

Cognome

Nome

Matricola

Note:

- Tutti i diagrammi disegnati devono utilizzare la sintassi del linguaggio UML 2.x ed essere opportunamente commentati. Il foglio protocollo va utilizzato solo per la brutta copia. Per disegnare i diagrammi definitivi utilizzare gli spazi liberi delle pagine 1,2 e 3 del presente testo, usando la penna e non la matita.
- Qualora si utilizzasse anche un foglio protocollo per i diagrammi definitivi, indicarlo nel presente foglio.
- Riportare il nome, cognome e matricola, su tutti i fogli consegnati.

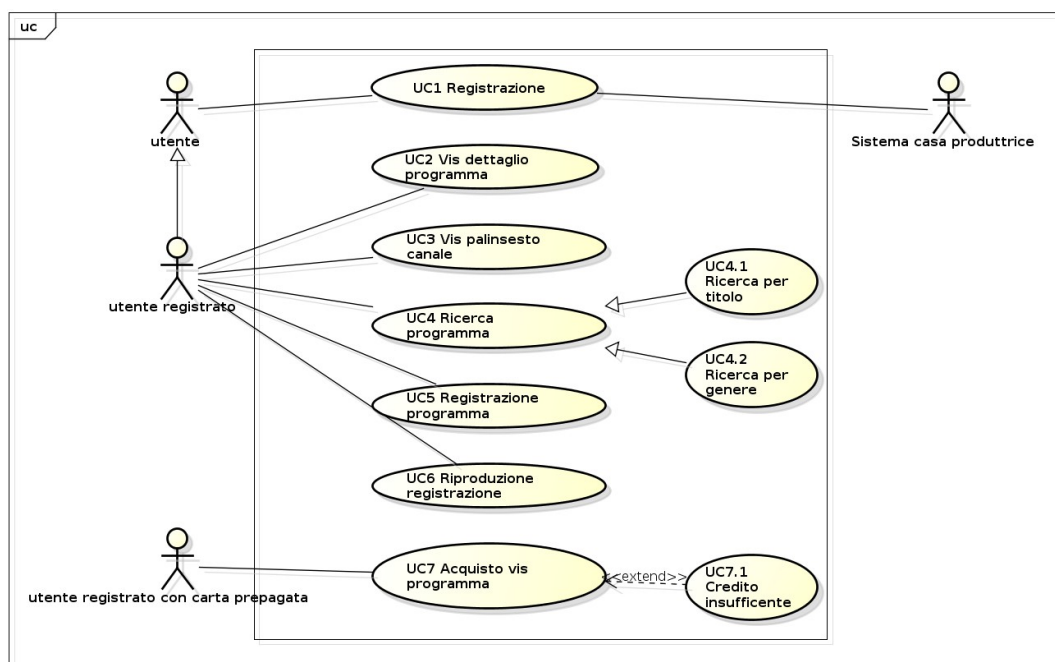
### Punto 6/30

Gli ultimi modelli di televisori presenti sul mercato contengono la proprio interno un vero e propriosistema operativo. L'utente può accedere alle funzionalità disponibili attraverso un'apposita interfaccia. Per farlo, è necessario prima di tutto registrare la propria televisione presso i sistemi della casa produttrice. La registrazione prevede l'inserimento del codice identificativo dalla TV e la data di acquisto riportata sullo scontrino o sulla fattura. Se il codice inserito non è corretto, viene visualizzato un messaggio di errore e l'utente non può completare l'abilitazione delle funzionalità interattive.

Successivamente, il televisore permette l'accesso alle seguenti funzionalità. Un utente può visualizzare per ogni programma in riproduzione una serie di informazioni, fra le quali l'ora di inizio/fine del programma, una sua sinossi e un giudizio, ossia una votazione da 1 a 5 stelle. L'utente può inoltre visualizzare il palinsesto di giornata per ogni canale, rappresentato da una lista di programmi. In tale lista, per ogni programma, sono presenti il nome dello stesso e l'ora di inizio e fine. E' possibile inoltre ricercare un programma: esistono due tipologie di ricerca, quella per titolo e quella per genere. Ogni programma può essere registrato nella memoria interna della televisione e rivisto in un momento successivo.

Per ragioni di copyright, la riproduzione di ogni programma registrato può essere solamente fermata o riavviata, ma non può essere riavvolta in alcun modo. Un utente che disponga di una scheda prepagata può inoltre comprare la visualizzazione di un programma televisivo. Se il credito della scheda fosse insufficiente, viene visualizzato uno specifico messaggio di errore e l'utente viene rediretto alla funzionalità di visualizzazione del palinsesto.

Si utilizzino i diagrammi dei casi d'uso per modellare gli scenari descritti. Non è richiesta la descrizione testuale dei casi d'uso individuati.



uc

### UC1 - Registrazione



UC1.1 Ins.codice  
Identificativo tv

UC1.2 Ins.data di  
acquisto tv

UC1.3 Codice errato:  
registrazione fallita

UML Use Case Diagram for UC1 - Registrazione. Actor 'utente' is connected to use cases UC1.1, UC1.2, and UC1.3. UC1.1 and UC1.2 are connected to UC1.3 via an 'extend' relationship.

uc

### UC2 - Vis.dettagli programma



UC2.1 Vis.ora di inizio

UC2.2 Vis.ora di fine

UC2.3 Vis. sinossi

UC2.4 Vis.giudizio

UML Use Case Diagram for UC2 - Vis.dettagli programma. Actor 'utente registrato' is connected to use cases UC2.1, UC2.2, UC2.3, and UC2.4.

uc

### UC3 - Vis.palinstesto canale



UC3.1 Vis.nomi programmi

UC3.2 Vis orari inizio programmi

UC3.3 Vis.orari di fine programmi

UML Use Case Diagram for UC3 - Vis.palinstesto canale. Actor 'utente registrato' is connected to use cases UC3.1, UC3.2, and UC3.3.

uc

### UC6 - Riproduzione registrazione



UC6.1 Avvio riproduzione

UC6.2 Stop riproduzione

UC6.3 Ripresa riproduzione

UC6.4  
Riproduzione già  
avviata una volta

UML Use Case Diagram for UC6 - Riproduzione registrazione. Actor 'utente registrato' is connected to use cases UC6.1, UC6.2, UC6.3, and UC6.4. UC6.1 and UC6.4 are connected via an 'extend' relationship.

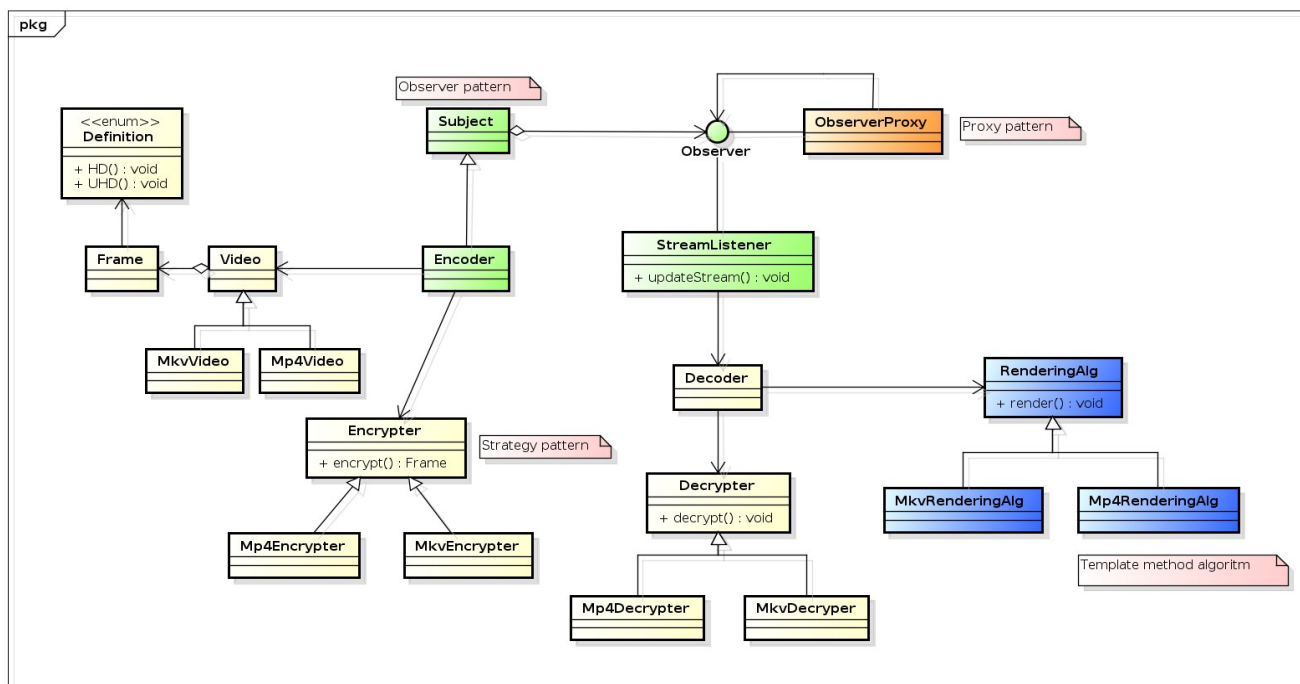
Cognome

Nome

Matricola

### Punto 7/30

Negli ultimi anni l'interesse verso lo stile di programmazione noto come reactive programming è considerevolmente cresciuta. Esso sfrutta in particolar modo il pattern Observer: una risorsa resta in ascolto su un'altra risorsa ed esegue operazioni solamente in risposta al verificarsi di certi eventi. Questo tipo di programmazione ha alcune caratteristiche positive, quali un'oculata gestione delle risorse e uno spiccato livello di resilienza. Uno dei principali promotori di questo stile di programmazione è Netflix. Ad esempio, il software sul decoder di Netflix è organizzato come un'architettura distribuita che utilizza una Observer per gestire lo streaming di un video. Una programma televisivo è quindi fatto da una serie di fotogrammi (o immagini). Ogni qualvolta un'immagine è aggiunta al programma, questo notifica il cambiamento a un'opportuna componente in ascolto. La sorgente del programma e la componente osservatore sono dislocate su macchine differenti, ma interagiscono fra loro come se condividessero lo stesso contesto di esecuzione. Ovviamente, le informazioni non vengono trasmesse in chiaro, ma vengono crittate e decrittate da oggetti specializzati. Ogniqualvolta una nuova immagine, ossia un array di byte, arriva al decoder, questa viene elaborata da un algoritmo di trasformazione specifico per il formato del programma in trasmissione. MP4 e MKV sono tra i formati supportati. Il prodotto della conversione è sempre un oggetto Frame. Esistono due tipi di frame: quello in high definition (HD), e quello in ultra high definition (UHD). La componente che visualizza le immagini sullo schermo, applica una funzione di rendering che differisce solo in parte sulla base del formato, ma che possiede numerosi parti in comune fra le immagini in MP4 e in MKV.



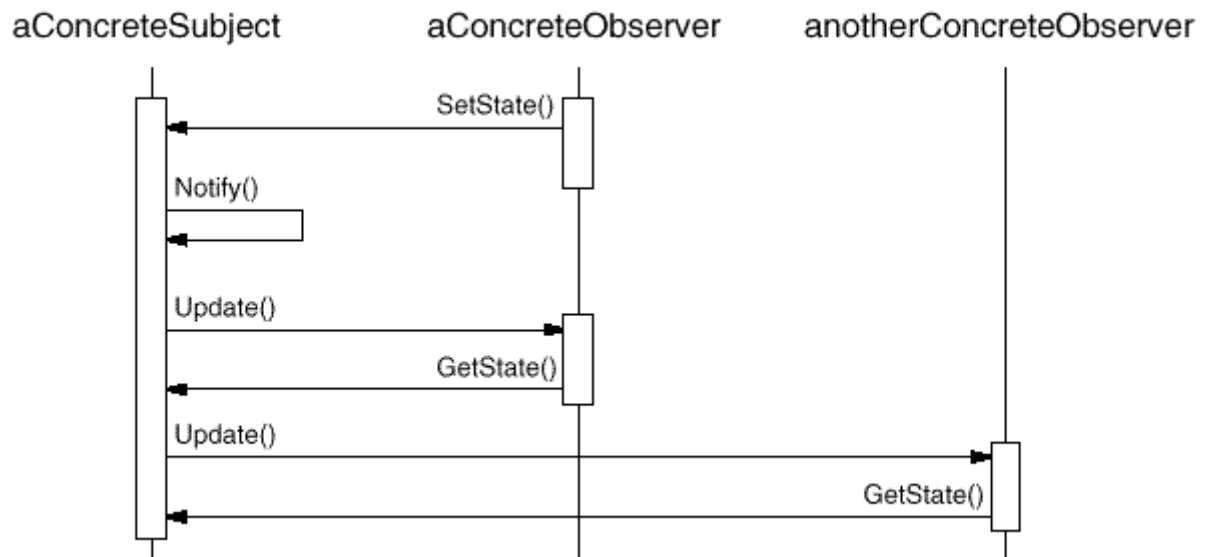
Cognome

Nome

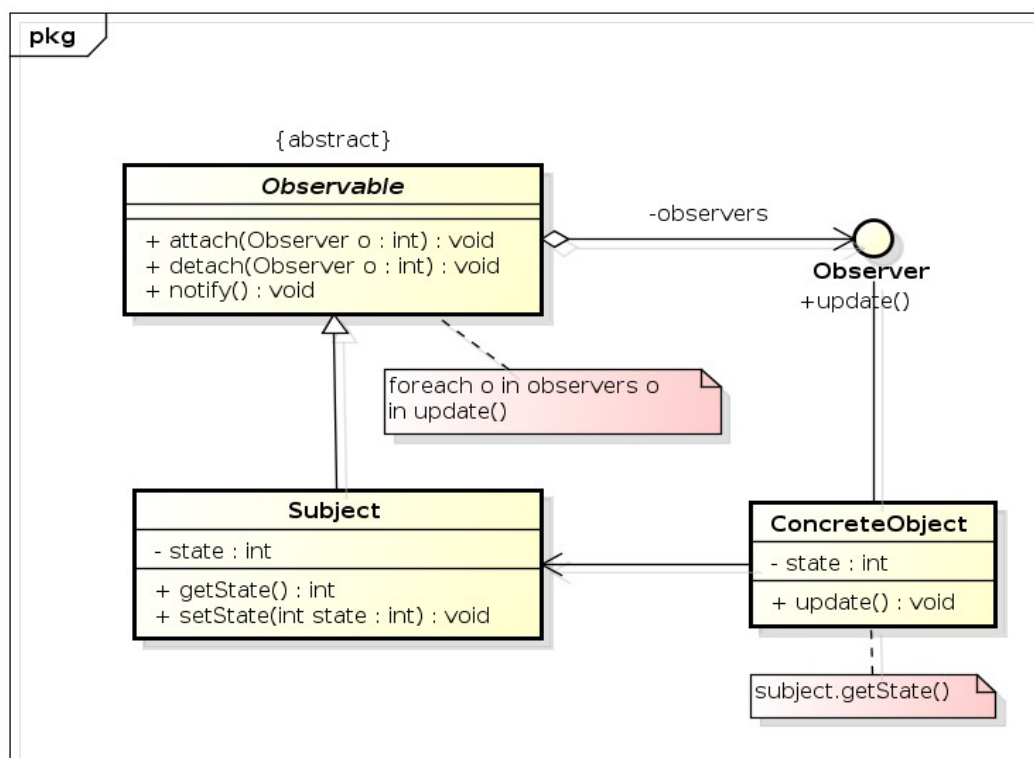
Matricola

**Punto 3/30**

Dato il seguente diagramma di sequenza:



si fornisca il diagramma delle classi del design pattern corrispondente. In particolare, si presti molta attenzione ai metodi esposti da ciascun tipo.



Cognome

Nome

Matricola

---

**Punti 4/30 - Totalizzati 3 punti**

*Facendo riferimento allo standard ISO/IEC 12207, discutere la differenza di obiettivi, attività coinvolte, strategie di conduzione e strumenti, tra i processi di verifica e validazione.*

La verifica viene effettuata sui prodotti di ogni singola attività, è di interesse solamente interno all'azienda. Viene gestita ed effettuata dai verificatori di progetto e consiste nel controllo sistematico, disciplinato e misurabile dei prodotti di ogni attività (come documenti e/o codice). Tale controllo può essere effettuato a "pettine" (walthrough) con impiego di tempo e costi maggiori (ma necessari se l'azienda non ha ancora esperienza sufficiente) o a "campione" (inspection), controllando cioè solo le parti più critiche (ovvero più inclini ad errori) del prodotto.

Il ciclo PDCA è uno dei metodi per applicare la verifica, in quanto comporta controlli frequenti e cicli, con conseguente miglioramento del prodotto che sia rispondente nelle aspettative iniziali del cliente. Tale processo si svolge controllando che ogni requisito formulato in analisi sia stato effettivamente e completamente soddisfatto. Per fare ciò, una componente fondamentale la svolgono i test funzionali, che appunto verificano le funzionalità del prodotto e il collaudo del cliente. Per facilitare questo processo è essenziale il tracciamento dei requisiti verso le componenti del prodotto.

---

**Punti 5/30 - Totalizzati 5 punti**

*Presentare, per obiettivi, critici di valorizzazione, possibilità di automazione, due metriche significative per la misurazione della qualità della progettazione software e del codice (quindi almeno una metrica per ciascun ambito). Ove possibile, fare riferimento a esperienze specifiche e personali per valutare l'esito osservato dall'eventuale uso pratico delle metriche discusse.*

Una metrica per la progettazione software è l'instabilità, che indica il rapporto tra coesione e accoppiamento di una componente. In coesione indica quanto le parti interne della componente siano legate tra loro. Una coesione alta è indice di una componente modulare, compatta e specializzata. L'accoppiamento indica invece quante dipendenze la componente ha con l'esterno. Maggiore è l'accoppiamento, minore è la mantenibilità e la modularità della componente. Un valore basso di instabilità indica una forte coesione e uno scarso accoppiamento, mentre un valore alto è sintomo di un accoppiamento troppo forte. L'obiettivo di tale metrica è di rendere il software più modulare e manutenibile possibile. Una metrica per il codice è la complessità ciclomatica, che indica il numero di cammini indipendenti che l'esecuzione di un metodo può intraprendere. Un valore alto è sintomo di un metodo troppo complesso, scarsamente modulare e manutenibile. Nel progetto didattico è stata utilizzata questa metrica, che ci ha permesso di rendere i metodi più modulari e facili da testare.

---

**Punti 5/30 - Totalizzati 3++ punti**

*Inquadrare la pratica nota come "continuous integration" nel dominio dell'ingegneria del software e illustrare concisamente alcuni dei metodi e degli strumenti che consentono di attuarla. Ove possibile, rapportare tali considerazioni all'esperienza personale guadagnata nella loro applicazione.*

La continuous integration o integrazione continua si applica quando ci sono sistemi di versionamento del codice e si vuole mantenere delle versioni stabili nel repository. Inoltre nei sistemi di building automatico si vuole accertare che alcune modifiche non vadano a corrompere il codice funzionante.

In particolare la continuous integration è utilizzata nella fase di test. I test possono essere:

- Test di sistema (effettua il test del sistema nel suo insieme, si effettua su più unità)
- Test di unità (effettua il test di un insieme di componenti)
- Test componente (effettua il test su singole porzioni di codice)
- Test integrazione (effettua il test che assembla le vari parti per vedere che tutto funzioni correttamente)

In particolare "Jenkins" è uno degli strumenti che permette di attuare continuous integration oppure "maven" che abbiamo utilizzato durante il progetto di build del software della piattaforma scritta in Spring.

La continuous integration è una pratica che ha da pochi anni iniziato a prendere piede e tuttavia è una pratica che fatica ad essere attuata nelle aziende in quanto spesso si tende a sfiorare i tempi di pianificazione e vengono eliminati i tempi di test e di controllo del codice e dei documenti.