Facebook ha deciso di passare a Java come linguaggio per la sua piattaforma. Nel

compito di oggi vi viene richiesto di realizzare una parte del codice. In

Facebook, ogni utente è collegato ad un insieme di amici. L'utente può inoltre

caricare delle foto ed indicare quali amici sono presenti in ogni foto. In

questo esercizio svilupperete una versione semplificata che collega (“tagga") un

solo amico ad ogni foto. Si consideri la classe Amici, avente come variabili

d'istanza nome (stringa), cognome (stringa) e età (intero). Si consideri inoltre

la classe Foto, avente come variabili d'istanza titolo (string) e amico, che

rappresenta un riferimento a un oggetto di classe Amico. Si assuma siano già

disponibili gli usuali metodi costruttori, accessori e modificatori per ciascuna

delle due classi menzionate. Sviluppare tutti i metodi della seguente classe.

public class Tag {

private ArrayList<Foto> fotografie;

private Amico[] amici;

private int numAmici;

private final int MAX\_NUM\_AMICI = 200;

public Tag(){...}

public void aggFoto(String unTitolo){...}

public void aggAmico(String unNome, String unCognome, unaEta){...}

public void aggTag(String unTitolo, String unNomeAmico, String

unCognomeAmico){...}

public void rimuoviAmico(String unNomeAmico, String unCognomeAmico){..}

public double calcolaEtaMedia(){...}

public int contaFotoAmico(String unNome, String unCognome){...}

public Amico calcolaMaxApparizioni() {...}

}

Le variabili d'istanza fotografie e amici contengono rispettivamente tutti gli

amici e le foto presenti.

La variabile amici è un array di dimensione MAX NUM AMICI, e deve essere gestito

come un array parzialmente riempito. Il metodo aggAmico inserisce un nuovo amico

(se non esiste già e se vi e disponibilità di spazio). Il metodo aggFoto

inserisce una nuova foto (se non esiste già) indicando il titolo che la

identifica univocamente e senza alcun amico taggato. Il metodo aggTag inserisce

un collegamento tra la foto indicata e l'amico indicato. Se la foto non esiste,

il metodo esce senza fare nulla. Se l'amico non esiste deve crearlo, ponendo

l'età al valore -1, che serve ad indicare una età sconosciuta. Se non vi è

disponibilità di spazio per un nuovo amico, il metodo esce senza fare nulla. Il

metodo rimuoviAmico rimuove l'amico indicato assieme a tutte le foto in cui egli

è presente. Il metodo calcolaEtaMedia calcola l'età media degli amici, non

considerando gli amici di cui non si conosce l'età. Il metodo contaFotoAmico

calcola il numero di foto in cui appare l'amico indicato. Il metodo

calcolaMaxApparizioni restituisce un riferimento ad un oggetto di classe Amico

presente nel maggior numero di foto (risolvere arbitrariamente i casi di

parità).

Consigli:

Si invita ad implementare due metodi ausiliari che verifichino la presenza di un

amico e di una foto rispettivamente nell’Array amici e nell’ArrayList

fotografie. Il metodo rimuoviAmico può dare difficoltà in quanto non solo

rimuove l’amico dall’Array amici, ma rimuove anche tutte le foto in cui l’amico

da eliminare è taggato(si consiglia di rivedere l’esercizio 7.1 poiché le foto

da eliminare possono essere contigue nell’ArrayList fotografie).

Nella classe Foto si consiglia di implementare un metodo setAmico che aiuta la

stesura della classe Tag.