del pagamento.

3) Si elenchino e si descrivano brevemente i sei criteri di *usabilità di un'interfaccia utente* (6/32 punti).

4) Si illustri il principio di progettazione *modularità* commentandone i benefici sullo sviluppo del software (6/32 punti).

1) Si vuole modellare la banca dati che gestisce l'assegnazione dei premi Oscar ad attori, registi e film, con l'obiettivo di monitorare, per ogni attore/regista, la carriera (in termini del numero di Oscar vinti) e la produttività (numero di film), a livello sia annuale sia complessivo. La cerimonia di assegnazione degli Oscar si tiene ogni anno in una location diversa. Ogni cerimonia ha un presentatore, che può anche non essere un attore o un regista. Di ogni attore e regista si registrano nome e cognome (un attore può anche essere regista), di ogni film si conoscono titolo, genere e anno di produzione. Per ogni film si registrano il regista e tutti gli attori partecipanti con il rispettivo ruolo. Durante la cerimonia vengono assegnati in modo esclusivo diversi premi a un film (per esempio miglior film o migliore colonna sonora), a un attore (miglior attore protagonista e non protagonista) o un regista (miglior regista). Attori e registi possono ricevere un solo riconoscimento per film, mentre un film può riceverne più di uno.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con un diagramma degli stati la pubblicazione di articoli presso riviste scientifiche. La procedura inizia con l'invio dell'articolo alla rivista scientifica da parte degli autori. Quando l'articolo è ricevuto dalla rivista, esso entra in fase di ammissione. In questa fase l'editor della rivista valuta preliminarmente l'articolo. Se l'articolo non rispetta il *topic* della rivista esso viene rifiutato e la procedura si conclude. Se l'articolo rispetta il topic, l'editor lo assegna a 3 revisori e l'articolo entra in fase di revisione. Ogni revisore assegna un voto all'articolo (da -3 a +3). Quando i revisori concludono la revisione, il sistema calcola il voto medio. Se tutti gli autori hanno assegnato il voto massimo (cioè 3) all'articolo, l'articolo è accettato ed entra in fase di *camera ready*, la quale termina quando la rivista viene pubblicata. Se il voto medio è insufficiente (cioè < 0), l'articolo viene rifiutato e la procedura si conclude. Altrimenti, l'esito delle revisioni è inviato agli autori i quali hanno 2 settimane per preparare una nuova versione dell'articolo e inviarla alla rivista. Se la nuova versione non viene inviata entro 2 settimane, l'articolo viene rifiutato e la procedura si conclude. Quando la nuova versione dell'articolo è ricevuta dalla rivista, essa entra direttamente in fase di revisione e la procedura riprende come descritto in precedenza per un massimo di 2 volte. Alla terza mancata accettazione l'articolo viene automaticamente rifiutato (8/32 punti).

3) Si descriva (riportando anche un esempio) il concetto di polimorfismo, descrivendone i vantaggi in abbinamento all'*istanziamento dinamico* o *late binding* (6/32 punti).

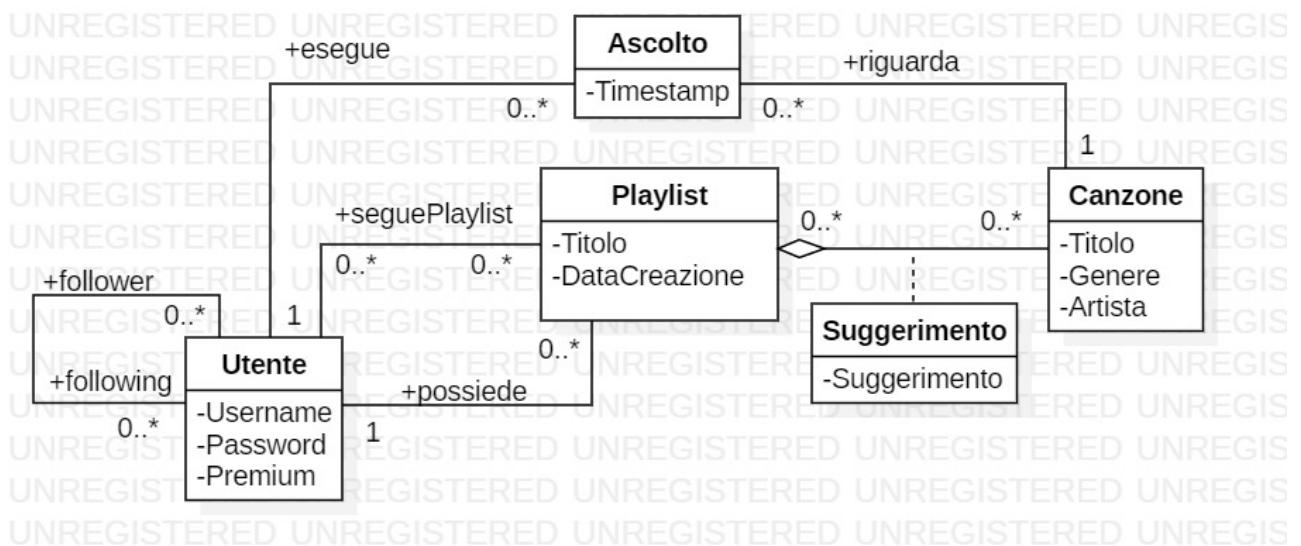
4) Si illustrino i tre criteri principali per il *testing in the small* commentandone l'efficacia (6/32 punti).

1) Si vuole modellare un'applicazione di *car sharing*. Al momento della registrazione al servizio, un utente può iscriversi come “passeggero” e/o “guidatore”. Di ogni utente si conosce il codice fiscale. La registrazione di un guidatore richiede l'inserimento del numero della patente e delle macchine interessate (per semplicità si assume che ogni macchina sia guidata da un solo utente). Di ogni macchina si registrano targa e capienza massima. Per avviare lo sharing, un guidatore inserisce nell'applicazione il viaggio che intende effettuare in una certa data. Il viaggio corrisponde a una sequenza di tratte tra due città diverse. I passeggeri interessati contattano il guidatore per prenotare un posto in macchina su una o più tratte consecutive. Il costo della singola tratta è stimato in base ai km tra le due città. Al termine del viaggio, ogni utente della macchina assegna un voto (da 1 a 5) agli altri utenti e il sistema memorizza il voto medio dei passeggeri e del guidatore. Nel caso in cui il voto medio del passeggero sia inferiore a 2 o quello del guidatore sia inferiore a 3, l'applicazione sospende l'utente dal servizio. Per semplicità si assumano i problemi di capienza risolti automaticamente dall'applicazione.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Dato il diagramma delle classi in figura, che rappresenta gli ascolti di musica online effettuati sulla piattaforma “Spoti5”, effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro (8/32 punti):



- Visualizzazione dei dettagli dei brani ascoltati: 1.000.000 al giorno
- Elenco delle playlist seguite da un utente: 1.000.000 a settimana
- Visualizzazione di “following” tra utenti: 500.000 al mese

3) Si descrivano i diversi modelli di riferimento per la progettazione di un'interfaccia, evidenziandone le differenze (6/32 punti).

4) Si descriva brevemente il metodo di conteggio dei function point spiegandone anche il ruolo e l'utilità (6/32 punti).

1) Il sito verdepesto.it permette agli utenti registrati di inserire ricette, valutarle e commentarle. In fase di registrazione, gli utenti (alcuni dei quali possono svolgere il ruolo di moderatore) devono inserire nome, cognome, username e password. Il sito contiene un'anagrafica di ricette standard (identificate da un nome) di cui gli utenti possono proporre la propria versione. Ogni versione inserita deve riportare un nome ed un'indicazione della durata e del livello di difficoltà, oltre all'elenco degli ingredienti utilizzati; di ogni ingrediente va indicata la quantità e l'unità di misura (in grammi o in unità). Una volta inserita sul sito, la versione di una ricetta può essere valutata e commentata dagli altri utenti. In particolare, un utente può esprimere un giudizio (con un voto da 1 a 10; non si può esprimere più di un giudizio per versione) o lasciare dei commenti sulle versioni. I commenti possono essere effettuati direttamente sulla versione della ricetta, oppure possono essere lasciati in risposta ad un altro commento. I moderatori possono intervenire sui commenti pubblicati modificandone il testo per censurare URL e linguaggio non consono; in tal caso, il sistema tiene traccia sia del testo originale che di quello moderato.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con diagramma degli stati il ciclo di vita della richiesta di un visto australiano. La richiesta di visto si basa sul completamento e sulla validazione di una form con i dati dell'utente richiedente. Quando un utente inizia una richiesta di visto, questa è identificata come incompleta. Una volta completata la form, la richiesta entra in stato di pronto. Quando l'utente ne conferma l'invio, la richiesta viene inviata al dipartimento degli affari interni, previo pagamento di una tassa governativa prevista solo per alcune tipologie di visto (ossia quelle soggette a clausola "BPAY"). In particolare, se la richiesta è soggetta a clausola "BPAY" essa rimane in attesa di pagamento fino a quando l'utente non salda il costo della pratica. Quando la richiesta è presa in carico dal dipartimento degli affari interni, essa è identificata come ricevuta. Al termine di questa fase, la richiesta può essere finalizzata (la decisione di concessione/rifiuto è notificata via e-mail all'utente) o informazioni aggiuntive possono essere richieste all'utente. In questa evenienza, una volta forniti i dati richiesti questi devono essere validati. Se le informazioni fornite sono sufficienti, la richiesta è finalizzata e l'esito è comunicato all'utente, altrimenti ulteriori informazioni sono richieste. (8/32 punti)

3) Si descrivano brevemente le caratteristiche dei diversi tipi di *manutenzione del software* (6/32 punti).

4) Si illustrino le caratteristiche comuni ai *modelli prescrittivi* di produzione del software; si citi poi almeno un esempio di modello prescrittivo e uno di modello non prescrittivo (6/32 punti).

1) MyFutsal è un sito web che facilita la gestione di una squadra di calcetto amatoriale. Dopo essersi registrati (indicando nome, cognome, username, password), gli utenti possono creare una o più squadre da gestire. Una volta creata una squadra (indicandone il nome), l'utente creatore viene considerato come allenatore della squadra; altri utenti possono essere aggiunti col ruolo di allenatore. Ad una squadra possono essere aggiunti fino a 20 giocatori (anch'essi utenti), e un utente può essere aggiunto a tante squadre. All'interno di una squadra, ogni giocatore ha un numero di maglia preferito. Gli allenatori possono creare eventi di tipo *partitella* o *campionato*; per ogni evento è necessario indicare (oltre al tipo) la data ed il luogo di svolgimento (quest'ultimo comprensivo di indirizzo e numero di telefono) ed il numero di giocatori che si prevede di convocare. Il sito indica il numero minimo e massimo di giocatori che si possono convocare per ogni tipologia di evento. In caso di partita di tipo *campionato* va indicata la squadra avversaria, anche se quest'ultima non è presente in MyFutsal. In fase di creazione dell'evento, gli allenatori possono decidere se inviare anche una notifica ai giocatori della squadra. Attraverso il sito, i giocatori possono consultare i prossimi eventi ed indicare la propria disponibilità a parteciparvi. In base alle disponibilità fornite, gli allenatori decidono quale dei giocatori disponibili sono convocati; per ogni disponibilità indicata si vogliono quindi memorizzare l'eventuale convocazione e, di conseguenza, l'eventuale numero di gol segnati dal giocatore.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con diagramma degli stati il funzionamento di una centralina di controllo in un'automobile con doppia alimentazione benzina-gas. La centralina ha il compito di decidere quale dei due tipi di alimentazione deve essere utilizzato dal motore; inoltre, essa dispone di un led per segnalare il suo stato e di un pulsante che consente al guidatore di cambiare manualmente l'erogazione. La centralina si avvia con l'avvio del motore ed entra in una fase di pre-riscaldamento, in cui l'alimentazione è inizialmente a benzina e il led viene mostrato giallo lampeggiante. Non appena il motore raggiunge una temperatura sufficiente, la centralina passa all'erogazione del gas, durante la quale il led viene mostrato verde e fisso. Durante l'erogazione del gas, la centralina controlla ogni 5 secondi il livello di gas residuo e, quando questo si esaurisce, passa automaticamente all'erogazione di benzina. In alternativa, il guidatore può commutare manualmente all'erogazione di benzina premendo il pulsante. In entrambi i casi, viene mostrato un led giallo fisso ad indicare che la centralina non tenterà più di passare all'alimentazione a gas: a quel punto, solo il guidatore può fare la commutazione premendo nuovamente il pulsante; in tal caso, la centralina può decidere di tornare alla fase di pre-riscaldamento qualora la temperatura del motore non fosse ancora sufficiente. Se il pulsante viene premuto nella fase di pre-riscaldamento, la centralina passa all'erogazione fissa di benzina. La centralina si spegne con lo spegnersi del motore; se la benzina dovesse esaurirsi in un qualunque momento, la centralina forza lo spegnimento del motore.

3) Si descriva in cosa consiste il *principio di modularità* nella progettazione del software, indicandone anche i benifici, le linee guida e quale impatto ha nella definizione dell'interfaccia (6/32 punti).

4) Si illustrino le caratteristiche dei *modelli agili* e si descrivano brevemente le attività strutturali dell'Extreme programming (6/32 punti).

1) Si vuole modellare il sistema informativo di un'agenzia interinale che seleziona candidati per posti di lavoro su richiesta di aziende terze. Le aziende che vogliono aprire delle posizioni ne comunicano i dettagli all'agenzia (titolo della posizione, descrizione e RAL - reddito annuo lordo); una posizione si intende aperta per una sola persona. L'agenzia mantiene l'anagrafica delle persone che si candidano per una posizione e delle candidature sottomesse (comprese di CV e di una lettera di presentazione opzionale); una persona può candidarsi a più posizioni, ma un'unica volta per la stessa posizione. L'agenzia assegna un recruiter di riferimento per ciascuna azienda; il recruiter si occupa di valutare le candidature ricevute per una determinata posizione, assegnando un punteggio da 1 a 100. In base ai punteggi assegnati, il recruiter decide l'organizzazione di uno o più colloqui. I colloqui possono essere interni o esterni; nel primo caso vi partecipa un recruiter dell'agenzia (che non corrisponde necessariamente a quello che ha selezionato le candidature), nel secondo caso vi partecipa un responsabile indicato dall'azienda che offre il posto di lavoro. Di ogni colloquio si vogliono memorizzare la data di svolgimento ed il punteggio assegnato, anch'esso da 1 a 100. Al termine del percorso di selezione si vuole tenere conto della candidatura eventualmente selezionata per il posto di lavoro offerto. Si noti che il sistema è a uso esclusivo dell'agenzia (non è prevista un'interfaccia web a cui i candidati abbiano accesso).

Si modellino le specifiche sopra riportate in UML attraverso:

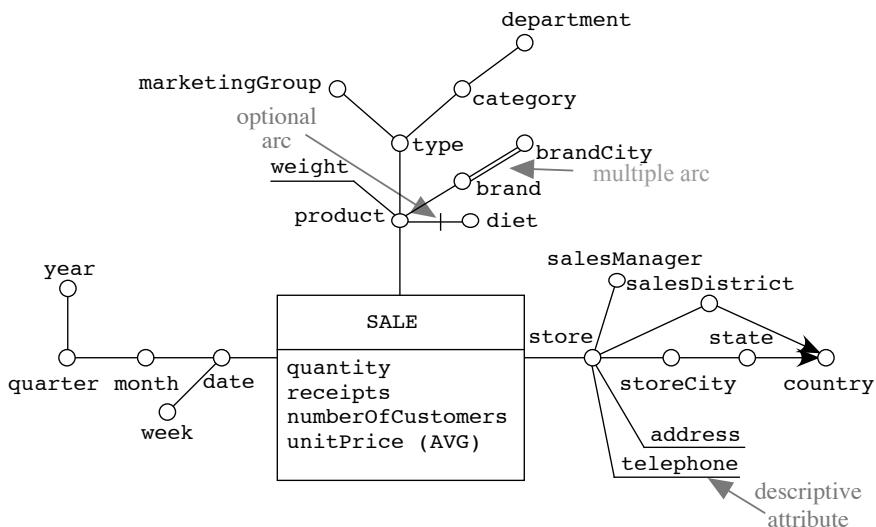
- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)

2) Si vuole modellare con un *diagramma di sequenza* il funzionamento di NLI2DB, un sistema in grado di tradurre una stringa di testo (espressa in linguaggio naturale) in una query da eseguire su un database. L'interazione viene iniziata dal client, il quale invia una stringa di testo al server. Il server analizza la stringa per capire se si tratta di una richiesta di tipo *describe* o di tipo *query*. Nel primo caso, la stringa contiene semplicemente il nome di un attributo di una certa tabella di cui l'utente vuole conoscere i valori più utilizzati; il server esegue quindi una query sul database per recuperare i valori più utilizzati e li restituisce al client. Nel secondo caso, la stringa deve essere interpretata dal sistema per tradurre la richiesta da linguaggio naturale a query SQL. L'interpretazione del testo può essere ostacolata dalla presenza di alcune ambiguità (ad esempio, l'utente può menzionare un attributo "città" di cui esistono due varianti: "città di nascita" e "città di residenza"); in questo caso, il server interroga il client per risolvere le ambiguità e prova nuovamente ad interpretare il testo considerando anche le informazioni aggiuntive ricevute. In ogni caso, non è scontato che l'interpretazione riesca a generare una query SQL valida; in tal caso, viene restituito un errore al client. Se la query risulta invece valida, questa viene eseguita sul database ed i risultati vengono inoltrati al client. Infine, il server salva (in maniera asincrona) l'SQL della query generata in una tabella di log del database (8/32 punti).

3) Si illustri il ruolo dello *studio di fattibilità* all'interno del ciclo di vita del software, precisandone obiettivi e risultati (6/32 punti).

4) Si spieghi cos'è un *package* e si illustrino struttura e utilizzo dei *diagrammi dei package* (6/32 punti).

1) Si vuole meta-modellare con un diagramma delle classi una versione semplificata del Dimensional Fact Model (DFM), un formalismo grafico per la modellazione concettuale di Data Warehouse. Il DFM si sviluppa attorno al concetto di fatto, ovvero un fenomeno di business (ad esempio le vendite di prodotti, *SALE* in figura). Ogni fatto include un insieme (anche vuoto) di misure (ad esempio la quantità di prodotti venduta, *quantity*, e l'incasso, *receipts*). A ciascuna misura è associato almeno un operatore di aggregazione (ad esempio somma o media); una misura si dice additiva se è associata alla somma, non additiva in caso contrario. Un fatto è caratterizzato da due o più gerarchie, ciascuna strutturata come un grafo di attributi (mostrati in figura come cerchietti). Tra gli archi di questi grafi, alcuni possono essere multipli e/o opzionali (v. figura). Tra gli attributi, alcuni possono essere descrittivi (per esempio *address*). In ogni gerarchia, uno degli attributi è graficamente attaccato al fatto ed è detto dimensione (*store*, *product* e *date* in figura) (11/32 punti).



2) Si modelli con diagramma degli stati il funzionamento di COOL, un'applicazione a supporto della traduzione in SQL di query in linguaggio naturale. All'avvio di COOL, il sistema attende una prima query dell'utente. Una volta ricevuta la query, viene avviata l'interpretazione della query, che si compone di due fasi: parsing e verifica della query. Se l'interpretazione va a buon fine, COOL restituisce il risultato della query all'utente e si mette in attesa di nuove query. Altrimenti, nel caso in cui il parsing fallisca, il sistema ritorna in fase di attesa e l'errore è segnalato all'utente. Se in fase di verifica COOL riscontra l'esistenza di ambiguità, il sistema entra in fase di disambiguazione. Per ogni ambiguità, il sistema pone una domanda all'utente e ne attende la risposta. Una volta esaurite le domande, il sistema conclude l'interpretazione. Se l'utente non risponde alla domanda entro due minuti, il sistema ritorna automaticamente in fase di attesa segnalando un errore di "timeout". In ogni momento il comando reset ripristina COOL in stato di attesa di una nuova query (9/32 punti).

3) Si spieghi la differenza tra testing in the small e testing in the large, portando anche alcuni esempi (6/32 punti).

4) Si illustrino i benefici dell'approccio orientato agli oggetti nella progettazione del software, citando i problemi presenti nell'approccio funzionale (6/32 punti).

- 1)** Si vuole modellare con un diagramma delle classi il sistema informativo di un’azienda che si occupa di compilare le dichiarazioni dei redditi. Ogni dichiarazione è effettuata da una o più persone (in caso di coniuge e/o figli a carico), viene gestita da un addetto dell’azienda e firmata da un responsabile dell’azienda. Le dichiarazioni possono essere effettuate anche dai dipendenti dell’azienda; in ogni caso, viene associato un ID a tutti coloro che effettuano dichiarazioni. Di ogni dichiarazione si memorizza l’anno, il costo per la compilazione e l’eventuale importo del conguaglio con l’Agenzia delle Entrate (qualora la persona che ha effettuato la dichiarazione sia tenuta a fare un versamento aggiuntivo di tasse). Ogni dichiarazione è compilata sulla base di una serie di documenti, relativi a tante tipologie diverse (tra cui i *redditi*, le *spese mediche*, le *spese per ristrutturazioni*, ecc.); il sistema informativo memorizza anche il numero massimo di documenti presentabili, che (in base alla normativa vigente) dipende dalla relativa tipologia (*11/32 punti*).
- 2)** Si modelli con diagramma di attività il processo di compilazione di una dichiarazione dei redditi nel contesto dell’esercizio 1. L’addetto apre la pratica in seguito alla richiesta del cliente, ma quest’ultimo può consegnare i documenti solamente a partire dal 1 aprile dell’anno corrente. Alla ricezione, l’addetto verifica la correttezza e la completezza dei documenti e chiede eventuali integrazioni al cliente. La compilazione della dichiarazione avviene una volta terminata la verifica; dopodiché, la dichiarazione viene controllata e firmata dal responsabile, mentre il cliente si occupa del pagamento. Al termine di entrambe le attività, l’addetto si occupa di inviare la dichiarazione all’Agenzia delle Entrate (*9/32 punti*).
- 3)** Si illustrino le differenze tra *analisi orientata agli oggetti, agli stati e alle funzioni*, facendo anche esempi di diagrammi UML che ricadono in ciascuna categoria (*6/32 punti*).
- 4)** Si descriva brevemente il metodo *function points* spiegandone il ruolo e l’utilità (*6/32 punti*).

- 1)** Si vuole modellare con un *diagramma delle classi* il sistema informativo che supporta un'app per i trasferimenti di denaro tra utenti. Per utilizzare l'app è necessario iscriversi fornendo le proprie generalità (nome, cognome, codice fiscale) ed un IBAN di riferimento; l'utente deve anche decidere l'importo del proprio portafoglio digitale, che verrà prelevato tramite l'IBAN fornito in fase di registrazione. I trasferimenti di denaro possono essere effettuati da un utente ad un altro utente; di ogni trasferimento si registrano l'importo, la data e l'ora (l'applicazione impedisce il trasferimento in caso il saldo del portafoglio digitale del mittente non sia superiore o uguale all'importo che si vuole trasferire). Ogni trasferimento può essere associato a dei commenti; un commento può essere associato direttamente ad un trasferimento o ad un altro commento. Per incentivare la diffusione dell'app vengono definite delle campagne pubblicitarie che assegnano dei premi in denaro per chi invita dei nuovi utenti. L'importo del premio può variare a seconda della campagna, la quale ha una durata specifica. Si consideri che un utente non può invitare più di 5 nuovi utenti nel corso della stessa campagna (*10/32 punti*).
- 2)** Si modelli con un *diagramma di sequenza* il processo di trasferimento di denaro da un utente (mittente) ad un altro (destinatario) nel contesto delle specifiche dell'esercizio 1. Quando il mittente esegue la richiesta di trasferimento, il sistema prende in carico la richiesta e verifica che il saldo del portafoglio digitale sia sufficiente. Nel caso non lo sia, viene chiesto all'utente se vuole prelevare l'importo mancante dal proprio IBAN. Se l'utente si rifiuta, il sistema annulla il trasferimento; altrimenti, il sistema invia alla banca del mittente la richiesta di prelievo (si assume per semplicità che il prelievo non possa essere negato). Quando il saldo del portafoglio è sufficiente, il sistema invia il denaro al destinatario in maniera asincrona e chiude il trasferimento (*10/32 punti*).
- 3)** Si elenchino e si discutano brevemente 3 *qualità del software* (*6/32 punti*).
- 4)** Si illustri il principio di *modularità* commentandone i benefici sullo sviluppo del software (*6/32 punti*).

1) Si vuole modellare con un *diagramma delle classi* l'app Wazzup, tramite cui gli utenti possono scambiarsi messaggi. Di ogni utente si conosce username e data di nascita. Un utente può prendere parte a gruppi e può inviare messaggi a utenti o gruppi. Ogni gruppo è identificato da un nome e composto da almeno due utenti. Un messaggio può essere inviato a un gruppo o a un utente. Ogni messaggio contiene l'ora d'invio, un testo e/o un'immagine. Ogni utente può marcare il proprio messaggio come "ristretto"; i messaggi ristretti non sono mostrati a utenti minorenni. Ogni utente può modificare i propri messaggi e si vuole tenere traccia di tutte le modifiche; il messaggio modificato fa riferimento a quello originale e della modifica si registra l'orario (*11/32 punti*).

2) Si vuole modellare l'invio del messaggio sull'applicazione Wazzup. Quando l'utente inizia a digitare, il messaggio entra in fase di scrittura, durante la quale si mostra l'anteprima del messaggio. Quando l'utente invia il messaggio, il processo termina se il messaggio è vuoto (non c'è nessun messaggio da inviare). Altrimenti, il messaggio entra in stato di invio e Wazzup mostra il popup di caricamento. Se l'invio procede senza problemi il processo termina e il messaggio viene cancellato. In caso di errore (ad esempio in assenza di rete) Wazzup prova a re-inviare il messaggio fino a un massimo di 3 tentativi. Prima di effettuare nuovamente l'invio, il sistema mostra un popup di attesa e dopo 2 secondi prova a rinviare il messaggio. Se il messaggio non è inviato correttamente dopo 3 tentativi, Wazzup mostra un popup di errore, attende che l'utente ne confermi la presa visione e cancella il messaggio. In ogni momento l'utente può terminare il processo cancellando il messaggio.

Si disegni il *diagramma degli stati* della classe messaggio (*9/32 punti*).

3) Si elenchino e si discutano brevemente tre *qualità del software* (*6/32 punti*).

4) Si spieghi la differenza tra un software *corretto* e un software *robusto* (*6/32 punti*).

1) MyFitTrain è un sito web che offre la possibilità di svolgere attività fisica online. Gli utenti della palestra online (iscritti con username e password) possono essere istruttori o allievi. Ciascun allievo fornisce al sistema informazioni riguardanti la sua forma fisica (come il peso e l'altezza) e ha la possibilità di iscriversi alle classi messe a disposizione dalla palestra. Le classi possono essere di tipologie diverse (ad esempio, "yoga", "cardio"), per ciascuna delle quali possono essere memorizzate informazioni diverse (ad esempio, per le classi "cardio" si memorizza l'attrezzatura necessaria). Per ogni tipologia di classe si vuole registrare il numero massimo e minimo di partecipanti. Il sistema registra le iscrizioni degli allievi alle classi, memorizzando la data di iscrizione. Ogni classe è composta da almeno una sessione e ha una durata che corrisponde alla somma delle durate delle sessioni che la formano; una sessione è svolta da almeno un istruttore. Infine, è necessario tener conto, per ciascun allievo, delle informazioni relative alla partecipazione a una sessione (ossia, le calorie bruciate e l'eventuale valutazione espressa dall'utente su quanto svolto).

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (12/32 punti).

2) Si modelli con un *diagramma di attività* il processo di richiesta di una black-box assicurativa per autoveicoli e la relativa installazione. Il cliente spedisce i documenti al referente assicurativo, il quale ne verifica la correttezza. L'addetto procede all'invio della black-box soltanto nel caso in cui i documenti siano corretti; in caso contrario, il processo termina. Alla ricezione della black-box, il cliente prosegue con l'installazione della stessa e con la configurazione dell'applicazione smart-phone. In particolare, la black-box deve essere posizionata nella parte alta del parabrezza, dopodiché bisogna aspettare dieci secondi che questa aderisca nella maniera adeguata. Nel mentre, il cliente può installare e configurare, tramite il numero di polizza, la relativa applicazione sul proprio smartphone. Al termine di tali attività il cliente può procedere ad accoppiare la black-box all'applicazione (8/32 punti).

3) Com'è classificata la *riusabilità* di un software?

- a. Esterna e di Processo
- b. Interna e di Prodotto
- c. Esterna e di Prodotto
- d. Interna e di Processo

4) Quali di questi diagrammi UML vengono normalmente utilizzati a supporto del *collaudo* del software, con particolare riferimento al "testing in the large"?

- a. Sequenza
- b. Attività
- c. Classi
- d. Casi d'uso

5) Quali di queste *tecniche di verifica* sono applicabili in un contesto black-box?

- a. Code walk-through
- b. Testing in the small
- c. Code inspection
- d. Testing in the large
- e. nessuna delle precedenti

6) Un software viene modificato per aumentare l'efficienza di una funzionalità di importazione massiva di dati. Di che tipo di *manutenzione* si tratta?

- a. Correttiva
- b. Evolutiva
- c. Perfettiva
- d. Adattiva

7) Quali di questi *modelli* sono specificamente pensati per situazioni progettuali in cui i requisiti utente sono incerti o mutevoli nel tempo?

- a. Modello MDA
- b. Modello RAD
- c. Modello a cascata
- d. Nessuno dei precedenti

8) Quali delle seguenti affermazioni, riguardanti le funzioni di tipo dato e tipo transazione, sono vere?

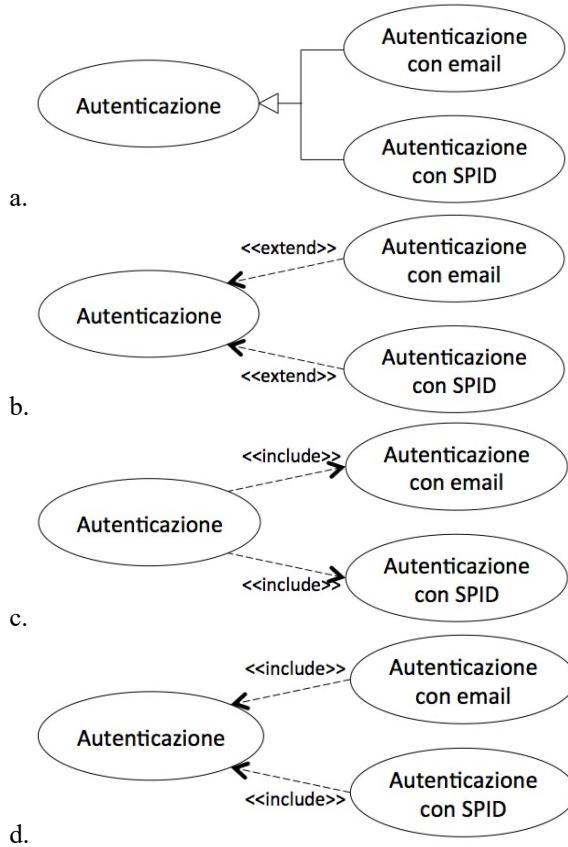
- a. La logica di un processo EQ (External Inquiry) non contiene formule matematiche o calcoli e non crea dati derivati

- b. Il compito primario di un EIF (External Interface File) è di contenere dati referenziati da uno o più processi elementari dell'applicazione che si sta contando
- c. Il compito principale di un EO (External Output) è di modificare il comportamento del sistema
- d. Nessuna delle precedenti

9) Le special GUI sono pensate per situazioni in cui...

- a. ...l'utente deve effettuare un data-entry massivo
- b. ...l'utente può avere scarsa esperienza con l'utilizzo dei PC
- c. ...la soddisfazione dell'utente nell'uso dell'interfaccia è importante
- d. ...memorabilità e apprendibilità dell'interfaccia non sono importanti
- e. nessuna delle precedenti

10) L'autenticazione a un portale si può effettuare tramite email oppure tramite lo SPID (Sistema Pubblico d'Identità Digitale). Scegliere tra le seguenti soluzioni di modellazione quella più corretta.



Compito di IS del 15/2/2021

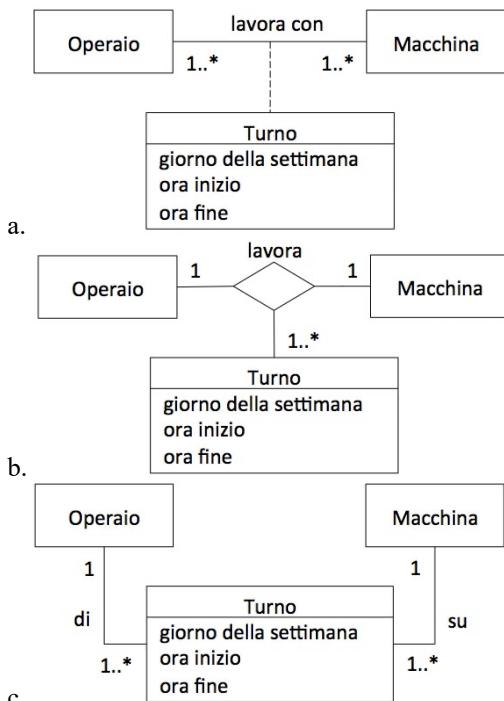
Tempo concesso: 60 minuti

1) Il VAB (Vigilanza Antincendi Boschivi) offre un portale per l'organizzazione di missioni a cui partecipano i propri membri. Un membro si registra come utente con username, password e, opzionalmente, può inserire la propria data di nascita. Ciascun utente fornisce al sistema tutti i certificati da lui acquisiti, con le rispettive informazioni: identificativo, data di rilascio, data di scadenza e tipologia di certificato. Si specifica che un utente potrebbe non aver acquisito alcun certificato. Esistono due tipi di utenti: volontari e amministratori. Un amministratore ha la possibilità di organizzare delle missioni e, all'occorrenza, di scaricarne il resoconto (solamente delle missioni da lui organizzate; lo scarico del resoconto può essere effettuato una sola volta, e il sistema deve tenere traccia della data di scarico). Le missioni avvengono in una certa data e in un certo luogo, e vi partecipano uno o più utenti. Nel sistema si tiene traccia anche dei veicoli utilizzati nelle missioni, registrandone targa, modello, numero di posti e tipo di patente necessaria per la guida. Per ogni partecipazione di un utente ad una missione, il sistema registra l'ora di inizio e di fine permanenza, il veicolo utilizzato e il ruolo dell'utente nell'uso del veicolo (cioè, se sia guidatore o passeggero). A fini di monitoraggio, è importante calcolare e mantenere l'ammontare delle ore uomo totali della missione, somma delle permanenze di tutti gli utenti. Infine, per ogni missione si tiene traccia dei tipi di certificati che gli utenti devono possedere per potervi partecipare.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (12/20 punti).

2) Si modelli mediante un *diagramma degli stati* il seguente contesto, che descrive le modalità di comportamento di una persona durante una pandemia. Si supponga che, in partenza, la persona sia identificata come *sana*. Qualora la persona sana abbia un *contatto casuale* con un paziente positivo, essa deve effettuare il cosiddetto *monitoraggio passivo* (ossia adottare particolari misure di cautela) per un periodo massimo di 14 giorni. Durante questa fase (ma anche in assenza di contatti casuali), il verificarsi di un *contatto stretto* con un paziente positivo fa scattare la *quarantena*. La fase di quarantena dura anch'essa un massimo di 14 giorni; in questo caso, però, la persona deve rimanere chiusa in casa. Al termine della quarantena, la persona può considerarsi nuovamente sana. In qualunque momento, il manifestarsi di sintomi fa scattare l'*isolamento*: anch'esso prevede che la persona rimanga chiusa in casa, ma deve eseguire il tampone su richiesta dell'AUSL, potenzialmente anche più volte. La persona può uscire dall'*isolamento* solamente se (1) l'esito del tampone è negativo, o (2) sono passati 21 giorni dall'inizio dell'*isolamento*. Si supponga che l'esito del tampone sia istantaneo. Anche in questo caso, la persona può considerarsi sana all'uscita dall'*isolamento* (8/12 punti).

3) Un operaio effettua turni settimanali di lavoro alle macchine. Durante ogni turno ciascun operaio lavora a una singola macchina, e ogni macchina è manovrata da un singolo operaio. Scegliere, tra le seguenti soluzioni di modellazione, quella ritenuta più appropriata.



4) Cosa si intende per “messa in produzione” del software?

- il momento in cui inizia la fase di analisi dei requisiti
- il momento in cui inizia la fase di esercizio del software
- il momento in cui inizia la fase di implementazione

- d. il momento in cui inizia la fase di diagnosi e manutenzione
- e. il momento in cui inizia la fase di progettazione del software
- f. il momento in cui inizia la fase di collaudo
- g. il momento in cui il software viene installato

5) Quali tra i seguenti sono meccanismi di *estendibilità* in UML?

- a. specifiche
- b. viste
- c. profili
- d. diagrammi
- e. proprietà
- f. stereotipi

6) E' dato il seguente frammento di pseudocodice:

```
begin
  read(A,B,C)
  if (A>0) AND (B<0) then
    C := C/(A+B)
end
```

Quali dei seguenti insiemi di test soddisfano il *criterio di copertura delle decisioni*?

- a. $\{(A=2, B=-1, C=3), (A=-1, B=2, C=0)\}$
- b. $\{(A=2, B=1, C=0), (A=-1, B=-1, C=3)\}$
- c. $\{(A=2, B=-1, C=3)\}$
- d. $\{(A=2, B=-1, C=3), (A=-1, B=-1, C=3)\}$

7) Cosa si intende per *affordance* in ambito interfacce utente?

- a. il posizionamento strutturato di testo, disegni e controlli all'interno di un'area grafica considerata
- b. gli aspetti grafici di un oggetto che invitano l'utente a interagire con quell'oggetto in un certo modo
- c. una figura che illustra un oggetto attraverso una somiglianza o un'analogia con un concetto del mondo reale

8) Quali tra le seguenti tecniche possono essere ricondotte al *principio di separazione degli argomenti*?

- a. la capacità di prevedere i cambiamenti a cui il software sarà sottoposto durante il suo ciclo di vita
- b. la ricerca di un problema noto che generalizzi il problema da risolvere
- c. l'utilizzo di formalismi e metodologie standard
- d. l'identificazione, nel ciclo di vita, di attività distinte e temporalmente correlate
- e. l'identificazione di moduli distinti e interconnessi all'interno del software

1) Si vuole modellare il dominio delle aziende telematiche. Di ogni azienda telematica si vogliono memorizzare il nome e la partita IVA. L'azienda è composta da una serie di dipartimenti, che a loro volta si dividono in uno o più team. Un team è formato da dipendenti (descritti da nome, cognome, data di nascita, documento di identità). I dipendenti junior vengono supervisionati da uno o più senior. Inoltre, ogni team ha, tra i suoi membri, un responsabile. Per ciascun dipendente si vuole tener traccia dei diversi contratti (ogni contratto è descritto da data inizio contratto, data fine contratto, RAL). Inoltre di ciascun dipendente si vuole memorizzare l'attuale TFR accumulato (calcolato in base alle RAL dei diversi contratti emanati). Ogni azienda telematica ha sempre almeno due dipartimenti: il dipartimento del contenuto e il dipartimento della pubblicità. Il dipartimento del contenuto idea uno o più programmi televisivi (di cui si memorizzano nome, target di età e tipologia) e ne decide la messa in onda. Un programma televisivo va in onda in uno o più canali televisivi (descritti da nome e numero) in certi giorni e ore (anche in contemporanea). Di ogni messa in onda si vogliono memorizzare lo share (numero di utenti che hanno visto il programma), per poi calcolare lo share medio dell'intero programma televisivo. Ogni dipartimento di pubblicità cura in esclusiva la pubblicità di uno o più canali televisivi.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (12/32 punti).

2) Si vuole modellare il funzionamento di IoTManager, un framework per la gestione di dispositivi IoT. IoTManager consente a un utente di inviare comandi a un dispositivo IoT. Tali comandi sono eseguiti tramite richieste HTTP generate da un "Context Broker", il quale comanda un dispositivo IoT tramite un "IoT agent". L'invio del comando funziona come segue. L'utente inoltra il comando al Context Broker, il quale verifica che il dispositivo sia effettivamente connesso al sistema. Se il dispositivo non è connesso, il Context Broker notifica l'errore all'utente. Se il dispositivo è connesso, il Context Broker genera una richiesta HTTP, inoltra la richiesta all'IoT Agent, e attende la risposta dell'IoT Agent. Quando l'IoT Agent riceve la richiesta, interpreta il comando e lo demanda al dispositivo IoT. Una volta eseguito il comando, il dispositivo IoT restituisce l'esito all'IoT Agent, il quale genera una risposta HTTP da restituire al Context Broker. Ricevuta la risposta, il Context Broker notifica l'esito all'utente.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma di sequenza* (8/32 punti).

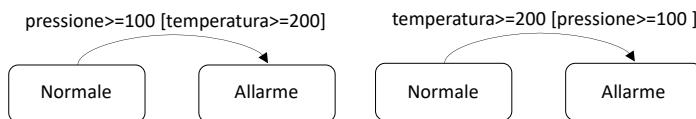
3) Dato il seguente frammento di pseudocodice, se ne calcoli la *complessità ciclomatica*:

```

begin
  read(N);
  read(x);
  for i:=1 to N do
    read(V[i]);
  i:=1;
  K:=0;
  while (i<N) AND (K<100) do
  begin
    if (x<>0)
      V[i]:=(V[i]+V[i+1])/x;
    K:=K+V[i];
    i:=i+1;
  end
end

```

4) Dati i due seguenti frammenti di diagrammi degli stati, dire se modellano lo stesso comportamento al variare di temperatura e pressione.



- a. SI
- b. NO
- c. solo quando o temperatura o pressione si mantengono costanti

5) In un diagramma di deployment, quali tra i seguenti possono essere modellati come *manufatti*?

- a. un file sorgente Java
- b. un file eseguibile
- c. un PC usato come server
- d. l'ambiente Apache

6) Quali sono i risultati attesi da uno *studio di fattibilità*?

- a. stima dei costi del software
- b. stima dei tempi di realizzazione del software
- c. valutazione dei benefici del software
- d. macroanalisi dei requisiti
- e. progetto di massima del software
- f. casi d'uso del software

7) A quali di questi criteri è consigliato attenersi nella *scelta dei colori* per un'interfaccia grafica?

- a. basarsi su un codice di due soli colori
- b. basarsi su un codice di non più di cinque colori
- c. usare colori vivaci per aree grandi e neutri per aree piccole
- d. non usare colori contrastanti tra loro per evitare affaticamento della vista
- e. se lo sfondo è chiaro, usare un colore scuro per il testo
- f. usare colori brillanti per applicazioni gestionali

8) Cosa afferma il *principio di anticipazione dei cambiamenti*?

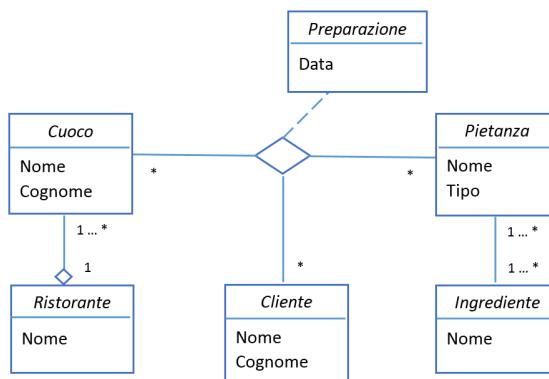
- a. un software deve soddisfare esclusivamente le specifiche funzionali attuali
- b. un software deve soddisfare sia le specifiche funzionali attuali sia quelle prevedibili per il breve/medio termine
- c. un software deve soddisfare le specifiche funzionali attuali ma essere predisposto per poter soddisfare, con bassi costi di manutenzione, anche quelle prevedibili per il breve/medio termine

1) Si vuole modellare un database a supporto del processo di contravvenzione. Una contravvenzione viene elevata da un vigile o da un ausiliario a un veicolo in una certa data e ora e in un certo luogo, a seguito di una certa infrazione. Le possibili infrazioni sono classificate in tipi, e ciascuna di esse può determinare o meno una decurtazione di punti. Per ciascuna contravvenzione si memorizzano una descrizione testuale, l'importo, e l'eventuale numero di punti decurtati. I veicoli sono identificati dal numero di targa e descritti da modello (che appartiene a una marca), cilindrata, alimentazione (benzina, diesel, elettrica o gpl) e anno di immatricolazione. Ogni veicolo è intestato a un proprietario, identificato da codice fiscale e descritto da dati anagrafici nonché dalla data di conseguimento della patente. I vigili sono identificati dalla loro matricola e descritti da dati anagrafici più anzianità di servizio; ogni vigile fa capo a un comando. Gli ausiliari sono identificati da codice fiscale e descritti da dati anagrafici più data di assunzione.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (12/31 punti).

2) Dato il diagramma delle classi in figura, effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro (8/31 punti):

1. Visualizzare tutti i clienti serviti da un dato ristorante in una certa settimana (1 volta alla settimana).
2. Visualizzare, per ciascun cuoco, l'elenco di tutte le pietanze da preparare in data odierna e i corrispondenti ingredienti (1 volta al giorno).
3. Visualizzare, per un dato cliente, l'elenco dei ristoranti che lo hanno servito nel corso del mese (1 volta al mese).



3) Si confrontino i due seguenti frammenti di *diagrammi delle classi*:



- a. sono perfettamente equivalenti
- b. quello di sinistra è più generale
- c. quello di destra è più generale

4) Cosa si intende per *robustezza* di un software?

- a. la sua capacità di reagire in modo "ragionevole" anche a fronte di situazioni non previste dalle specifiche
- b. l'assenza di bug
- c. il fatto che rispetta le specifiche di progetto
- d. la sua capacità di cooperare con altri software

5) Quali tipi di relazioni tra package possono comparire in un *diagramma dei package* in UML 2?

- a. contenimento
- b. specializzazione
- c. associazione
- d. dipendenza
- e. aggregazione
- f. interazione

6) Quali tra le seguenti sequenze, risultanti dall'applicazione della tecnica di *analisi del flusso dei dati* a una variabile usata in un frammento di codice, vengono considerate anomale?

- a. duuduua
- b. aduuddua
- c. dauuduua
- d. aduuaduua

7) Quale dei seguenti *tipi di interfaccia* è da ritenersi ottimale per un'applicazione di acquisto biglietti da installare su un totem presso una stazione ferroviaria?

- a. interfaccia code-based
- b. interfaccia 3270
- c. pseudo-GUI
- d. standard GUI
- e. special GUI

8) Quali delle seguenti affermazioni riguardanti la *prototipazione usa-e-getta* sono vere?

- a. permette di dimostrare in anticipo i requisiti agli utenti
- b. tende a corrompere il sistema rendendone costosa la manutenzione
- c. si focalizza sui requisiti meno chiari
- d. può essere usata per prototipare una interfaccia utente

1) La piattaforma Petflix permette la visione di film, serie e docufilm. Ciascuno di questi è caratterizzato da titolo, regista, attori principali, genere e una breve descrizione (sinossi). In più, film e docufilm hanno una durata, mentre le serie si articolano in stagioni numerate, ciascuna delle quali include diversi episodi; ogni singolo episodio ha un suo titolo, una sua sinossi e una durata. Ciascun utente registrato (identificato dalle sue credenziali di accesso) ha una lista di contenuti che intende vedere (“la mia lista:”). Inoltre, per ogni contenuto che ha iniziato a vedere, occorre tenere traccia del punto esatto da cui riprendere la visione (“continua a vedere:”). L’elenco dei contenuti già interamente visionati viene mantenuto per sei mesi. Sulla base dei contenuti già visti, la piattaforma propone un elenco di contenuti di interesse a ciascun utente (“vedi anche:”). Un utente può esprimere il suo gradimento per un qualsiasi contenuto che ha visto (“mi piace”/“non mi piace”), e può suggerire un contenuto a un altro utente (“condividi” via e-mail o WhatsApp). Quando esce una nuova stagione di una serie che un certo utente ha seguito, gli viene inviata una notifica dalla piattaforma. Se in fase di registrazione un utente si è qualificato come “bambino”, alcuni contenuti non gli vengono mostrati.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (12/31 punti) e un *diagramma dei casi d’uso* (8/31 punti).

2) In quale delle seguenti fasi del ciclo di vita vengono definite le *specifiche* per un software?

- a. pianificazione
- b. analisi
- c. progettazione del sistema
- d. progettazione esecutiva
- e. realizzazione
- f. collaudo

3) Un software viene modificato per cambiare la funzione di calcolo di una provvigione a seguito di una disposizione di legge. Di che tipo di *manutenzione* si tratta?

- a. Correttiva
- b. Evolutiva
- c. Perfettiva
- d. Adattiva

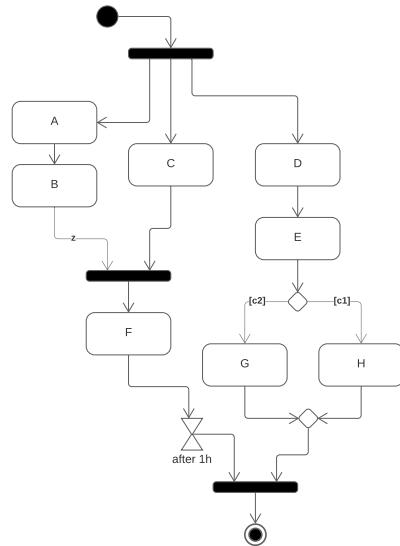
4) Nell’ambito di Unified Process (UP), cosa si intende per *elaboration*?

- a. la fase in cui si definiscono gli obiettivi di progetto
- b. la fase in cui vengono definite le caratteristiche funzionali, strutturali e architetturali del software
- c. la fase in cui il software viene sviluppato e collaudato
- d. la fase in cui il software viene consegnato, installato e configurato

5) Quali delle seguenti affermazioni generali relative al *linguaggio UML* sono vere?

- a. UML è un linguaggio standard basato su un metamodello
- b. UML deve essere utilizzato in abbinamento a Unified Process
- c. un’entità di UML può essere rappresentata con notazioni diverse a seconda del diagramma in cui compare
- d. un’entità di UML può essere rappresentata con notazioni diverse a seconda del livello di dettaglio che si vuole mostrare
- e. un modello è una particolare visualizzazione di alcuni tipi di elementi di un diagramma
- f. una associazione è un caso particolare di aggregazione
- g. se una classe è un’interfaccia, allora è una classe astratta
- h. se una classe è astratta, allora è una interfaccia

6) Dato il *diagramma di attività* in figura, quali delle seguenti affermazioni riguardanti le azioni raffigurate sono vere?



- a. A deve essere svolta prima di C
- b. B viene terminata dall'evento z
- c. D può essere svolta prima di B
- d. F può essere svolta prima di D
- e. se la condizione c1 è falsa viene eseguita G
- f. dopo un'ora dal termine di F, l'attività termina
- g. F può iniziare solo quando C è terminata
- h. G e H possono essere svolte concorrentemente
- i. B e C possono essere svolte concorrentemente

7) E' data una funzionalità di stampa dell'anagrafica completa di un utente. In quale modo può essere classificata nel metodo *function points*?

- a. ILF
- b. EIF
- c. EI
- d. EO
- e. EQ

1) Un gruppo di ricerca è formato da membri, ciascuno descritto da matricola, nome, e cognome. Ciascun membro può essere supervisionato da uno o più altri membri. Tra i membri, si distinguono professori e dottorandi. In ciascun anno accademico, un professore può insegnare in un numero variabile di corsi universitari. Dei corsi universitari si vogliono memorizzare nome, settore scientifico disciplinare, crediti formativi e numero di ore previste (derivabili dal numero di crediti formativi). In ciascun anno un corso è tenuto da un professore, eventualmente con la collaborazione di dottorandi col ruolo di tutor. Ciascun membro può essere autore di articoli in riviste scientifiche. Le riviste hanno un nome e un settore scientifico di appartenenza. Per un articolo si vogliono memorizzare titolo, lista degli autori e data di pubblicazione. Una pubblicazione può citare al suo interno altre pubblicazioni. Risulta utile memorizzare esplicitamente, per ogni pubblicazione, il numero di citazioni ricevute.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (12/31 punti).

2) Si modelli con un *diagramma di sequenza* (8/31 punti) il processo di richiesta del bonus terme da parte di un cittadino. Il cittadino richiede il buono presso un centro termale, inviando i propri dati anagrafici. Il centro termale si fa carico di richiedere il buono, per conto del cittadino, a un ente denominato Invitalia. Quest'ultimo avvia un processo di controllo per verificare la disponibilità e ne riporta l'esito al centro termale, che a sua volta riporta l'esito al cittadino. Nel caso in cui l'esito sia positivo, il centro termale provvede ad inviare i dati anagrafici del cittadino ad Invitalia, il quale avvia un processo per la generazione del buono. Una volta generato il buono, esso viene inviato al centro termale che a sua volta lo invia al cittadino.

3) In ambito qualità del software, cosa si intende per *produttività*?

- a. l'efficienza del processo di produzione del software in termini di velocità di consegna
- b. la capacità del processo di produzione del software di valutare e rispettare i tempi di consegna del prodotto
- c. il fatto che le sue caratteristiche (correttezza, performance, ecc.) siano facilmente valutabili

4) Cosa viene restituito come risultato dal metodo *Co.Co.Mo*?

- a. il numero di punti funzione di un software
- b. il numero di mesi uomo per lo sviluppo di un software
- c. il numero di punti di biforcazione nel flusso di controllo di un software
- d. il numero dei cammini indipendenti nel flusso di controllo di un software
- e. il costo economico di un software

5) Quale di questi modelli di produzione del software è specificamente pensato per la gestione del *rischio*?

- a. modello a cascata
- b. modello RAD
- c. modello evolutivo a spirale
- d. model-driven development
- e. modello extreme programming

6) Qual è la frequenza tipica delle visite di sorveglianza in ambito *certificazione ISO 9000 del software*?

- a. da una a quattro all'anno
- b. una al mese
- c. una ogni 2-3 anni

7) A quali tipi di elementi UML si applica la relazione di *contenimento*, rappresentata da un segno + circondato da un cerchio?

- a. alle classi
- b. ai casi d'uso
- c. ai package
- d. agli stati
- e. ai componenti
- f. ai nodi

8) E' data una funzionalità che calcola e visualizza il cedolino di stipendio per un dipendente a partire dai suoi dati di carriera. In quale modo può essere classificata nel metodo *function points*?

- a. ILF (file interno logico)
- b. EIF (file esterno di interfaccia)
- c. EI (input esterno)
- d. EO (output esterno)

e. EQ (interrogazione esterna)

9) In che modo si modella il legame tra un componente e il manufatto che lo implementa in un *diagramma di deployment* UML?

- a. dipendenza
- b. associazione
- c. realizzazione
- d. raffinamento
- e. aggregazione
- f. specializzazione

10) Nell'ambito di Unified Process (UP), cosa si intende per *inception*?

- a. la fase in cui si definiscono gli obiettivi di progetto
- b. la fase in cui vengono definite le caratteristiche funzionali, strutturali e architetturali del software
- c. la fase in cui il software viene sviluppato e collaudato
- d. la fase in cui il software viene consegnato, installato e configurato

11) Un software viene modificato per migliorare l'efficienza di una funzione di caricamento massivo di dati. Di che tipo di *manutenzione* si tratta?

- a. Correttiva
- b. Evolutiva
- c. Perfettiva
- d. Adattiva

12) Quale dei seguenti *tipi di interfaccia* è da ritenersi ottimale per un'applicazione di editing di oggetti grafici in cui sia richiesta elevata flessibilità del flusso di lavoro?

- a. interfaccia code-based
- b. interfaccia 3270
- c. pseudo-GUI
- d. standard GUI
- e. special GUI

1) Com'è classificata l'*efficienza* di un software?

- a. Esterna e di Processo
- b. Interna e di Prodotto
- c. Esterna e di Prodotto**
- d. Interna e di Processo

2) Le *interfacce 3270* sono pensate per situazioni in cui...

- a. ...l'utente deve effettuare un data-entry massivo**
- b. ...il workflow è flessibile
- c. ...la soddisfazione dell'utente nell'uso dell'interfaccia è importante
- d. ...la riusabilità delle conoscenze acquisite non è importante**
- e. ...obiettivo primario è l'autoesplicazione

3) E' dato il seguente frammento di pseudocodice:

```
begin
  read(A,B,C)
  if (A>0) AND (B<0) then
    C := C/(A+B)
end
```

Quali dei seguenti insiemi di test soddisfano il *criterio di copertura dei comandi (o programmi)*?

- a. {(A=2, B=-1, C=3), (A=-1, B=2, C=0)}
- b. {(A=2, B=1, C=0), (A=-1, B=-1, C=3)}
- c. {(A=2, B=-1, C=-3)}**
- d. {(A=2, B=0, C=3), (A=-1, B=-1, C=3)}

4) Quali delle seguenti affermazioni riguardanti la *prototipazione evolutiva* sono vere?

- a. permette di dimostrare in anticipo i requisiti agli utenti**
- b. tende a corrompere il sistema rendendone costosa la manutenzione**
- c. si focalizza sui requisiti meno chiari
- d. può essere usata per prototipare una interfaccia utente**

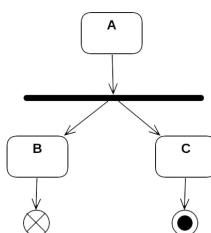
5) E' data una funzionalità che mostra all'utente un form con diversi campi da compilare; alla pressione del tasto "ok", i dati letti vengono salvati su database. In quale modo può essere classificata nel metodo *function points*?

- a. ILF
- b. EIF
- c. EI**
- d. EO
- e. EQ

6) Quale di questi modelli di produzione del software è specificamente pensato per ridurre la durata del ciclo di sviluppo per software facilmente *modularizzabili*?

- a. modello a cascata
- b. modello RAD**
- c. modello evolutivo a spirale
- d. model-driven development
- e. modello extreme programming

7) Con riferimento alle azioni A, B e C nel frammento di *diagramma di attività UML* mostrato in figura, quali delle seguenti affermazioni sono vere?



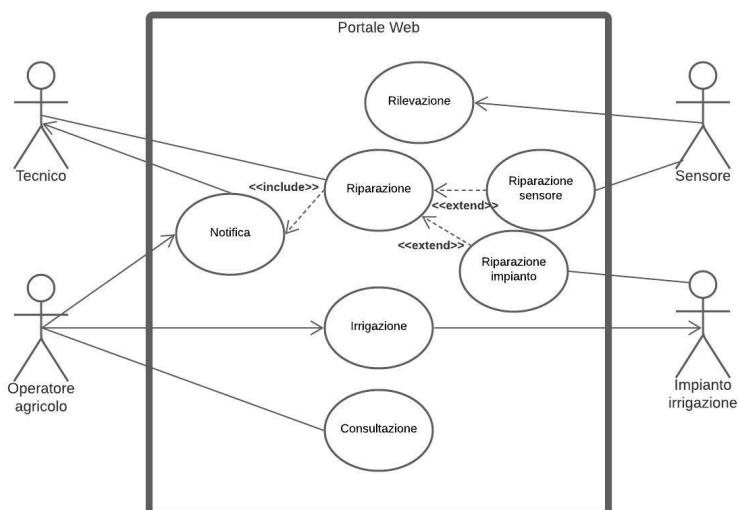
- a. A non può iniziare se non è terminata B

- b. C non può iniziare se non è terminata A**
- c. solo una tra B e C può avere luogo
- d. B e C possono avere luogo in contemporanea**
- e. quando B termina, viene terminata anche C
- f. quando C termina, viene terminata anche B**

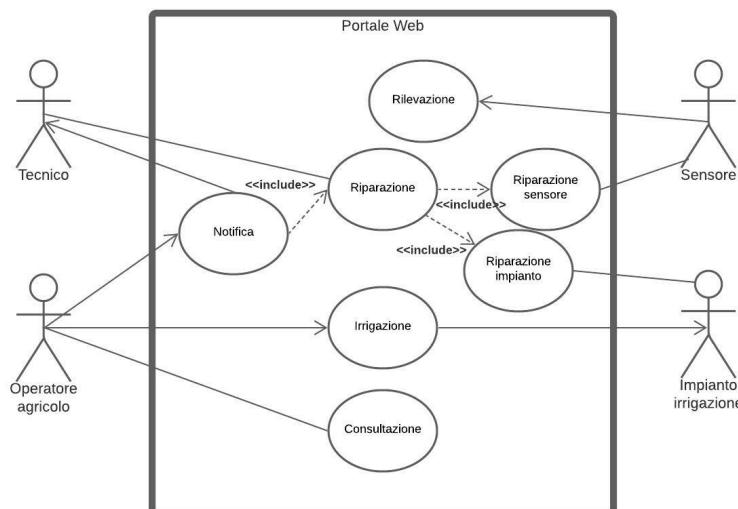
8) Si vuole modellare un portale web che supporta un sistema di smart irrigation di campi agricoli. Un campo (identificativo, tipo coltura) è costituito da un impianto di irrigazione e da uno o più sensori. Dell'impianto di irrigazione si vuole memorizzare soltanto la portata, mentre dei sensori si vogliono memorizzare l'identificativo e il valore medio rilevato. I sensori possono essere di varie tipologie (ad esempio, "Umidità terreno", "Velocità vento", "Temperatura Aria"). Del sensore relativo all'umidità del terreno si vuole memorizzare la profondità a cui è posizionato. Inoltre, per ciascun campo risulta interessante tener traccia del numero di sensori posizionati. Ciascun sensore aggiorna il portale web tramite trasmettendo delle rilevazioni, per ciascuna delle quali si memorizzano il timestamp di invio e il valore misurato. Gli operatori agricoli consultano il portale web e azionano di conseguenza l'impianto di irrigazione. Di ciascuna irrigazione si devono memorizzare il timestamp di inizio e la durata in minuti. Se un operatore si accorge di un qualsiasi malfunzionamento ne invia notifica a un tecnico, il quale si preoccuperà della riparazione; una riparazione può riguardare un sensore o un impianto di irrigazione.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (14/31 punti).

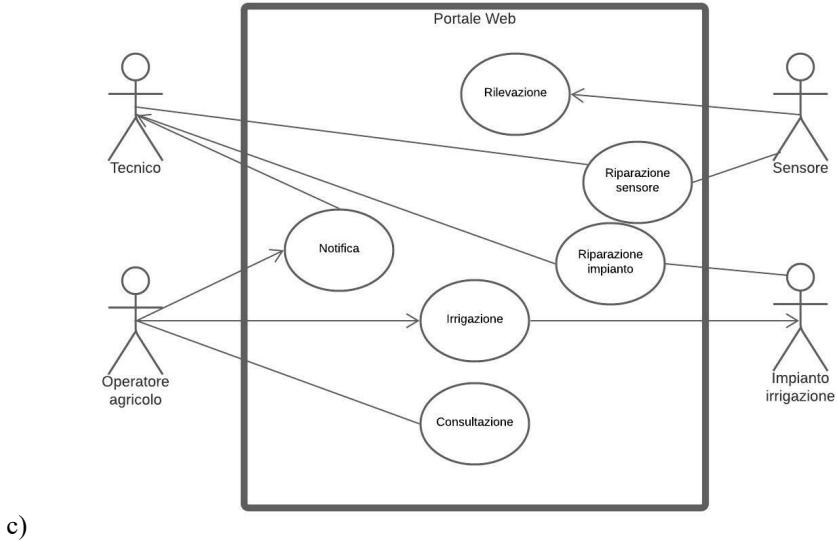
9) Quali tra i seguenti *diagrammi dei casi d'uso* modellano le specifiche precedenti in modo corretto? (6/31 punti)



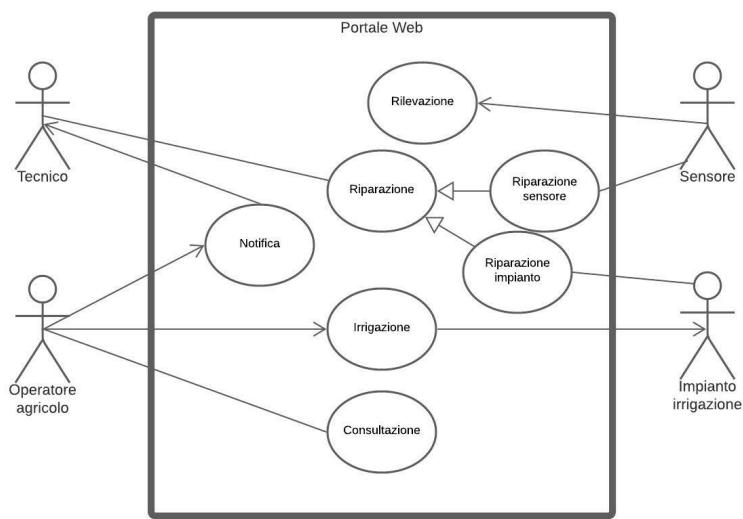
a)



b)



c)



d)

1) Nel metodo Co.Co.Mo, qual è il ruolo degli *stimatori di costo*?

- a. **rifinire la stima di mesi nominali**
- b. calcolare il numero di mesi nominali in funzione della classe del software
- c. permettere di determinare la classe del software

2) Com'è definita, in termini generali, una *dipendenza* in UML?

- a. **A dipende da B se una modifica in B può comportare una modifica in A**
- b. A dipende da B se una modifica in A può comportare una modifica in B
- c. A dipende da B se A usa dei servizi di B
- d. A dipende da B se A implementa l'interfaccia di B

3) Dato il seguente frammento di pseudocodice, se ne calcoli la *complessità ciclomatica*:

```
begin
    read(x);
    read(y);
    if (x=0)
        then x:=1;
    y:=y/x;
    if (x>0)
        then print(x);
        else print(y);
end
```

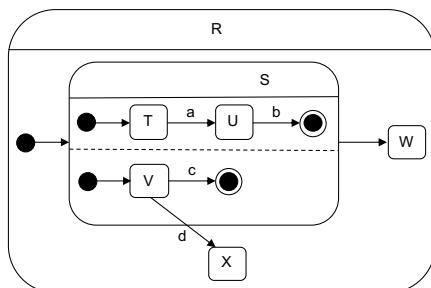
4) Nel *testing in the large*, quale diagramma UML è principalmente usato nel determinare gli insiemi di test da utilizzare?

- a. **diagramma dei casi d'uso**
- b. diagramma delle classi
- c. diagramma dei package
- d. diagramma delle attività
- e. diagramma degli stati
- f. diagramma di sequenza
- g. diagramma dei componenti

5) Quali tra i seguenti aspetti caratterizzano i *modelli agili* di produzione del software?

- a. sono prescrittivi
- b. incoraggiano la consegna anticipata del software**
- c. scoraggiano l'incrementalità
- d. richiedono un elevato livello di formalità nella documentazione
- e. richiedono frequente comunicazione tra sviluppatori e utenti**
- f. impiegano team di progettisti molto ampi

6) Con riferimento al *diagramma degli stati* UML mostrato in figura, quali delle seguenti affermazioni sono vere?

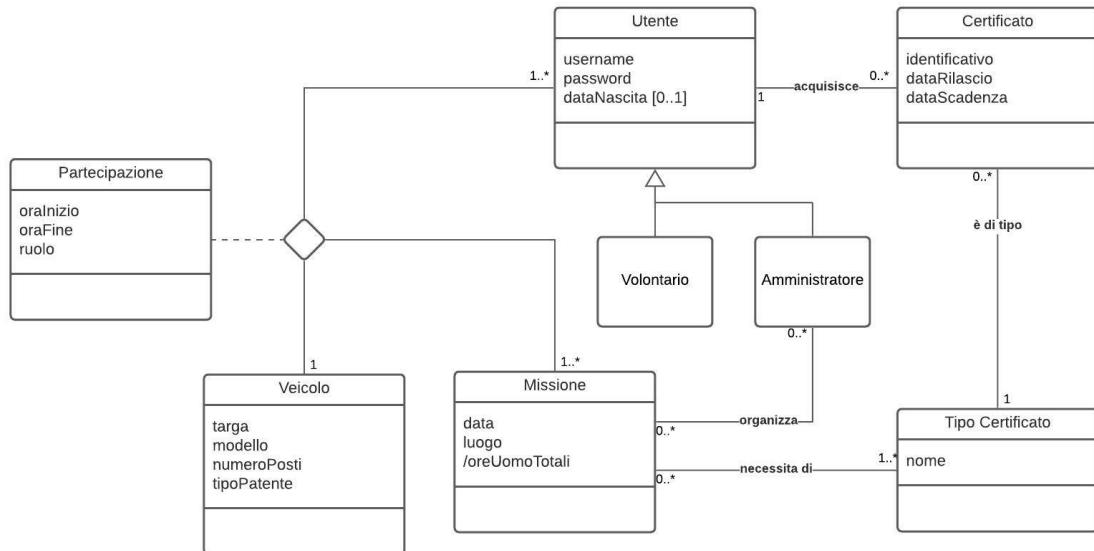


- a. quando l'oggetto si trova nello stato R, può trovarsi indifferentemente negli stati S, X e W
- b. quando l'oggetto entra nello stato R, entra negli stati T e V
- c. quando accade l'evento b, l'oggetto entra nello stato W
- d. l'oggetto può trovarsi contemporaneamente negli stati U e V**
- e. l'oggetto può trovarsi contemporaneamente negli stati U e X
- f. l'oggetto non può raggiungere lo stato X se prima non è accaduto l'evento b

7) A quali di questi criteri è consigliato attenersi nella *scelta dei colori* per un'interfaccia grafica?

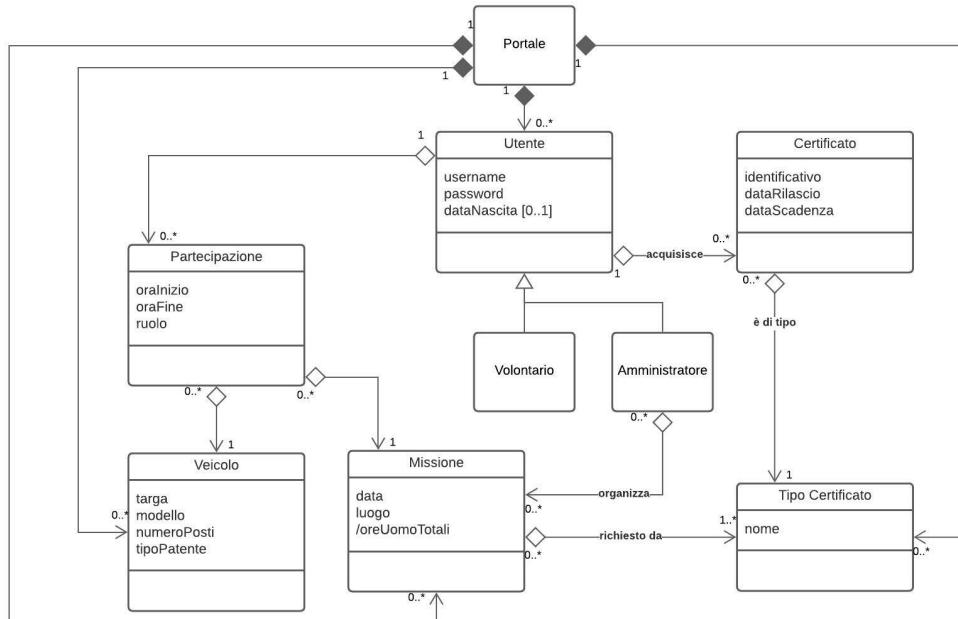
- a. basarsi su un codice di due soli colori
- b. basarsi su un codice di non più di cinque colori**
- c. usare colori vivaci per aree grandi e neutri per aree piccole
- d. non usare colori contrastanti tra loro per evitare affaticamento della vista
- e. se lo sfondo è chiaro, usare un colore scuro per il testo
- f. usare colori brillanti per applicazioni gestionali

8) E' dato il diagramma delle classi in figura, che rappresenta un portale per la gestione delle missioni effettuate dai volontari della vigilanza antincendio.

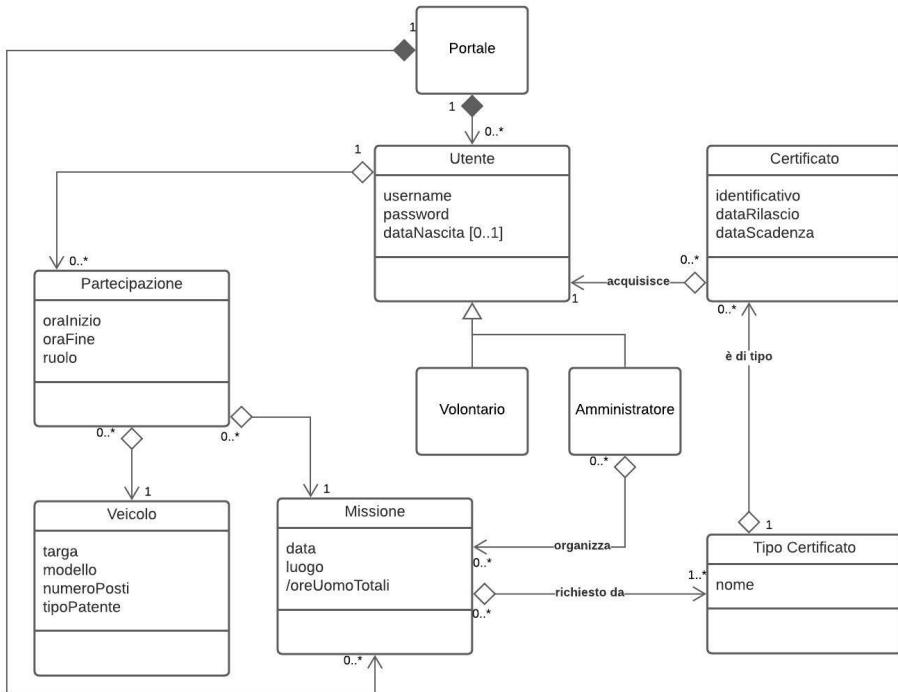


Tra le seguenti, indicare le soluzioni di progetto delle associazioni ritenute corrette alla luce del seguente carico di lavoro:

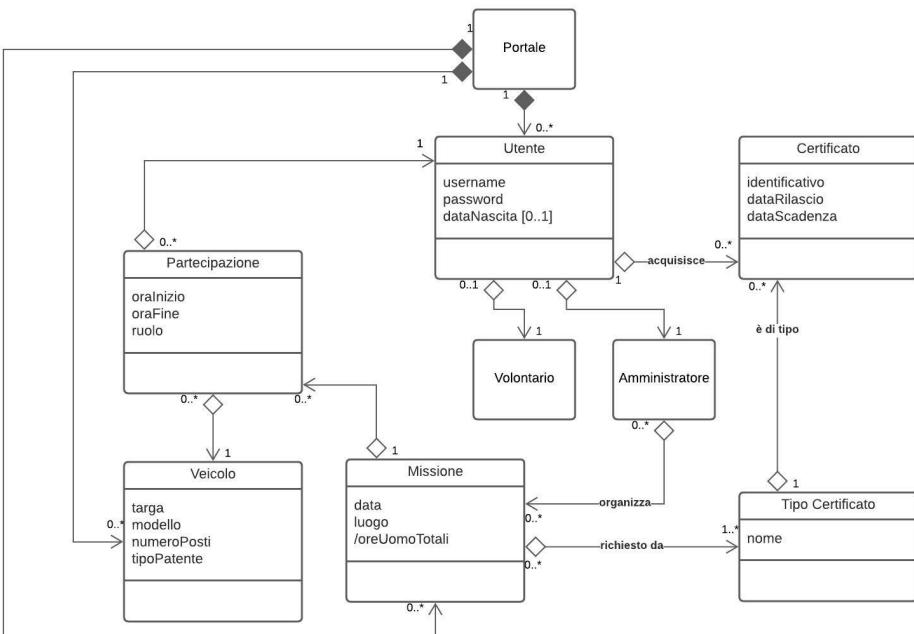
- visualizzare gli utenti che possiedono tutti i certificati richiesti da una certa missione (più volte al giorno);
- visualizzare le missioni organizzate da un certo amministratore (una volta alla settimana);
- visualizzare le ultime 3 missioni a cui un dato volontario ha partecipato (una volta al mese).



a)



b)



c)

9) Un'officina necessita di un portale che gestisca clienti, dipendenti e le loro interazioni. Di ogni cittadino (cliente o dipendente) si vogliono memorizzare: codice fiscale, nome, cognome e data di nascita. I dipendenti si differenziano in base al reparto di appartenenza: reparto riparazioni e reparto compravendita. Il primo è formato da meccanici, il secondo da agenti automobilistici. Di ciascuno di essi si vuole memorizzare la retribuzione oraria. Nel reparto riparazioni, ogni riparazione riguarda un veicolo e viene effettuata da uno o più meccanici. Di ciascuna riparazione è importante tener traccia del numero di ore dedicate da ogni meccanico e di eventuali pezzi di ricambio utilizzati (casa produttrice, modello di riferimento, descrizione pezzo, costo). Inoltre, si vogliono memorizzare la data di inizio e fine riparazione, nonché il costo totale della riparazione (essendo quest'ultimo derivabile dalla retribuzione oraria di ogni meccanico che ha contributo alla riparazione in oggetto, dal relativo numero di ore impiegate e dal costo di ciascun pezzo di ricambio). Nel reparto compravendita, ogni transazione di compravendita è effettuata da un agente automobilistico con un cliente e riguarda un veicolo. Di ogni transazione si vuole memorizzare se questa sia di acquisto o di vendita, e il prezzo contrattato. Di ogni veicolo si vogliono memorizzare la casa produttrice, il modello, la data di produzione e la targa (nel caso sia già stata messa in circolazione). Inoltre, un veicolo può essere oggetto di molteplici attestati di proprietà (data inizio, data fine), ognuno che specifica la relazione di proprietà tra il veicolo e il suo attuale proprietario (cittadino). Chiaramente, la data di fine non è specificata nell'attestato attualmente in vigore (l'ultimo

realizzato). Si noti che un cittadino può acquistare lo stesso veicolo in intervalli temporali diversi (venderlo per poi riacquistarlo).

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (14/31 punti).

1) Quali tra le seguenti sequenze, risultanti dall'applicazione della tecnica di *analisi del flusso dei dati* a una variabile usata in un frammento di codice, vengono considerate anomale?

- a. duaduuua
- b. aduaddua**
- c. dauuduua
- d. aduduadu

2) Com'è classificata la *portabilità* di un software?

- a. Esterna e di Processo
- b. Interna e di Prodotto
- c. Esterna e di Prodotto**
- d. Interna e di Processo

3) Quali delle seguenti affermazioni riguardanti il *numero ciclomatico* sono vere?

- a. il numero ciclomatico di un grafo fortemente connesso è il numero massimo di archi che si possono eliminare per trasformarlo in un albero
- b. il numero ciclomatico di un programma esprime il numero di cammini linearmente indipendenti nel suo grafo di controllo**
- c. il numero ciclomatico di un programma è pari al numero dei punti di decisione del programma
- d. la complessità ciclomatica di un modulo non dovrebbe superare il valore 50

4) Quali sono le differenze tra *collaudo in fabbrica (alfa-test)* e *collaudo del sistema installato (beta-test)*?

- a. il primo è effettuato dagli sviluppatori, il secondo dagli utenti finali**
- b. il primo viene effettuato su una versione prototipale del software, il secondo sulla versione finale
- c. il primo mira a trovare errori e malfunzionamenti nel software, il secondo a migliorarne le prestazioni
- d. il primo viene effettuato prima della messa in esercizio, il secondo dopo

5) Cos'è uno *stereotipo* in UML?

- a. una variazione di un elemento di modellazione esistente, con la stessa forma ma diverso scopo**
- b. un simbolo grafico o testuale che permette di definire nuovi elementi di modellazione nel linguaggio
- c. una stringa tra parentesi angolari che si può applicare a dipendenza per specificarne il significato
- d. una icona che si può sostituire a una classe per specificarne il significato

6) Un software viene modificato per introdurre una funzione di reporting sulle abitudini di acquisto dei clienti. Di che tipo di *manutenzione* si tratta?

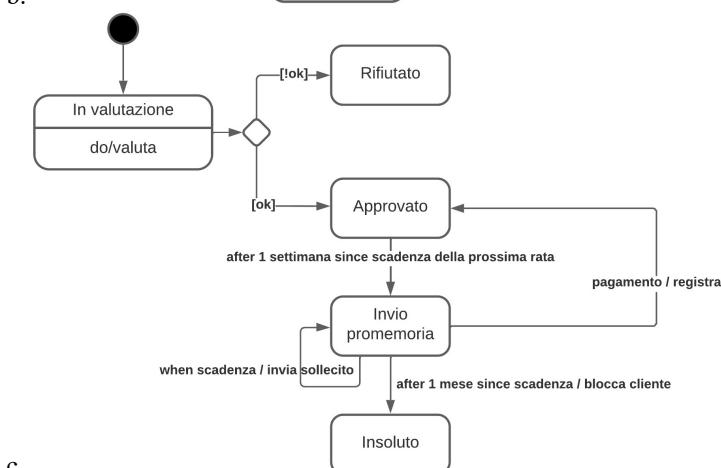
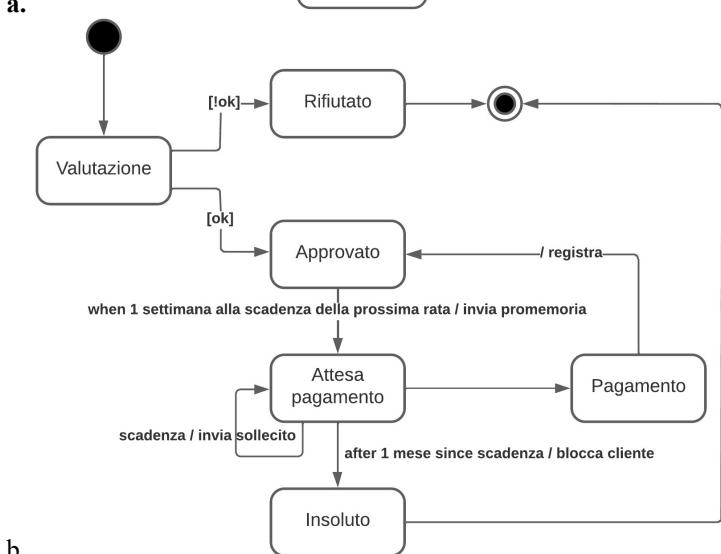
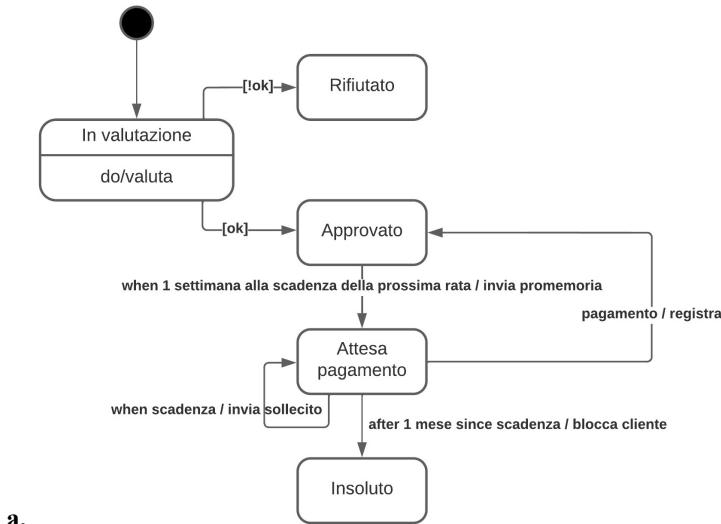
- a. Correttiva
- b. Evolutiva**
- c. Perfettiva
- d. Adattiva

7) In quali situazioni è particolarmente importante, tra i fattori di usabilità di un'interfaccia, l'*apprendibilità*?

- a. quando il ciclo di vita del software utilizzato è lungo
- b. quando il turn-over degli utenti è elevato**
- c. quando è necessario contenere i tempi del training**
- d. quando gli utenti usano il software saltuariamente**
- e. quando i risultati del software sono immediatamente visibili ai clienti esterni

8) Un prestito è approvato da un agente a favore di un cliente. Se il prestito viene approvato si stabilisce un piano di rate per la restituzione. Una settimana prima della scadenza di una rata, l'agente invia al cliente un promemoria. Se il pagamento della rata perviene entro la scadenza, esso viene registrato. In caso contrario, si invia un sollecito al cliente. Se il pagamento non perviene entro un mese dalla scadenza della rata, il cliente viene bloccato e non gli si concedono altri prestiti.

Tra i seguenti, indicare il diagramma degli stati per la classe *prestito* che più correttamente modella queste specifiche.



9) Si vuole modellare un sistema informativo per la gestione degli esami universitari. Di ogni persona si vogliono memorizzare: codice fiscale, nome, cognome, data di nascita, e le città di nascita e di residenza (con le rispettive nazioni di appartenenza). Ogni corso ha come titolare un docente, del quale si vogliono memorizzare il ruolo (per esempio, ricercatore confermato o professore) e la data di presa servizio. A un corso possono offrire supporto uno o più tutor, dei quali si vogliono memorizzare le date di presa e fine servizio. Uno studente (matricola, data immatricolazione) si iscrive a un appello di esame di un certo corso in una certa data. Un appello può essere online o in presenza. L'esame dello studente viene corretto da un tutor, che gli assegna un voto. Lo studente può verbalizzare il voto, nel qual caso occorre registrare la data di verbalizzazione. Dello studente si vogliono memorizzare i crediti acquisiti e la media voti conseguiti. Completati tutti gli esami, lo studente si iscrive a un appello di laurea (anch'esso in un certa data, online o in presenza). Lo studente sceglie un corso come argomento della sua laurea, e un docente come relatore. La laurea dello studente si conclude con un punteggio.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (14/31 punti).

1) Dato il seguente frammento di pseudocodice, se ne calcoli la *complessità ciclomatica*:

```
begin
    read(a);
    read(b);
    while (a<10) AND (b>100) do
        begin
            read(c);
            if (c>0)
                print(c)
            else
                print(-c);
            a:=a+1;
            b:=b-2;
        end
    end
end
```

3

2) Com'è classificata l'*interoperabilità* di un software?

- a. Esterna e di Processo
- b. Interna e di Prodotto
- c. Esterna e di Prodotto**
- d. Interna e di Processo

3) Quali sono le principali caratteristiche del *modello di sviluppo del software a cascata*?

- a. è efficace quando i requisiti di analisi non sono chiari a priori
- b. permette di modificare i risultati delle fasi precedenti alla luce di errori riscontrati a posteriori
- c. genera prototipi funzionanti al termine di ogni fase
- d. si basa su modelli formali del software che vengono fatti evolvere durante lo sviluppo
- e. nessuna di queste**

4) Quale dei seguenti *tipi di interfaccia* è da ritenersi ottimale per applicazioni che debbano gestire dati fortemente strutturati garantendo una buona flessibilità?

- a. interfaccia code-based
- b. interfaccia 3270
- c. pseudo-GUI**
- d. standard GUI
- e. special GUI

5) Un software viene modificato per cambiare la funzione di calcolo degli stipendi a seguito di una riforma fiscale. Di che tipo di *manutenzione* si tratta?

- a. Correttiva
- b. Evolutiva
- c. Perfettiva
- d. Adattiva**

6) Quali delle seguenti classi dovrebbero essere presenti nel *metamodello* di un diagramma dei casi d'uso?

- a. Componente
- b. Dipendenza**
- c. Stereotipo**
- d. Classe**
- e. Associazione**
- f. Attività
- g. Stato
- h. Linea di vita

7) E' dato il seguente frammento di pseudocodice:

```

begin
  read(A, B, C)
  if (A>0) AND (B<0) then
    C := C / (A+B)
end

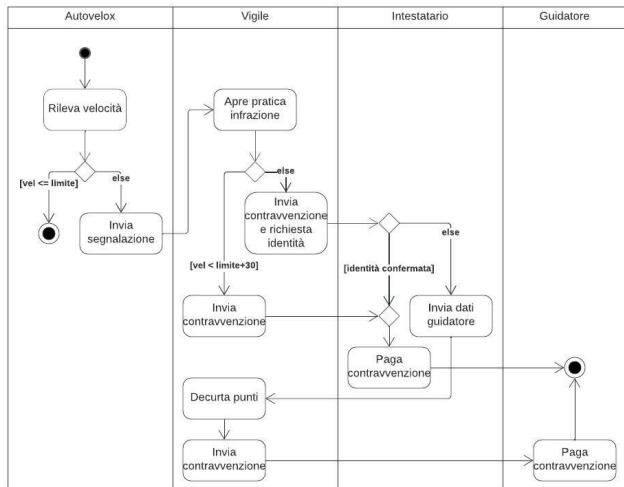
```

Quali dei seguenti insiemi di test soddisfano il *criterio delle decisioni e delle condizioni*?

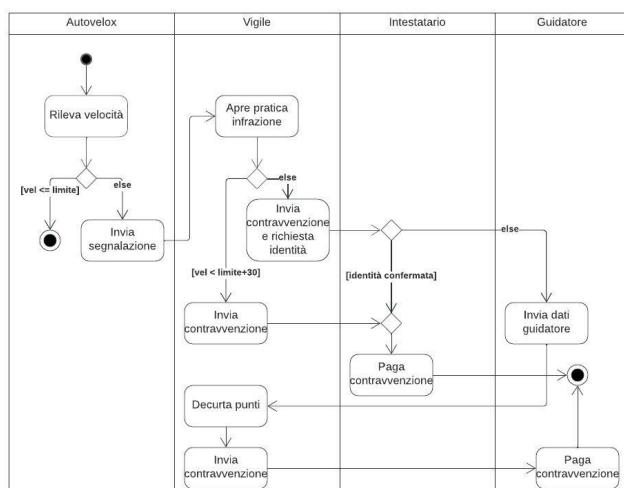
- a. $\{(A=2, B=-1, C=3), (A=-1, B=2, C=0)\}$
- b. $\{(A=2, B=1, C=0), (A=-1, B=-1, C=3)\}$
- c. $\{(A=2, B=-1, C=3)\}$
- d. $\{(A=2, B=-1, C=3), (A=-1, B=-1, C=3)\}$

8) Si vuole modellare il processo di contravvenzione a seguito di infrazioni rilevate da autovelox. L'autovelox rileva la velocità di un veicolo in una certa data e ora e in un certo luogo; qualora la velocità superi il limite, invia all'ufficio vigili urbani una segnalazione corredata dal numero di targa. Il vigile apre la pratica e, qualora il limite non sia stato superato di più di 30 km/h, invia la contravvenzione all'intestatario del veicolo. Altrimenti, è prevista anche una decurtazione di punti. In questo caso, viene inviata anche una richiesta di conferma identità del guidatore. Se l'intestatario non era alla guida, comunica i dati del guidatore a cui viene poi girata la contravvenzione.

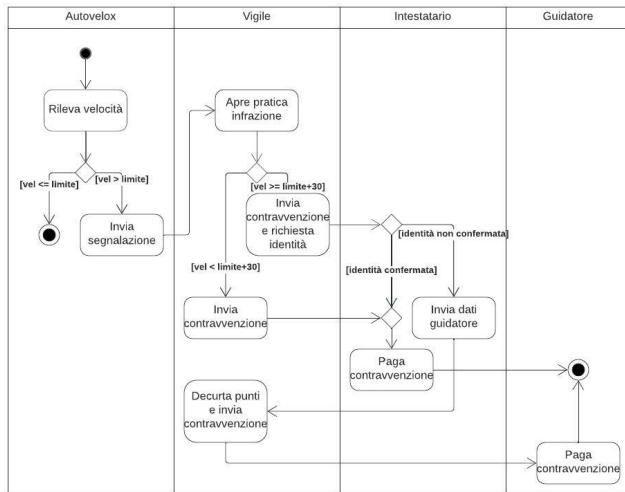
Tra i seguenti, indicare i *diagrammi delle attività* che modellano queste specifiche correttamente.



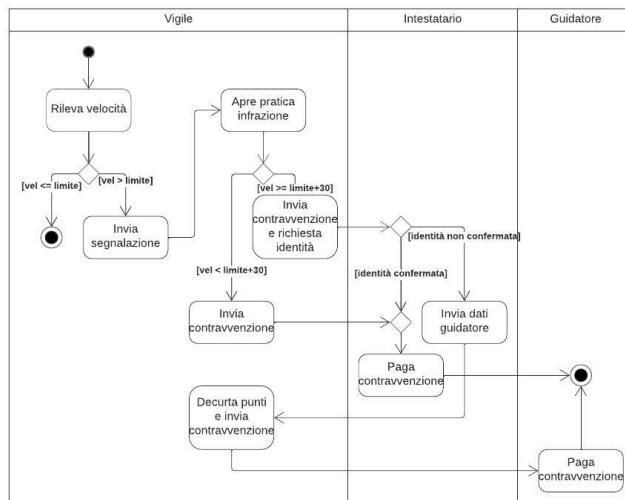
a.



b.



c.



d.

9) IStragram è un social media che permette la pubblicazione di contenuti multimediali e l'interazione tra utenti. Un utente (e-mail, telefono, nome, e cognome) può inviare uno o più Direct (messaggi diretti) ad un qualsiasi altro utente; il sistema memorizza: il timestamp e il testo del messaggio. Un utente può anche pubblicare uno o più contenuti multimediali, nel quale può menzionare uno o più utenti (es: perché compaiono nel contenuto pubblicato). Esistono due tipi di contenuto multimediale: Storia e Post. Della Storia si vuole memorizzare solo il media pubblicato (es: foto, testo, canzone); del Post si vogliono memorizzare i media (si possono pubblicare fino a 10 media nello stesso post) e la descrizione del post. Inoltre, un utente può rispondere alle storie e commentare i post. Di entrambe queste interazioni si vogliono memorizzare: il timestamp e il testo in oggetto. Un commento ad un post può specificare di star rispondendo ad un altro commento dello stesso post. Infine, un utente può mettere like a qualsiasi contenuto multimediale. Del post si vogliono memorizzare il numero di like complessivi ricevuti.

Si modellino le specifiche sopra riportate in UML attraverso un *diagramma delle classi* (14/33 punti).