

Ingegneria del Software a.a. 2012-13

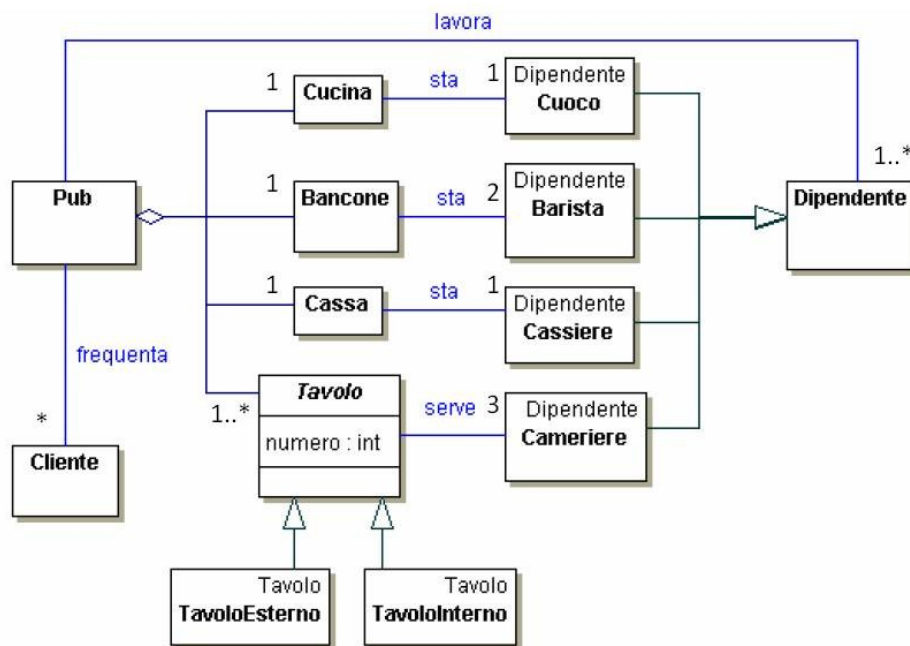
Prova Scritta del 14 gennaio 2013

Esercizio 1

Un **pub**, frequentato da clienti, è composto da una cucina, un bancone, una cassa e da diversi tavoli che possono essere all'interno o all'esterno del pub. Per il pub lavorano diversi dipendenti. In particolare: un cuoco (che lavora in cucina), due baristi (che lavorano al bancone), un cassiere (che gestisce la cassa) e tre camerieri che servono ai tavoli.

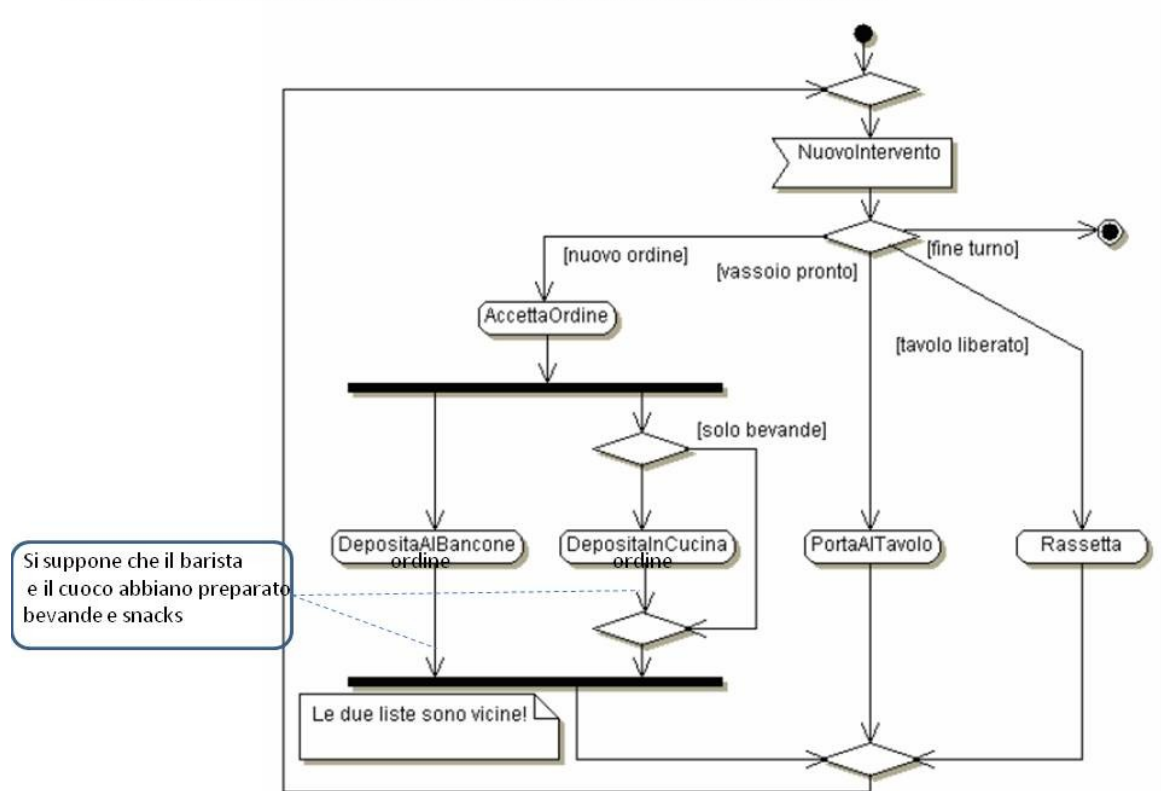
Il servizio ai tavoli prevede la raccolta delle ordinazioni da parte dei camerieri e la successiva consegna delle bevande e degli snack ordinati. Raccolta l'ordinazione, il cameriere la consegna al bar e in cucina (quasi contemporaneamente). Uno dei baristi prende un'ordinazione dalla lista di quelle da servire (rispettando l'ordine temporale di consegna), prepara le bevande indicate su un vassoio che appoggia sul bancone, a disposizione del cameriere. Lo stesso accade in cucina per gli snack, preparati dal cuoco. Il cameriere quando i vassoi sono pronti preleva le bevande e gli snack e li consegna ai clienti. Quando i clienti finiscono di consumare, il cameriere raschetta e pulisce il tavolo. A fine turno il cameriere termina il lavoro e va a casa.

a) Disegnare un diagramma delle classi relativo alla struttura del pub



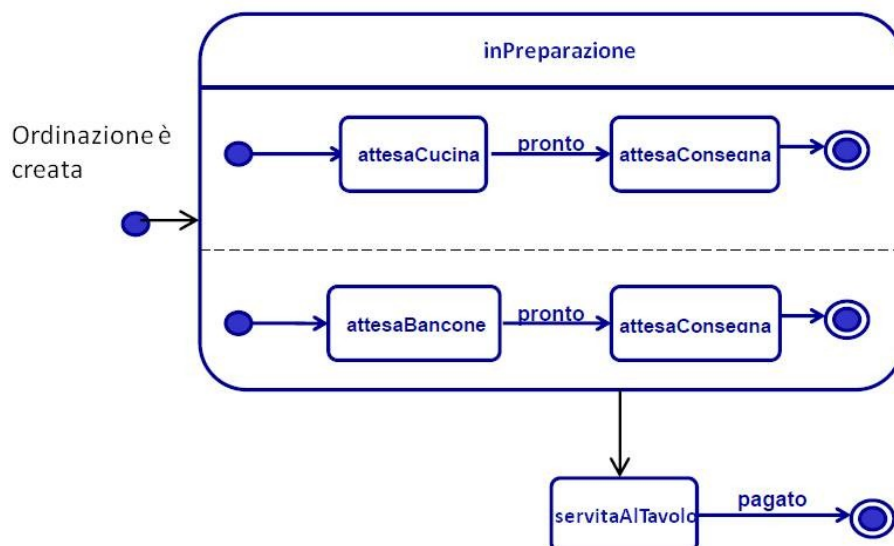
b) Disegnare un diagramma delle attività relativo al lavoro del cameriere così come descritto sopra

Activity diagram **solo** del cameriere (così come richiesto dall'esercizio)



c) Disegnare un diagramma di stato che rappresenti l'oggetto Ordinazione

Attenzione: siccome siamo a livello di dominio con ordinazione NON si intende SOLO la lista di bevande e snacks scritti sul blocchetto del cameriere MA proprio l'ordinazione



Volendo si potevano aggiungere ancora due stati: inRaccolta (prima di inPreparazione) e inConsegna (prima di servito)

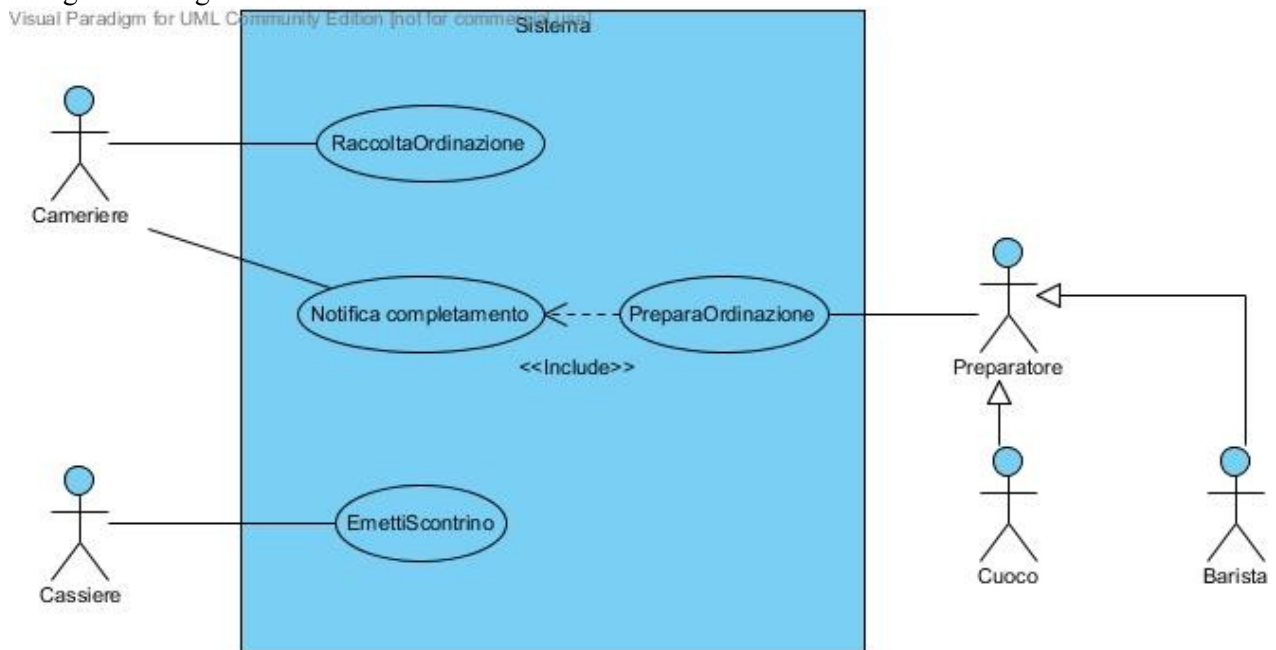
Esercizio 2

Il proprietario di un pub ha deciso di introdurre un sistema software per la raccolta e la gestione delle ordinazioni dei clienti, per migliorare il servizio diminuendo i tempi di attesa, e per semplificare le interazioni tra i dipendenti. Ogni cameriere è dotato di un palmare collegato senza fili al sistema di raccolta e gestione delle ordinazioni. Mentre cuoco, baristi e cassiere utilizzano un PC (Se l'esercizio 1 non è stato ancora svolto leggere almeno il testo prima di procedere con la risoluzione dell'esercizio 2)

Si considerino le seguenti funzionalità del sistema per la raccolta e la gestione delle ordinazioni:

- il sistema riceve le ordinazioni dei clienti raccolte dal cameriere;
- il sistema comunica al cuoco e ai baristi le ordinazioni ed è informato della loro preparazione;
- il cuoco e i baristi preparano le ordinazioni e quando hanno terminato avvertono il sistema che notifica al cameriere che l'ordinazione è pronta;
- il sistema riceve dal cassiere la richiesta del conto di un tavolo, emette lo scontrino e gestisce il relativo pagamento (per semplicità solo in contanti)

a) Disegnare il diagramma dei casi d'uso



b) Specificare, usando lo stile visto a lezione, i casi d'uso relativi alla: registrazione delle ordinazioni al tavolo e preparazione dell'ordinazione e notifica

I casi d'uso sono presentati ad un alto livello di astrazione.

Caso d'uso: RaccoltaOrdinazione

Descrizione: Il cameriere registra nel sistema una nuova ordinazione al tavolo

Attori primari: cameriere

Attori secondari: nessuno

Livello: User-goal

Sequenza eventi principale:

1. Il cameriere crea una nuova ordinazione impostando il numero del tavolo
2. Il sistema crea la nuova ordinazione
3. **Fintantochè** i clienti richiedono bevande o snacks:
 - 3.1 Il cameriere seleziona la corrispettiva voce sul palmare ed inserisce la quantità richiesta aggiungendo così la consumazione all'ordinazione
4. Il sistema mostra il riepilogo ed attende conferma

5. Il cameriere conferma (richiedendo così la preparazione dell'ordinazione)
6. Il sistema aggiorna lo stato dell'ordinazione ad "accettata" ed avverte i preparatori che c'è una nuova ordinazione

Sequenza eventi alternativa: *AnnullaVoce, AnnullaInteraOrdinazione, NonConfermaOrdinazione*

Caso d'uso: *PreparaOrdinazione*

Descrizione: Il preparatore (barista o cuoco) prepara il vassoio relativo ad un ordinazione

Attori primari: preparatore

Attori secondari: nessuno

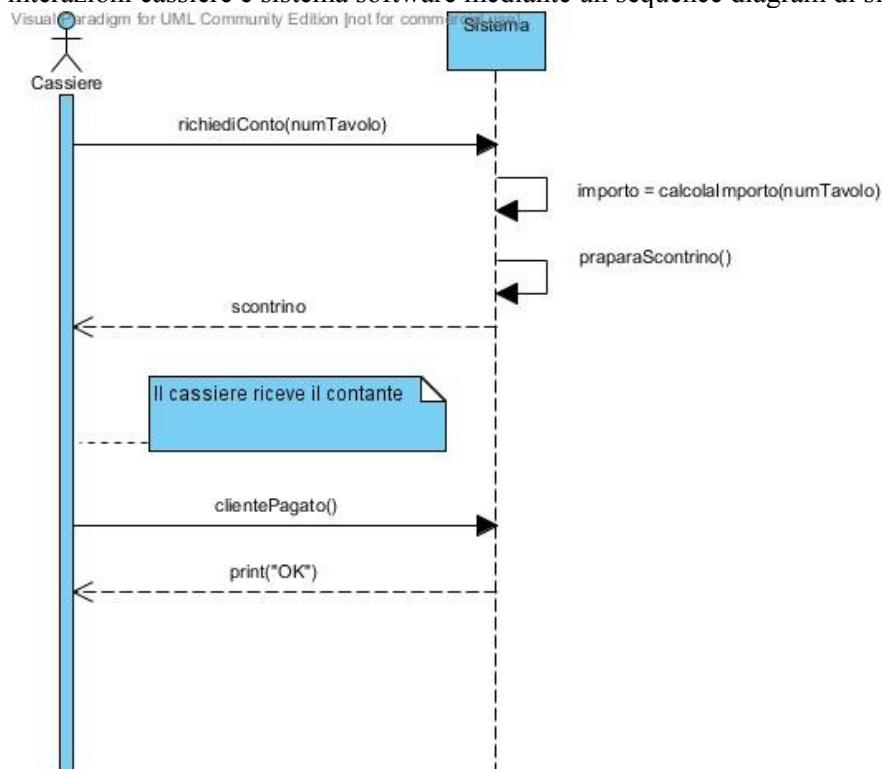
Livello: user-goal

Sequenza eventi principale:

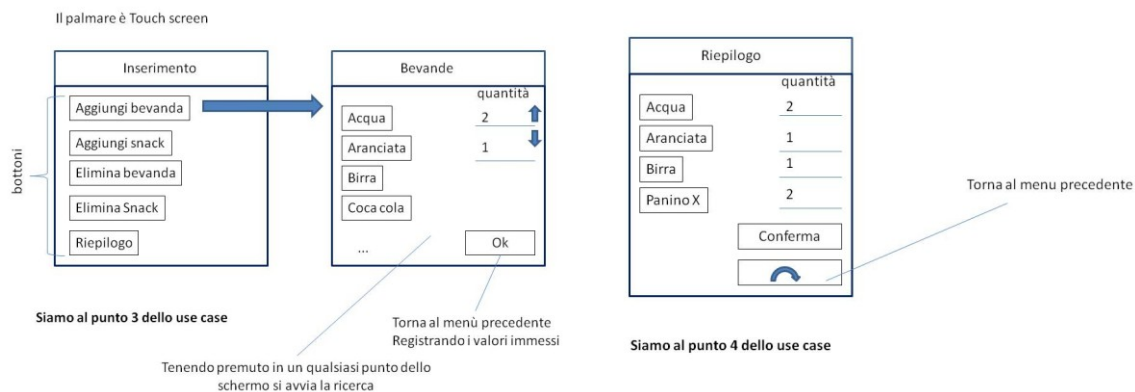
1. Il preparatore richiede al sistema la prima ordinazione da completare presente nella lista
2. Il sistema mostra la prima ordinazione
3. **Perogni** consumazione (bevanda o snack) nell'ordinazione:
 - 3.1 Il preparatore la segna come pronta non appena preparata
4. Il preparatore registra il vassoio come pronto e disponibile
5. Il sistema avverte il cameriere che c'è una nuova ordinazione pronta da consegnare
include(NotificaCompletamento)

Sequenza eventi alternativa: *consumazioneNonDisponibile (cioè manca un ingrediente o bevanda)*

c) Specificare le interazioni cassiere e sistema software mediante un sequence diagram di sistema



d) (facoltativo) Arricchire il caso d'uso "registrazione delle ordinazioni al tavolo" con degli screen mockup che ne chiariscano il funzionamento



Esercizio 3

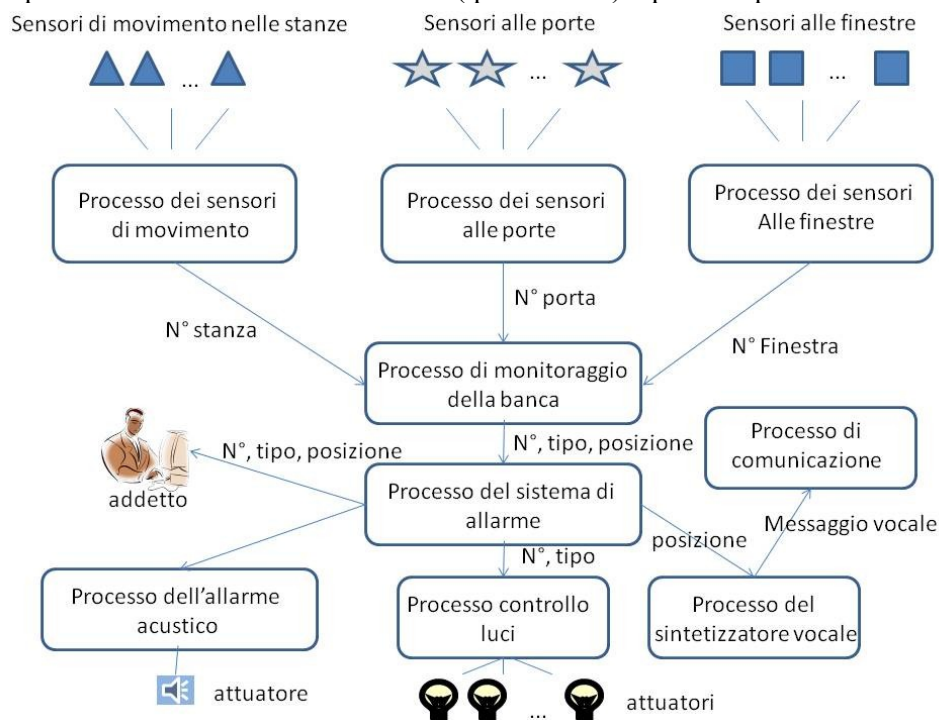
Si supponga di dover sviluppare un sistema software per controllare un sistema d'allarme antifurto da installare in una banca. Questo sistema utilizza diversi tipi di sensori, quali sensori di movimento nelle stanze, sensori alle porte per rilevare l'apertura delle porte del corridoio (ma anche sensore alla porta d'ingresso) e sensori alle finestre. Quando un sensore individua la presenza di un intruso il sistema software:

- (1) chiama automaticamente la polizia utilizzando un sintetizzatore vocale e riferisce la posizione dell'allarme;
- (2) accende le luci nelle stanze attorno al sensore attivo;
- (3) fa partire un allarme acustico;
- (4) comunica l'evento ad un pannello di controllo remoto che è controllato da un addetto

- a) Specificare il nome dello stile architetturale che ritenete più appropriato per il sistema software sopra descritto

Sistema di monitoraggio e controllo (Manager model). E' un sistema real-time!

- b) Disegnare un diagramma che descrive l'architettura del sistema in accordo a tale stile, commentando brevemente a parole il funzionamento e marcando (quando utile) input e output dei vari componenti



- c) (facoltativo) Descrivere in pseudocodice o con un activity diagram la componente del sistema che si occupa di monitorare i sensori e che fornisce/manda alle altre componenti la posizione di dove è avvenuto l'allarme

Class BankMonitor {

```
ProcessoSensoriMovimento SM = new ProcessoSensoriMovimento();
ProcessoSensoriFinestre SF = new ProcessoSensoriFinestre();
ProcessoSensoriPorte SP = new ProcessoSensoriPorte();
ProcessoSistemaAllarme Allarme= new PorcessoSistemaAllarme();
```

BankMonitor() {...} // inizializza tutti i sensori e avvia i processi

MonitoraggioProcessoSensori() {

Int mossa, finestra, porta, numero;

String posizione, tipo;

While (true) {

mossa=SM.getValore(); //interroga I sensori ogni "x" secondi (0 vuole dire tutto

OK)

finestra= SF.getValore();

porta = SP.getValore();

if (mossa<>0 or finestra<>0 or porta<>0) { // un sensore ha individuato un

intruso

posizione= calcolaPosizione(mossa, finestra, porta);

tipo = calcolaTipo(mossa, finestra, porta);

numero=mossa+finestra+porta; // si suppone che solo un sensore possa

// essere attivo ad un certo istante

allarme.pericolo(numero, tipo, posizione);

break;

}

}

}

calcolaPosizione(...) {...}

calcolaTipo(...) {...}

}