

Università degli Studi di Roma “La Sapienza”  
Facoltà di Ingegneria – Corso di Laurea in Ingegneria Gestionale  
**Corso di Progettazione del Software**  
Proff. Toni Mancini e Monica Scannapieco

Progetto **PC.20050329**

versione del 12 marzo 2007

Si vuole progettare un sistema per conto di *Fashion District*, un’agenzia a partecipazione pubblica che si occupa della promozione del settore della moda sul territorio.

Si richiede di effettuare le fasi di Analisi, Progetto, e Realizzazione del sistema in JAVA, utilizzando la metodologia illustrata nel corso.

## Requisiti

L’agenzia ha interesse a mantenere nel sistema informazioni circa le collezioni di capi d’abbigliamento prodotte, che sono caratterizzate da un identificativo (un intero), dagli stilisti che le realizzano, dall’anno e dal tema dominante (una stringa). Degli stilisti interessa il nome d’arte ed il numero di anni di esperienza (un attributo). Per semplicità, si può assumere che una collezione appartenga ad un solo stilista (cioè, se gli stilisti sono più di uno, sono comunque rappresentati da un unico nome d’arte, ad es. Dolce & Gabbana). Le collezioni possono essere presentate in sfilate, oppure essere pubblicizzate mediante altri canali (ad esempio riviste specializzate, video, portale Web dell’agenzia, etc.). Di una sfilata interessa il luogo in cui si tiene, la data di inizio e quella di fine.

Le collezioni possono essere di due tipi: primavera-estate e autunno-inverno. Delle collezioni primavera-estate si è interessati a conoscere se prevedono o meno costumi da bagno, mentre delle collezioni autunno-inverno se prevedono o meno pellicce.

L’applicazione deve consentire di effettuare indagini di mercato. Dato che il segmento di mercato più rappresentativo è costituito dalle collezioni pret-a-porter autunno-inverno, si vogliono rappresentare esplicitamente tali collezioni, caratterizzandole in base alla fascia di mercato (una stringa che rappresenta se la fascia di mercato è costituita da *teenager*, da *tempo libero*, etc.). Queste collezioni vengono acquistate da una o più aziende (almeno una), di cui interessa nome, partita IVA e capitale sociale. È inoltre di interesse conoscere il numero di capi di una collezione acquistati da ciascuna azienda.

L’applicazione deve fornire le seguenti funzionalità agli operatori dell’agenzia:

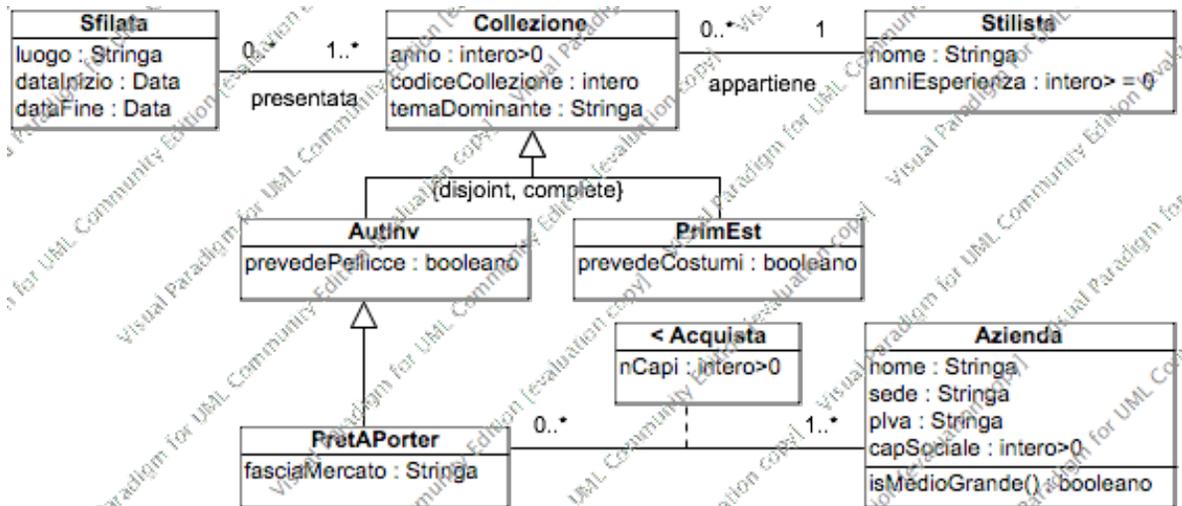
- Data una sfilata  $s$ , verificare se è una *sfilata di esordienti*, ovvero se la media degli anni di esperienza degli stilisti che hanno una collezione che sfila in  $s$ , è minore di 5. Si assuma per semplicità che ogni stilista ha al più una collezione che sfila in  $s$ . (Ulteriore esercizio: rimuovere tale assunzione).
- Data una collezione  $c$ , se  $c$  è pret-a-porter autunno-inverno si vuole restituire il numero totale di capi acquistati da aziende medio-grandi, cioè con capitale sociale maggiore di 10 milioni di euro, altrimenti (  $c$  è una collezione invernale non pret-a-porter oppure una collezione primavera-estate) il valore restituito deve essere 0.

## 1 Fase di Analisi

### 1.1 Diagramma degli Use Case



### 1.2 Diagramma delle classi UML



### 1.3 Specifica degli use case

SpecificaUseCase VerificheSfilate

verificaEsordienti(s : Sfilata) : booleano

pre: nessuna

post: Sia C l'insieme degli oggetti di classe Collezione coinvolti in link di tipo  
'presentata' con s, ovvero:

$$C = \{ c \text{ in Collezione} \mid \langle s, c \rangle \text{ in presentata} \}.$$

Sia inoltre

$$T = \{ t \text{ in Stilista} \mid \langle c, t \rangle \text{ in appartiene e } c \text{ in } C \}$$

l'insieme degli oggetti di classe Stilista coinvolti in link  
di tipo appartiene con oggetti in C.

result vale true se e solo se

$$\left( \sum_{(t \in T)} t.anniEsperienza \right) / |T| < 5.$$

calcolaVendite(c : Collezione): intero  $\geq 0$

pre: nessuna

post: Se c non e' di classe PretAPorter, allora result = 0.

Altrimenti (c e' di classe PretAPorter), detto

$$L = \{ \langle c, a \rangle \text{ in } c.acquista \mid a.isMedioGrande() = \text{true} \}$$

result e' pari a

$$\sum_{l \in L} (l.nCapi).$$

FineSpecifica

Si è quindi deciso di definire l'operazione *isMedioGrande()* : *booleano* nella classe *Azienda*.

### 1.4 Specifica delle classi

La classe Azienda

SpecificaClasse Azienda

isMedioGrande() : booleano

pre: nessuna

post: result = true se e solo se this.capSociale > 10 000 000.

FineSpecifica

## 2 Fase di Progetto

### 2.1 Corrispondenza tra tipi UML e tipi Java

Tipo UML	Tipo Java	Note
Stringa	String	-
Data	Data	usiamo vers. senza side-effect - disponibile
intero > 0	int	Verifica ammissibilità sul lato server
intero	int	-
booleano	boolean	-
Insieme(...)	HashSet<...>	Implementa l'interf. Set<...>

### 2.2 Specifica realizzativa delle classi

#### La classe Azienda

SpecificaClasse Azienda

+isMedioGrande() : boolean

pre: nessuna

algoritmo: se this.capSociale > 10 000 000 allora ritorna true;  
altrimenti ritorna false.

FineSpecifica

Nota: Data la similarità con la specifica concettuale, la specifica realizzativa della classe Azienda può essere omessa.

### 2.3 Specifica realizzativa degli use-case

SpecificaUseCase VerificheSfilate

+verificaEsordienti(s : Sfilata) : boolean

pre: nessuna

```
algoritmo:
    somma = 0;
    quanti = 0;
    per ogni 'l' in s.presentata {
        somma = somma + l.Collezione.appartiene.Stilista.anniEsperienza;
    }
    ritorna ( (somma/|s.presentata|) < 5 );

+calcolaVendite(c : Collezione): int
    pre: nessuna
    post: Se c non e' di classe PretAPorter, allora ritorna 0.
        Altrimenti (c e' di classe PretAPorter) {
            result = 0;
            per ogni 'l' in c.acquista {
                se l.Azienda.isMedioGrande() = true {
                    result += l.nCapi;
                }
            }
        }
    ritorna result;
FineSpecifica
```

## 2.4 Progetto dei diagrammi degli stati

Non sono stati definiti diagrammi degli stati in fase di Analisi.

## 2.5 Responsabilità sulle associazioni

Dai requisiti, dalla specifica delle operazioni di classi e di use case, e delle molteplicità nel diagramma delle classi emerge che:

Associazione	Classe	Ha resp?	Motivo
presentata	Sfilata	SI	vincolo 1..*, u.c. op. 1
	Collezione	NO	-
appartiene	Collezione	SI	vincolo 1..1 e u.c., op. 1
	Stilista	NO	-
acquista	PretAPorter	SI	vincolo 1..* e u.c., op. 2
	Azienda	NO	-

## 2.6 Vincoli sull'evoluzione delle proprietà mutabili

Tutte le proprietà mutabili possono variare arbitrariamente, tranne i link di associazione "presentata" e "acquista" che non possono essere cancellati.

## 2.7 Diagramma delle classi realizzativo

