Examen Parcial: Solució

6 d'abril de 2010

Nom i Cognoms: ..... NIUB: .....

Contesteu els següents exercicis en els espais en blanc.

<u>Exercici 1</u> (1.5 punts)

Escollís i explica tres dels següents conceptes:

1. Assercions
2. Encapsulament
3. Herència
4. Mètode de classe
5. Polimorfisme
6. Sobrecàrrega

Definicions del glossari de termes:

- **Assercions** *f* Sentencies de codi verdader-fals per comprovar suposicions en temps d'execució.

- **Encapsulament** *m* Ocultació de la implementació concreta d'una classe en què s'ofereixen només aquelles dades que es vol que les altres classes puguin utilitzar.

- **Herència** *f* Mecanisme amb què es defineix una classe a partir d'unes altres, descrivint només les característiques que la diferencien de les altres.

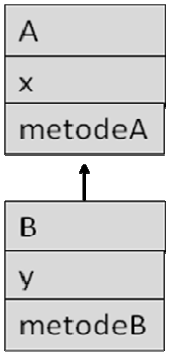
- **Mètode de classe** *m* Mètode que és el mateix per a tota una classe d'objectes i no depèn de les dades concretes de cap instància. No cal que existeixi cap objecte per a poder-lo aplicar.

- **Polimorfisme** *m* Capacitat que permet substituir el comportament d'un mètode dins una jerarquia de classes.

- **Sobrecàrrega** *f* Capacitat perquè el nom d'una característica, mètode o atribut faci referència a dos elements.

Exercici 2 (2 punts)
Donat el següent codi de la classe A i la classe B (que hereta de la classe A) i el diagrama il·lustrant la relació entre les classes

```
public class A{
        protected int x;
        public void metodeA(){
                ....
        }
}
public class B extends A{
        private int y;
        public void metodeB(){
                ....
        }
}
```



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) {
1               A a = new B();   OK
2               B b = new B();   OK
3               A o;             OK
4               o = b;           OK
5               int j = b.x;     OK
6               int i = a.x;      OK
7               int k = a.y;     Error de compilació, l'atribut y no està defint per a A.
8               a.metodeA();     OK
9               a.metodeB();     Error de compilació, metodeB no està definit per a A
10              b.metodeA();     OK
11              b.metodeB();     OK
12      }
```

2) Especifica si hi ha alguna **conversió** de tipus **implícita** en el codi anterior i en cas afirmatiu en quines línies.
Sí, a la línia 1 i 4

3) Si afegim un nou mètode a la classe A anomenat imprimir que imprimeix el missatge "Missatge d'A", però no el sobreescriu a la classe B, que passa quan fem una crida d'aquesta forma:
b.imprimir();
Apareixerà el missatge: "Missatge d'A"

4) Indica com has de sobreescriure el mètode imprimir a la classe B de manera que quan fas la crida
b.imprimir();
La sortida sigui: "Missatge de B"

```
public void imprimir(){
        System.out.println("Missatge de B");
}
```

5) Ara, indica com has de sobreescriure el mètode imprimir a la classe B de manera que quan fas la crida
   b.imprimir();
   La sortida sigui: "Missatge d'A"
   "Missatge de B"

```
public void  imprimir(){
        super.imprimir();
        System.out.println("Missatge de B");
}
```

6) Donades les classes A i B definides a dalt, indica quins seran els atributs visibles en una instància de la classe B.

   Al declarar la variable **x** com a ***protected*** indica que només aquesta classe, les clases que deriven d'ella i les classes del propio ***package*** tenen permis per utilizar-la.
   La variable **y** és *privada* (***private***), la qual cosa vol dir que només les funcions membre d'aquesta classe tenen permis per utilizar-la.

   - Atributs visibles internament: **x i y**
   - Atributs visibles des de classes del mateix paquet: **x**
   - Atributs visibles des de classes d'un altre paquet: **cap**

   Exemple:

   1. Nova classe del mateix paquet

```
package paquetB;

public class NovaB {
   public static void main(String[] args) {
      B b = new B();
       System.out.println("access a x" + b.x);
         // System.out.println("access a x" + b.y);
   }
}
```

   2. Nova classe d'un altre paquet

```
package nouPaquet;
import paquetB.B;

public class NovaB {
   public static void main(String[] args) {
      B b = new B();
       // System.out.println("access a x" + b.x);
       // System.out.println("access a x" + b.y);
   }
}
```

Exercici 3 (1 punts)

Donat el següent programa

```
public class Aliases {
    public static void main(String[]args) {
            Persona x = new Persona();
            Persona y = new Persona();
            Persona z;
            x.edat = 20;
            y.edat = 30;
            z = y;
            y = x;
            x = z;
            x.edat = 26;
            System.out.println("L'edat en l'objecte referenciat per x és:" + x.edar);
            System.out.println("L'edat en l'objecte referenciat per y és:" + y.edat);
            System.out.println("L'edat en l'objecte referenciat per z és:" + z.edat);
    }
}
class Persona {
   public String nom;
   public int edat;
}
```

Indica quina és la sortida:

```
Persona x = new Persona();
Persona y = new Persona();
Persona z;
x.edat = 20;
y.edat = 30;
z = y;
```

x →Objecte1 (edat=20)
y →Objecte2 (edat=30)
z →Objecte2 (edat=30)

y=x
x →Objecte1 (edat=20)
y →Objecte1 (edat=20)
z →Objecte2 (edat=30)

x=z
x → Objecte2 (edat=30)
y →Objecte1 (edat=20)
z →Objecte2 (edat=30)

x.edat = 26;
x → Objecte2 (edat=26)
y →Objecte1 (edat=20)
z →Objecte2 (edat=26)

Sortida:

L'edat en lobjecte referenciat per x es:26
L'edat en lobjecte referenciat per y es:20
L'edat en lobjecte referenciat per z es:26

Exercici 4 (1.5 punts)

Donada la interfície **InstrumentMusical** amb els mètodes tocar i afinar definida a continuació:

```
interface InstrumentMusical {
        public void tocar();
        public void afinar();
}
```

Defineix amb llenguatge Java la classe **Guitarra** que implementa aquesta interfície i imprimeix els següents missatges dins dels mètodes tocar: "La guitarra sona" i afinar: "La guitarra s'afina". Implementa també en la classe Guitarra un mètode sumaGuitarres que actualitza el compte de la quantitat d'objectes creats dins d'aquesta classe.

```
//*********************************************************************
// Guitarra.java
//*********************************************************************
public class Guitarra implements InstrumentMusical{
   private static int numGuitarres=0;
   // constructor
   public Guitarra(){
        sumarGuitarres();
   }
   // Estem obligats a implementar els mètodes de la interfície InstrumentMusical
   public void tocar(){
      System.out.println("La guitarra sona");
   }
   public void afinar(){
      System.out.println("La guitarra s'afina");
   }

   private static void sumarGuitarres(){
      numGuitarres++;
   }
// definim el mètode sumarGuitarres public o private depenent de la utilització que anem a fer
}


//*********************************************************************
// ProvaGuitarres.java
//*********************************************************************

public class ProvaGuitarres {

  public static void main(String [] args){

      for (int i=0;i<10;i++){
         Guitarra g = new Guitarra();
      }
      System.out.println("Número de Guitarres = " + Guitarra.numGuitarres);
   }
}

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

Número de Guitarres = 10
```

Exercici 5 (4 punts)

a) "A la Facultat de Matemàtiques de la UB es desitja desenvolupar un programa que permeti emmagatzemar en una llista els estudiants que s'inscriuen per participar en una activitat de la Facultat. La informació d'interès d'un estudiant és el nom, edat i número de carnet d'estudiant. En aquesta activitat hi poden participar estudiants de grau i estudiants de màster. Els primers tindran associada la següent informació: el nom del grau que estan cursant, el número d'assignatures que han matriculat aquest any i el preu de cada assignatura (preu únic per a totes les assignatures segons el grau). Els estudiants de màster tindran associada la següent informació: el nom del màster que estan cursant, el preu del màster i si disposen d'alguna beca o no."
Seguint aquest enunciat, implementa en Java les tres classes següents que es necessitaran per al programa:
- **Estudiant** (classe abstracta),
- **EstudiantGrau** (classe que hereta de la classe **Estudiant**)
- **EstudiantMaster** (classe que hereta de la classe **Estudiant**)

incloent els seus 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 calcular el preu de la matrícula d'aquest any anomenat imprimirPreuMatricula.
  El mètode imprimirPreuMatricula serà un mètode abstracte de la classe **Estudiant**, ja que s'han de conèixer els estudis que està cursant l'estudiant per tal de poder fer el càlcul del preu de la matrícula. A la classe **EstudiantGrau**, aquest mètode calcularà el preu de la matrícula multiplicant el número d'assignatures matriculades pel seu preu únic i imprimirà per pantalla el nom de l'estudiant i el preu resultant de la seva matrícula. A la classe **EstudiantMaster**, aquest mètode imprimirà per pantalla el nom de l'estudiant i el preu de la matrícula del màster que serà igual a la quantitat del preu del màster si l'estudiant no té una beca i serà 0 euros si l'estudiant té una beca.

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

b) Implementa una altra classe **Activitat** amb un mètode main on es crea una llista genèrica d'estudiants i s'ompli amb les següents instàncies:
- Maria de 18 anys, estudiant del grau de Matemàtiques, que té un carnet amb el número 0001 i que cursa 4 assignatures amb un preu de 100,00 euros cadascuna.
- Joan de 20 anys, estudiant del grau d'Enginyeria Informàtica, que té un carnet amb el número 0002 i que cursa 5 assignatures amb un preu de 100,00 euros cadascuna.
- Marc de 23 anys, estudiant del màster d'Intel·ligència Artificial amb un preu de matrícula de 300,00 euros, que té un carnet amb el número 0003 i que té beca.

Recorre aquesta llista fent servir un iterador i invoca el mètode imprimirPreuMatricula() per a cada element.

c) Indica quina és la sortida del mètode main.

```
//*********************************************************************
// Estudiant.java
//*********************************************************************
package edu.ub.prog2.activitat;
public abstract class Estudiant {
        protected String nom;
        protected int edat;
        protected int numCarnet;
```

```
        //Constructors
            public Estudiant(String pNom, int pEdat, int pNumCarnet) {
                        nom = pNom;
                        edat = pEdat;
                        numCarnet = pNumCarnet;
            }
        // Mètodes consultors:
            public String getNom() {
                        return nom;
        }
            public int getEdat() {
                        return edat;
        }
        public int getNumCarnet() {
                        return numCarnet;
        }

        // Mètodes modificadors:
            public void setNom(String pNom) {
                         nom = pNom;
        }
            public void setEdat(int pEdat) {
                        edat = pEdat;
        }
            public void setNumCarnet(int pNumCarnet) {
                         numCarnet = pNumCarnet;
        }

        //Mètode abstracte, es declara sense implementació
        public abstract void imprimirPreuMatricula();
}


//*********************************************************************
// EstudiantGrau.java
//*********************************************************************
package edu.ub.prog2.activitat;
public class EstudiantGrau extends Estudiant{
    private String nomGrau;
    private int numAssig;
    private double preuAssig;

    //Constructor
    public EstudiantGrau(String pNom, int pEdat, int pNumCarnet, String pNomGrau, int
pNumAssig, double pPreuAssig) {
        super(pNom, pEdat, pNumCarnet); // aquesta crida sempre al principi del mètode.
        nomGrau = pNomGrau;
        numAssig = pNumAssig;
        preuAssig = pPreuAssig;
    }
//Comentaris:
// No es pot definir un constructor sense paràmetres si no he definit a la classe pare un
// constructor sense paràmetres.
// S'ha d'utilitzar super, ja que sinó es crida al constructor per defecte de la classe pare
// Estudiant() i no està disponible perqué s'ha sobreescrit amb el constructor amb arguments

    // Mètodes consultors:
    public String getNomGrau() {
```

```
        return nomGrau;
    }
    public int  getNumAssig() {
        return numAssig;
    }
    public double getPreuAssig() {
        return preuAssig;
    }

    // Mètodes modificadors:
    public void setNomGrau(String pNomGrau) {
         nomGrau = pNomGrau;
    }
    public void setNumAssig(int pNumAssig) {
        numAssig = pNumAssig;
    }
    public void setPreussig(double pPreuAssig) {
        preuAssig = pPreuAssig;
    }

    //Implementació del mètode abstracte de la classe pare Estudiant
    public void imprimirPreuMatricula() {
       double preuMatricula;
       preuMatricula = numAssig * preuAssig;
       System.out.println("El preu de la matrícula de l'estudiant de grau " + nom + " és " +
preuMatricula);
    }
}

//*********************************************************************
// EstudiantMaster.java
//*********************************************************************

package edu.ub.prog2.activitat;
public class EstudiantMaster extends Estudiant{
    private String nomMaster;
    private double preuMaster;
    private boolean becat;

    //Constructor
    public EstudiantMaster(String pNom, int pEdat, int pNumCarnet, String pNomMaster, double
pPreuMaster, boolean pBecat) {
       super(pNom, pEdat, pNumCarnet);
       nomMaster = pNomMaster;
       preuMaster = pPreuMaster;
       becat = pBecat;
    }

    // Mètodes consultors:
    public String getNomMaster() {
        return nomMaster;
    }
    public double getPreuMaster() {
        return preuMaster;
    }
    public boolean getBecat() {
        return becat;
    }
```

```
    // Mètodes modificadors:
    public void setNomMaster(String pNomMaster) {
            nomMaster = pNomMaster;
    }
    public void setPreuMaster(double pPreuMaster) {
            preuMaster = pPreuMaster;
    }
    public void setBecat(boolean pBecat) {
            becat = pBecat;
    }

    //Implementació del mètode abstracte de la classe pare Estudiant
    public void imprimirPreuMatricula() {
        double preuMatricula;
        if(becat){
            preuMatricula= 0;
        }else{
            preuMatricula= preuMaster;
        }
        System.out.println("El preu de la matrícula de l'estudiant de màster " + nom + " és "+
preuMatricula);
             }
}
//*********************************************************************
// Activitat.java
//*********************************************************************
package edu.ub.prog2.activitat;
import java.util.ArrayList;
import java.util.Iterator;

public class Activitat {
    static public void main(String[] args){
        // Creem tres instàncies d'estudiants:
        Estudiant est1 = new EstudiantGrau("Maria", 18, 0001, "Matemàtiques", 4, 100.00);
        Estudiant est2 = new EstudiantGrau("Joan", 20, 0002, "Informàtica", 5, 100.00);
        Estudiant est3 = new EstudiantMaster("Marc", 23, 0003, "Intel·ligència Artificial", 300.00,
true);

         // Creem una llista genèrica d'estudiants:
        ArrayList<Estudiant> llistaInscrits = new ArrayList<Estudiant>();

        // Omplim la llista amb les tres instàncies:
        llistaInscrits.add(est1);
        llistaInscrits.add(est2);
        llistaInscrits.add(est3);

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

Sortida:

```
El preu de la matrícula de l'estudiant de grau Maria és 400.0
El preu de la matrícula de l'estudiant de grau Joan és 500.0
El preu de la matrícula de l'estudiant de màster Marc és 0.0
```