

Examen Parcial: Solució

6 d'abril de 2011

Nom i Cognoms:

NIUB:

Contesteu els següents exercicis en els espais en blanc.

Exercici 1 (1 punts)

Escull i explica quatre dels següents conceptes:

1. Abstracció
2. Signatura d'un mètode
3. Herència
4. Mètode d'objecte
5. Polimorfisme
6. Tipus dinàmic

Definicions del glossari de termes:

- **Abstracció** *f* Capacitat de concebre un concepte sense pensar en cap exemple específic.
- **Signatura d'un mètode** *f* Allò que identifica un mètode de forma única: la classe en la que es defineix, el nom, el número, ordre i tipus dels arguments.
- **Herència** *f* Mecanisme amb què es defineix una classe a partir d'unes altres, descrivint només les característiques que la diferencien de les altres.
- **Mètode d'objecte** *m* Mètode que pot ser el mateix per a tota una classe d'objectes i depèn de les dades concretes d'una instància. Cal que existeixi un objecte al qual poder-lo aplicar.
- **Polimorfisme** *m* Capacitat que permet substituir el comportament d'un mètode dins una jerarquia de classes.
- **Tipus dinàmic** *m* Tipus d'una variable que només es pot conèixer en temps d'execució i pot canviar.

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

Exercici 2 (2 punts)

Donat el següent codi de les classes Animal, Gos i Gat i el diagrama il·lustrant la relació entre les classes

<pre>public abstract class Animal{ public void ferSoroll(){ System.out.println("L'animal fa soroll"); } } public class Gat extends Animal{ public void esgarrapar() { System.out.println("esgarrapa"); } } public class Gos extends Animal{ public void persegueix() { System.out.println("persegueix"); } }</pre>	<pre>classDiagram class Animal { +void ferSoroll() } class Gat { +void esgarrapar() } class Gos { +void persegueix() } Animal < -- Gat Animal < -- Gos</pre>
--	--

Respon a les següents preguntes:

- 1) Indica al costat de cadascuna de les línies del següent codi si hi haurà errors de compilació o no i explica breument perquè:

0	public static void main(String[] args) {	//OK
1	Gos unGos = new Gos();	//OK
2	Gat unGat = new Gat();	//OK
3	Animal[] llistaAnimals = new Animal[2];	//OK
4	llistaAnimals[0]=unGos;	//OK
5	llistaAnimals[1]=unGat;	//OK
6	Animal primerAnimal;	//OK
7	primerAnimal = llistaAnimals[0];	//OK
8	primerAnimal.ferSoroll();	//OK
9	primerAnimal.persegueix();	//Error de compilació
10	primerAnimal.esgarrapar();	//Error de compilació
11		
12	Gat segonAnimal;	//OK
13	segonAnimal = llistaAnimals[1];	//Error de compilació
14		
15	}	

- 2) Especifica si hi ha alguna assignació polimorfa en el codi anterior i en cas afirmatiu en quines línies.

Solució: Sí, a la línia 4 i 5

- 3) Indica que passa quan fem una crida d'aquesta forma:
unGos.ferSoroll();

Solució:
Apareixerà el missatge: "L'animal fa soroll"

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

- 4) Indica que hem de fer per que quan fas la crida
 unGos.ferSoroll();
La sortida sigui: "El gos fa soroll"

Solució:

Sobrescriure el mètode ferSoroll a la classe Gos

```
public void ferSoroll(){  
    System.out.println("El gos fa soroll");  
}
```

- 5) Indica que hem de fer per que quan fas la crida
 unGos.ferSoroll();
La sortida sigui: "L'animal fa soroll"
 "I el gos fa soroll"

Solució:

Sobrescriure el mètode ferSoroll a la classe Gos

```
public void ferSoroll(){  
    super. ferSoroll ();  
    System.out.println("I el gos fa soroll");  
}
```

- 6) Donades les classes anteriors i donat el següent mètode main:

```
public static void main(String[] args){  
  
    Animal[] llistaAnimals = new Animal[2];  
    llistaAnimals[0]= new Gos();  
    llistaAnimals[1]= new Gat();  
  
}
```

Indica quin codi has d'afegir en aquest mètode per tal de que si cridem
altreGos.persegueix();
aparegui la següent sortida per pantalla:

"persegueix"

Solució:

```
Gos unAltreGos;  
if (llistaAnimals[0] instanceof Gos){  
    unAltreGos = (Gos) llistaAnimals[0];  
    unAltreGos.persegueix();  
end
```

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

Exercici 3 (1 punts)

Donat el següent programa

```
public class Aliases {  
    public static void main(String[] args) {  
        Gos gos1 = new Gos();  
        Gos gos2= new Gos();  
        Gos gos3= new Gos();  
        gos1.nom = "coco";  
        gos2.nom = "ella";  
        gos3.nom = "roc";  
        gos3 = gos1;  
        gos1 = gos2;  
        gos1.nom = "lassie";  
        gos2.nom = "rex";  
        System.out.println("El nom de l'objecte referenciat per gos1 és:" + gos1.nom);  
        System.out.println("El nom de l'objecte referenciat per gos2 és:" + gos2.nom);  
        System.out.println("El nom de l'objecte referenciat per gos3 és:" + gos3.nom);  
    }  
}
```

```
public class Gos {  
    public String nom;  
}
```

Indica quina és la sortida d'aquest codi.

Solució:

El nom de l'objecte referenciat per gos1 és:rex

El nom de l'objecte referenciat per gos2 és:rex

El nom de l'objecte referenciat per gos3 és:coco

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

Exercici 4 (4 punts)

A l'Hotel Plaza es desitja desenvolupar un programa que permeti emmagatzemar una llista de clients que s'inscriuen com clients habituals i que com a tals paguen una quota anual. La informació d'interès dels clients és el nom, DNI, telèfon i quota anual. En aquesta llista hi poden haver dos tipus de clients, els clients estàndard i els clients vip. Els primers paguen la totalitat de la quota anual i els segons tenen un descompte del 50% sobre aquest preu.

a) Seguint aquest enunciat, implementa en llenguatge Java les tres classes següents que es necessitaran per al programa:

- **Client**,
- **ClientEstandard** (classe que hereta de la classe **Client**)
- **ClientVip** (classe que hereta de la classe **Client**)

Inclou a cada classe els atributs (que s'han de deduir de l'enunciat) i els següents mètodes:

- Un sol constructor on s'inicialitzen tots els atributs de la classe
- Tots els mètodes consultors i modificadors per als atributs de la classe.
- Un mètode per imprimir la quota anual, anomenat `imprimirQuotaAnual`, que serà diferent per les tres classes:
 - i. Serà un mètode abstracte de la classe **Client**.
 - ii. A la classe **ClientEstandard**, aquest mètode mostrarà per pantalla el nom del client i la seva quota anual.
 - iii. A la classe **ClientVip**, aquest mètode imprimirà per pantalla el nom del client i la quota anual aplicant-li el descompte corresponent a aquest tipus de clients.

Organitza el codi en fitxers amb noms i indica els paquets i els imports necessaris.

b) Implementa una altra classe **GestorClients** amb un mètode `main` on es creï una llista de clients i s'ompli amb les següents instàncies:

- John Smith, amb el DNI 10922456 i telèfon 93 123 45 67 és un client vip.
- Peter Adams, amb el DNI 21987410 i telèfon 93 321 78 96 és un client estàndard.

Considera que la quota anual és de 110.50€.

Recorre aquesta llista fent servir un iterador i invoca el mètode `imprimirQuotaAnual` per a cada element.

c) Indica quina és la sortida del mètode `main`.

Solució:

```
//*****
// Client.java
//*****
package examen;

public abstract class Client {
    private String nom;
    private int dni;
    private String telefon;
    private double quotaAnual;

    // Constructor
    public Client(String nom, int dni, String telefon, double quotaAnual){
        this.nom = nom;
        this.dni = dni;
        this.telefon = telefon;
        this.quotaAnual = quotaAnual;
    }
}
```

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

```
//Mètode abstracte, es declara sense implementació
public abstract void imprimirQuotaAnual();

// Mètodes modificadors:
public void setNom(String nom){
    this.nom=nom;
}
public void setDNI(int dni){
    this.dni=dni;
}
public void setTelefon(String telefon){
    this.telefon=telefon;
}
public double setQuotaAnual(){
    return quotaAnual;
}

// Mètodes consultors:
public String getNom(){
    return this.nom;
}
public String getTelefon(){
    return this.telefon;
}
public String getDNI(){
    return this.telefon;
}
public double getQuotaAnual(){
    return quotaAnual;
}
}

//*****
// ClientVip.java
//*****
package examen;

public class ClientVip extends Client{
    private double descompte;

    // Constructor
    public ClientVip(String nom, int dni, String telefon, double quotaAnual){
        super(nom, dni, telefon, quotaAnual);
        this.descompte = 0.5;
    }
    // Mètode propi d'aquesta classe
    public double calcularQuota() {
        double quota;
        quota = super.getQuotaAnual();
        return quota - (quota*descompte);
    }

    //Implementació del mètode abstracte de la superclasse Client
    public void imprimirQuotaAnual() {
        System.out.println("La quota anual a pagar pel client " + super.getNom() + " és: " +
this.calcularQuota());
    }
}
```

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

```
//*****
// ClientEstandard.java
//*****
package examen;

public class ClientEstandard extends Client{

    // Constructor
    public ClientEstandard(String nom, int dni, String telefon, double quotaAnual){
        super(nom, dni, telefon, quotaAnual);
    }
    //Implementació del mètode abstracte de la superclasse Client
    public void imprimirQuotaAnual() {
        System.out.println("La quota anual a pagar pel client " + super.getNom() + " és: " +
        super.getQuotaAnual());
    }
}

//*****
// GestorClients.java
//*****
package examen;

import java.util.ArrayList;
import java.util.Iterator;

public class GestorClients {
    public static void main(String[] args){
        String nom;
        int dni;
        String telefon;
        double quotaAnual;
        quotaAnual = 110.50;

        // Creem les dues instàncies que necessitem:
        nom = "John Smith";
        dni = 10922456;
        telefon = "93 123 45 67";
        ClientVip client1 = new ClientVip(nom, dni, telefon, quotaAnual);

        nom = "Peter Adams";
        dni = 21987410;
        telefon = "93 321 78 96";
        ClientEstandard client2 = new ClientEstandard(nom, dni, telefon, quotaAnual);

        // Creem una llista de Clients
        ArrayList<Client> llistaClients = new ArrayList<Client>();

        // Omplim la llista amb les tres instàncies:
        llistaClients.add(client1);
        llistaClients.add(client2);

        //Recorrem aquesta llista invocant al mètode imprimirQuotaAnual() per a cada element:
        for(Iterator<Client> i = llistaClients.iterator();i.hasNext();){
            i.next().imprimirQuotaAnual();
        }
    }
}
```

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

Sortida:

La quota anual a pagar pel client John Smith és: 55.25

La quota anual a pagar pel client Peter Adams és: 110.5

Exercici 5 (2 punts)

L'aplicació presentada a l'exercici anterior es vol ampliar afegint el control de reserves de l'hotel que fa cada client, així com el control del número de reserves que s'han pagat en total a l'hotel.

- a) Per fer això, implementa en llenguatge Java una classe **ReservaHotel** que implementa **una** reserva d'un client. Aquesta classe ha de tenir com a mínim un mètode constructor al que li passem un objecte de tipus **Client** i un mètode anomenat *pagametRealitzat* que incrementa en un la quantitat de reserves pagades.
- b) Indica a més el codi que s'hauria d'afegir al mètode main de la classe **GestorClients**, implementat a l'exercici anterior, per tal de que el client Jonh Smith faci dues reserves d'hotel, les pagui i es mostri per pantalla el número de reserves pagades.

Per fer els dos apartats (a) i (b) d'aquest exercici no feu servir ni taules ni arrays.

Solució:

```
//*****  
// ReservaHotel.java  
//*****  
package examen;  
  
public class ReservaHotel {  
    private Client client;  
    private static int numReservesPagades;  
  
    public ReservaHotel(Client client){  
        this.client = client;  
    }  
  
    public void pagametRealitzat (){  
        numReservesPagades++;  
    }  
  
    // Mètodes consultors:  
    public static int getNumReserves(){  
        return numReservesPagades;  
    }  
  
    public Client getClient(){  
        return client;  
    }  
  
    // Mètode modificador:  
    public void setClient(Client client){  
        this.client = client;  
    }  
}
```


Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2010-2011.

```
//*****
// GestorClients.java
//*****

public class GestorClients{

    public static void main(String [] args){

        // Afegir al codi de l'exercici anterior
        ReservaHotel reserva1 = new ReservaHotel(client1);
        reserva1.pagamentRealitzat();
        ReservaHotel reserva2 = new ReservaHotel(client1);
        reserva2.pagamentRealitzat();

        System.out.println("Número de Reserves pagades = " + ReservaHotel.getNumReserves());
    }
}

//*****
Sortida:
//*****

Número de Reserves pagades = 2
```