

Ingegneria del Software con Laboratorio (12 crediti) – Programma a.a. 2010-11

Prova Scritta del 13 luglio 2012

Tempo a disposizione: 3 ore (Solo parte generale: 1 ora e 30', solo UML: 1 ora e 30').

La parte generale oltre ai 2 esercizi prevede 1 domanda che verrà distribuita 30' prima della scadenza per la consegna – da quel momento non sarà più consentita la consultazione del materiale.

Svolgere parte UML e parte generale su fogli separati.

Esercizio 1 – Parte UML

Si supponga di voler sviluppare un sistema che migliori il servizio offerto dal distributore automatico di bibite e merendine disponibile al quinto piano. Il nuovo sistema sarà formato da un gruppo di distributori automatici, installati sempre al quinto piano, collegati tra loro in modo da offrire un servizio integrato. Gli utenti potranno accedere ad uno qualsiasi dei distributori mediante un codice personale che permette loro di usufruire del credito disponibile per l'acquisto dei prodotti. Le novità del sistema sono costituite dall'elaborazione delle preferenze degli utenti sulla base dei loro acquisti e dalla possibilità che ha l'utente di prendere visione, da un qualsiasi distributore, oltre che dei prodotti disponibili nel magazzino locale del distributore, anche di altri prodotti del catalogo dei prodotti in vendita (ad esempio, in distributori a fisica o a chimica gestiti dallo stesso sistema). In particolare, si visualizzano, usando le informazioni sulle preferenze dell'utente, i primi tre prodotti che non sono disponibili in locale ma sono disponibili su altri distributori. L'utente interessato a un prodotto non disponibile localmente, perché temporaneamente esaurito, può sapere quale distributore può soddisfare la sua richiesta.

La procedura di acquisto è molto semplice: ogni prodotto è caratterizzato da un nome (evocativo del prodotto stesso), da un codice (di solito una sequenza di cifre numeriche) e da un costo; dopo essersi autenticato mediante il codice personale, l'utente può scegliere uno o più prodotti tra quelli presenti nel magazzino locale e acquistarli sulla base del credito disponibile. Tale credito è automaticamente aggiornato al termine della procedura di acquisto.

Tra le funzioni a disposizione del gestore del sistema, ricordiamo la possibilità di gestire i magazzini locali ed eventualmente di costituire un magazzino centralizzato per le scorte, e le funzioni standard per aggiungere e rimuovere utenti, prodotti, distributori.

- a) Identificare gli attori, distinguendo in attori primari e secondari.
- b) Identificare i casi d'uso del sistema e produrre un diagramma dei casi d'uso.
- c) Specificare il caso d'uso corrispondente all'operazione di acquisto.

Esercizio 2 – Parte UML

Consideriamo una macchina Self-Service che distribuisce prodotti alimentari (una versione molto più semplificata rispetto a quella dell'Esercizio 1). La macchina è composta da tre oggetti:

- La Parte Frontale – cioè l'interfaccia che la macchina presenta all'utente.
- La Cassetta delle Monete - la parte in cui vengono accumulate le monete e che gestisce i vari controlli della macchina (ad esempio che i soldi sono sufficienti).
- Il Contenitore dei Prodotti - La parte che contiene gli alimenti che vengono acquistati dal cliente.

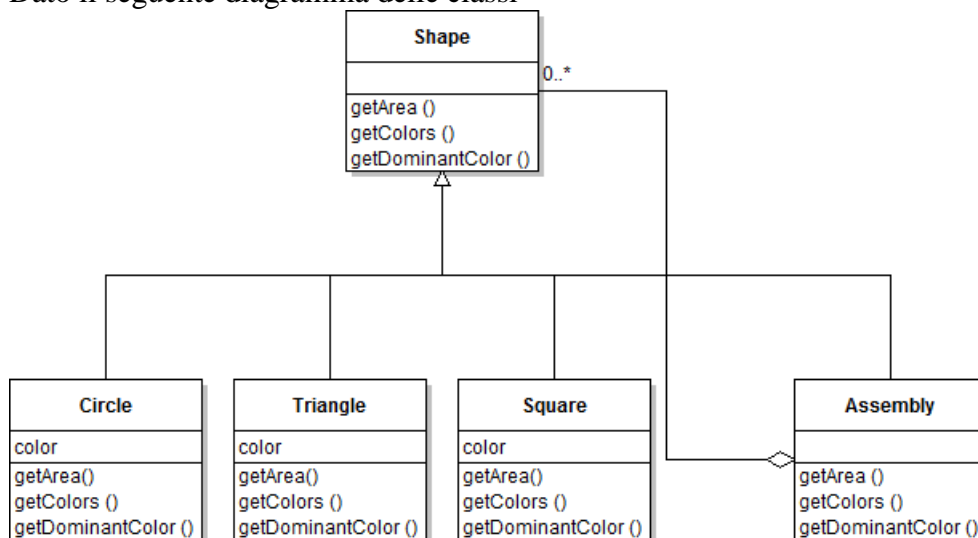
a) Modellare un class diagram UML che riporta le operazioni e uno o più sequence diagram UML che rappresentano la seguente sequenza di azioni:

- Il cliente inserisce le monete nella macchina
- Il dispositivo di controllo verifica che l'ammontare sia quello richiesto (considerare entrambi i casi)
- Il cliente esegue la selezione del prodotto desiderato
- Le monete arrivano nella Cassetta delle Monete
- Il dispositivo di controllo della Cassetta delle Monete verifica se il prodotto desiderato è presente nel Contenitore dei Prodotti (considerare entrambi i casi)
- La Cassetta delle Monete aggiorna la sua riserva di monete
- Il dispositivo di controllo della Cassetta delle Monete informa il Contenitore dei Prodotti che può espellere il prodotto desiderato dalla Parte Frontale della macchina (se presente)

b) Scrivere in pseudo-codice una possibile implementazione del/dei sequence diagram progettato/i al punto precedente (come fatto a lezione, vedere parte in verde sulle slide relative alla lezione sui sequence diagram).

Esercizio 3 – Parte generale

Dato il seguente diagramma delle classi



in cui Shape rappresenta una classe astratta corrispondente a forme generiche che possono essere assemblate a partire da forme “di base”. Le forme “di base” hanno un unico colore (che coincide con il colore dominante), mentre le figure “assemblate” possono avere componenti di diversi colori e il loro colore dominante è il colore cui corrisponde la maggiore area.

- Identificare, se possibile, nel diagramma precedente uno dei Design Pattern visti, mettendo in relazione le classi del diagramma con quelle del Design Pattern originario.
- Proporre un progetto alternativo basato sull'utilizzo di uno dei Design Pattern visti, (diverso da quello eventualmente identificato al punto precedente). Produrre almeno il diagramma delle classi del progetto e motivare la scelta effettuata.
- Confrontare le due organizzazioni di classi in termini di coupling e cohesion.

Esercizio 4 – Parte generale

Dato il seguente codice Java:

```

// Questo programma sceglie una carta da gioco a caso e stampa la carta scelta.
public class ScegliCarta {
    public static void main(String[] arg) {
        // scelgo il valore di una carta:
        int val = scegliValoreCarta();

        // scelgo il seme della carta:
        int sm = scegliSemeCarta();

        // stampo la carta scelta
        stampaCarta(val, sm);
    }

    // questo metodo sceglie il valore di una carta da gioco
    // (un intero tra 1 e 13)
    public static int scegliValoreCarta() {
        int valore = (int) (Math.random()*12 + 1);
        return valore;
    }

    // questo metodo sceglie il seme di una carta da gioco (un intero tra 0 e 3)
    // 0 = cuori, 1 = quadri, 2 = fiori, 3 = picche)
    public static int scegliSemeCarta() {
        int sm = (int) (Math.random()*4);
        return sm;
    }

    // questo metodo si occupa di stampare una carta di un mazzo.
    // In input deve ricevere il valore della carta e il seme.
    public static void stampaCarta(int valore, int seme) {
        System.out.println();
        System.out.println("\t *****");
        System.out.println("\t*          *");
        if (valore == 1)
            System.out.println("\t*   asso   *");
        else if (valore == 11)
            System.out.println("\t*   fante  *");
        else if (valore == 12)
            System.out.println("\t*   regina *");
        else if (valore == 13)
            System.out.println("\t*    re    *");
        else if (valore == 10)
            System.out.println("\t*    10    *");
        else
            System.out.println("\t*      " + valore + "      *");
        System.out.println("\t*          *");
        System.out.println("\t*   di     *");
        System.out.println("\t*          *");
        System.out.println("\t*          *");
        if (seme == 0)
            System.out.println("\t*   cuori   *");
        else if (seme == 1)
            System.out.println("\t*   quadri  *");
        else if (seme == 2)
            System.out.println("\t*   fiori   *");
        else
            System.out.println("\t*   picche  *");
        System.out.println("\t*          *");
        System.out.println("\t ***** ");
        System.out.println();
    }
}

```

- a)** Disegnare il CFG (Control Flow Graph) della classe ScegliCarta (partire dal main() e considerare le chiamate ai vari metodi. Fare attenzione alla rappresentazione dei parametri).
- b)** Scrivere dei casi di test che assicurino copertura 100% (statement coverage) del metodo stampaCarta (solo di questo metodo).

| |
|--|
| Domanda 5 – Parte generale (CONSEGNATA A PARTE – SENZA MATERIALE) |
|--|

Illustrare sinteticamente i punti salienti del modello RUP precisando in particolare

- a) che differenza c'è fra fasi e discipline
- b) quali aspetti sono messi maggiormente in risalto da RUP rispetto al modello a spirale