LabProg 2011-2012 - TESTO ESAME
Andrea Trentini - DICo - UniMi
01 febbraio 2012 (a)

Avvertenza. Non è ammesso l'uso delle classi del package *prog.io* allegato al libro di testo del corso.

1 Testo esame

Lo scopo è realizzare un modello di "social network" minimale.
Le classi da realizzare sono le seguenti (dettagli nelle sezioni successive):

- *Persona*: un utente del network, avrà una lista di amici.
- *Gruppo*: raggruppa un insieme di Persona in un gruppo tematico
- *Main*: contiene il solo metodo *main()* con alcune invocazioni di test (importante è riuscire a testare TUTTI i metodi almeno una volta)

Tali classi dovranno esporre almeno i metodi specificati nelle sezioni seguenti. Eventuali metodi di servizio possono essere aggiunti a piacimento. Ogni classe deve avere il *toString()* che rappresenti lo stato delle istanze e i **costruttori** adeguati per gli attributi che vengono dichiarati. Creare opportunamente i metodi di accesso agli attributi (set&get). Si suggerisce, anche dove non segnalato, di utilizzare, se esistenti e se applicabili, le classi parametriche (es. *ArrayList<E>* invece di *ArrayList*). Alcuni controlli di coerenza vengono suggeriti nel testo, potrebbero essercene altri a discrezione. I tipi di ritorno possono essere variati (ad es. boolean invece di void se si vuole ottenere un feedback sul successo dell'operazione) previa autorizzazione del docente. Consiglio di posporre l'implementazione dei controlli di coerenza, fatelo come ultima operazione, prima realizzate un sistema funzionante.

1.1 Persona

Rappresenta un utente, ha (attributi) nome, cognome, nick, email ed è collegato ad altre Persona (lista amici).
Deve inoltre esporre i seguenti metodi **pubblici**:

- *Persona[] listaAmici()*: fornisce la lista degli amici
- *boolean richiestaAmicizia(Persona p)*: permette la richiesta di amicizia da parte di un altro utente, restituisce 'true' se viene accettata

Esempio (con lacune parametriche):

```
Persona p1=new Persona(.......);
Persona p2=new Persona(.......);
Persona p3=new Persona(.......);
p1.richiestaAmicizia(p2); // p2 chiede amicizia a p1
```

1.2 Gruppo

Rappresenta un gruppo di utenti, ha (attributi) un nome breve e una descrizione.
Deve inoltre esporre i seguenti metodi **pubblici**:

- *Persona[] listaMembri()*: fornisce la lista dei membri del gruppo
- *boolean richiestaJoin(Persona p)*: permette la richiesta di entrare nel gruppo da parte di un utente, restituisce 'true' se viene accettata

1.3 Suggerimento

Se notate somiglianze fra le classi comportatevi di conseguenza...

1.4 Main

Deve contenere il metodo *main()* in cui vanno istanziate le classi realizzate e opportunamente testate.

2 Consegna

Ricordo che le classi devono essere tutte *public* e che vanno consegnati tutti i file *.java* prodotti. NON vanno consegnati i .class Per la consegna, eseguite l'upload dei singoli file sorgente dalla pagina web: http://upload.dico.unimi.it.