

# Ingegneria del Software A.A. 2019/2020

## Esame 2020-04-21

---

### Esercizio 1 (4 punti)

#### Descrizione

Amazon è un sistema distribuito di enorme complessità. Per mantenere alta la qualità dell'esperienza utente, il sistema si basa su un'architettura *reactive*, costruita completamente sul *pattern publish – subscribe*. Apprezzato dagli utenti è il sistema di notifiche sullo stato degli ordini. Ogni qualvolta un pacco parte alla volta di un utente finale, il corriere inserisce una notifica nel sistema di *tracking*. Questo avvisa il sistema di notifica, posizionato in un altro nodo della rete. Le notifiche possibili sono diverse: e-mail, messaggio SMS e notifica sull'applicazione *mobile* di Amazon. Sulla base delle impostazioni del profilo dell'utente, il sistema decide quali media utilizzare. Per permettere al sistema di evolvere semplicemente, l'invio delle notifiche avviene utilizzando un'interfaccia unica.

Si modelli tale sistema mediante un diagramma delle classi, comprensivo dei *design pattern* a esso pertinenti.

### Esercizio 2 (3 punti)

#### Descrizione

Dato il sistema di notifiche precedentemente descritto, si descriva utilizzando un opportuno diagramma di sequenza, la collaborazione delle componenti per l'invio di una notifica e-mail ad un utente a fronte dell'invio di un pacco.

### Esercizio 3 (3 punti)

#### Descrizione

Il sito di Amazon permette di effettuare una ricerca avanzata dei prodotti disponibili a catalogo selezionando sia il fatto che sia il prodotto disponga della spedizione *Prime*, sia la media delle recensioni positive dei clienti (1 ... 5 stelle), sia se per il prodotto è disponibile la spedizione internazionale. Il risultato della ricerca è una lista di prodotti, che può essere ordinata per prezzo crescente o per "ultimi arrivi".

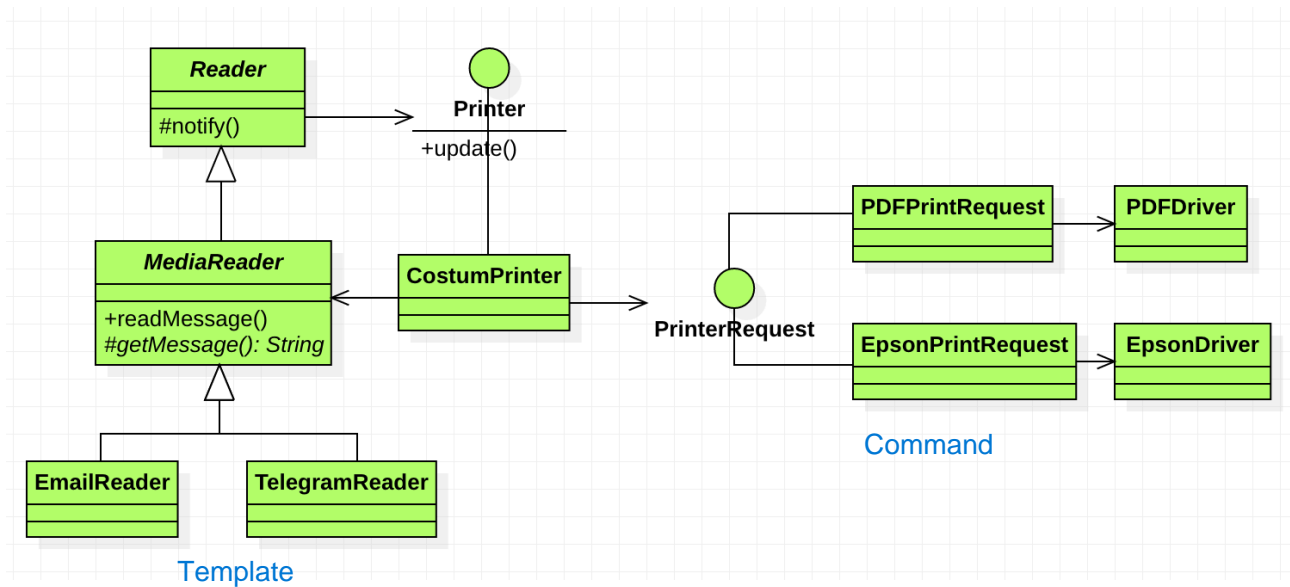
Si utilizzino i diagrammi dei casi d'uso per modellare gli scenari sopra descritti. Non ne è richiesta la descrizione testuale.

### Esercizio 4 (2 punti)

#### Descrizione

Sia dato il seguente diagramma delle classi.

## Observer



Quali pattern possono essere individuati al suo interno?

- 1) **Observer pattern**
- 2) Strategy pattern
- 3) Proxy pattern
- 4) **Command pattern**
- 5) Abstract Factory pattern
- 6) **Template Method pattern**