

# Programació I - Examen 2<sup>on</sup> parcial - 20 de gener de 2014

**ENUNCIAT:** Es desitja dissenyar un programa que identifiqui i guardi en un àlbum les fotos publicades a Instagram del conjunt de contactes que es tenen en el mòbil. Per això, s'ha decidit implementar un conjunt d'opcions en un programa de prova reflectides en el següent menú:

1. Actualitzar contacte: a partir d'un contacte, es guarden les fotos d'Instagram que contenen la seva cara en l'àlbum del mòbil.
2. Consultar fotos de grup en l'àlbum: conjunt de fotos on surten simultàniament més d'un contacte del mòbil.
3. Contactes sense foto en l'àlbum: identifica els contactes que no surten a cap foto a Instagram.

La informació associada a una **foto** és la seva mida en amplada i alçada (mesurades en píxels) i la matriu de píxels corresponent. Cada píxel està representat per un color de tres caràcters que codifiquen els tres canals de color R (vermell), G (verd), B (blau).

Cada **contacte** està identificat amb el nom, una foto, un telèfon i un mail. La mida de la foto del contacte serà fixa de 200x200 píxels. Es suposa que la foto del contacte conté només la cara del contacte. Com a màxim, en el mòbil es poden tenir mil contactes i com a molt 20 fotos per contacte.

A l'etapa de disseny s'ha decidit crear les següents classes: **Contacte** (per a mantenir totes les dades d'un contacte), **Foto** (que manté les dades d'una foto), **Agenda** (per a guardar la col·lecció de contactes que es tenen en el mòbil), **Album** (per a guardar la col·lecció de fotos del mòbil) i la classe **Gestio** (per a implementar el control principal de l'aplicació). Al final d'aquest enunciat, s'adjunta un llistat per a veure part de la implementació de la classe **Gestio**. Teniu-lo en compte per resoldre l'examen.

Per facilitar la resolució dels problemes següents, considera que:

- Es disposen dels següents mètodes, ja implementats a la classe **Instagram** :

```
public class Instagram {
    // Constructor que crea i conecta amb Instagram donats el nom d'usuari
    // i la clau secreta
    public Instagram (String usuari, String clau);
    // Dona el nombre de Fotos que l'usuari pot veure a Instagram
    public int getNumFotosVisibles();
    // Dona la següent Foto visible per l'usuari
    public Foto nextFoto();
}
```

- Es disposa d'una classe **Utilitats** que té implementat el mètode `identificarCara` que permet identificar una cara dins d'una foto, donada una foto on només surt la cara a identificar:

```
public class Utilitats {
    // Retorna true si la cara de la foto1 esta dins de la foto 2 i false en
    // cas contrari
    public static boolean identificarCara(Foto foto1, Foto foto2);
    // Retorna true si la foto1 es igual a la foto2, i false en cas contrari
    public static boolean iguals (Foto foto1, Foto foto2);
}
```

- La classe **Agenda** està implementada. Suposa que no hi ha dos contactes amb el mateix nom. Es disposen dels següents mètodes públics:

```
public class Agenda {
    // Constructor de la classe Agenda
    public Agenda (int maxContactes);
    // Afegeix el nouContacte a l'agenda de contactes
    public boolean addContacte(Contacte nouContacte);
    // Donat el nom d'un contacte retorna tota la seva informacio en el parametre
    // cont.
    // Si existeix retorna 0, sino retorna -1.
}
```

```

public int getContacte(String nom, Contacte cont);
// Donat el index d'un contacte retorna tota la seva informacio.
// Si no existeix retorna null.
public Contacte getContacte(int idx);
// Retorna el numero de contactes de l'agenda
public int getNumContactes();
}

```

**PROBLEMES A RESOLDRE I ENTREGAR:** Cal entregar CADA PROBLEMA EN UN FULL PER SEPARAT i posar el Nom, Cognoms i DNI en tots els fulls. Utilitzeu els tipus bàsics de Java per a definir les classes. No està permès utilitzar tipus de dades predefinitos com ArrayLists.

1. (1,5 punts) Defineix els atributs de les classes **Foto** i **Contacte**, contestant els següents apartats:
  - (a) Justifica els atributs que caracteritzen aquestes classes i els seus tipus. Segons la teva definició dels atributs, caldria implementar els mètodes get i set de cadascun dels atributs? Per què?
  - (b) Implementa només el constructor de la classe **Contacte** que permeti passar les dades necessàries per a inicialitzar els atributs. És necessari utilitzar la constructora de la classe **Foto**? I de la classe String? Justifica la teva resposta argumentant els avantatges i els inconvenients de la teva solució.
2. (1,5 punts) Defineix la classe **Album** amb els seus atributs.
  - (a) Justifica els tipus dels atributs que defineixen aquesta classe.
  - (b) Implementa el seu constructor que permeti realitzar les seves inicialitzacions. Podries sobrecarregar el constructor? Com?
3. (2 punts) Es vol implementar el mètode `actualitzarFotosContacte` que donat un contacte de l'agenda, es baixin d'Instagram totes les fotos que contenen la seva cara a l'àlbum del mòbil. Suposa que a Instagram hi ha com a màxim 20 fotos amb la cara d'una persona.
  - (a) Completa el mètode de la classe `Gestio` anomenat `opcActualitzarFotosContacte`, segons el llistat de la classe `Gestio`.
  - (b) Defineix la capçalera del mètode `actualitzarFotosContacte`. Justifica a quina classe pertany. És un mètode de classe o és un mètode d'objecte?
  - (c) Identifica la seqüència i l'esquema a aplicar per a implementar el mètode `actualitzarFotosContacte` i implementa'l.
4. (2,5 punts) Es vol implementar el mètode `contactesSenseFotoAlbum` que retorna una còpia dels contactes d'una certa agenda que no tenen cap foto en l'àlbum. Suposa que en les fotos del mòbil, s'han actualitzat prèviament les fotos de tots els contactes.
  - (a) Defineix la capçalera del mètode `contactesSenseFotoAlbum`. Justifica a quina classe pertany i quin tipus de retorn utilitzes.
  - (b) Identifica la seqüència i l'esquema a aplicar per a implementar el mètode `contactesSenseFotoAlbum` i implementa'l.
5. (2,5 punts) Es vol implementar el mètode `consultarFotoGrup` que retorna les còpies de totes les fotos del mòbil on surten més d'un contacte d'una agenda. En les fotos retornades no en sortiran repetides. Suposa que en les fotos del mòbil, només es tenen les fotos baixades d'Instagram.
  - (a) Defineix la capçalera del mètode `consultarFotoGrup`. Justifica a quina classe pertany i raona la teva definició dels paràmetres formals d'aquest mètode.
  - (b) Identifica la seqüència i l'esquema a aplicar per a implementar el mètode `consultarFotoGrup` i implementa'l. Comenta la teva estratègia de resolució.

Llistat de la classe principal de l'aplicació, amb els mètodes que ja teniu implementats:

# Listing 1: Gestio.java

```
import java.util.Scanner;

public class Gestio {
    public static final int ACTUALITZAR    = 1;
    public static final int FOTOS_GRUP    = 2;
    public static final int NO_FOTO        = 3;
    public static final int SORTIR         = 4;
    public static final int MAX_CONTACTES = 1000;
    public static final int MAX_FOTOS      = 20*MAX_CONTACTES;

    public static void main (String [] args) {
        Scanner sc;
        Instagram ins;
        Agenda agenda;
        Album album;

        int op;
        sc = new Scanner(System.in);
        agenda = new Agenda(MAX_CONTACTES);
        album = new Album(MAX_FOTOS);
        ins = new Instagram("Pepito", "clavePepito");

        op = pintaMenuPreguntaOpcio(sc);
        while (op!=SORTIR) {
            switch(op) {
                case ACTUALITZAR: opcActualitzarFotosContacte(ins, sc, agenda, album); break;
                case NO_FOTO:     opcioNoFotoAlbum(sc, agenda, album); break;
                case FOTOS_GRUP:  opcioFotosGrup(sc, agenda, album); break;
                default: break;
            }
            op = pintaMenuPreguntaOpcio(sc);
        }
    }

    public static opcActualitzarFotosContacte(Instagram ins, Scanner sc, Agenda ag,
        Album alb) {
        ...

        System.out.println("Quin_contacte_vols_actualitzar?")
        nomContacte = sc.next();
        ....

        if (..... ) {
            // Cas que el contacte estigui a l'agenda
            .....
        } else {
            // Cas que el contacte no existeixi
            System.out.println("contacte_inexistent_a_l'agenda_del_mobil");
        }
    }
}
```