

GRAU D'ENGINYERIA INFORMÀTICA

PROGRAMACIÓ II

Bloc 2:

Programació Orientada a Objectes (2)

Laura Igual

Departament de Matemàtiques i Informàtica

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

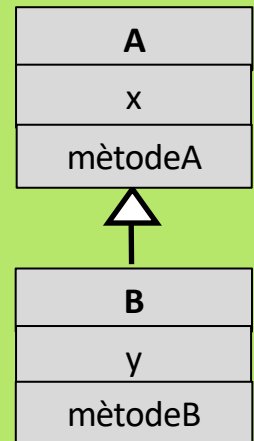


UNIVERSITAT DE
BARCELONA

Exercici 1: Fes de compilador!

- 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:

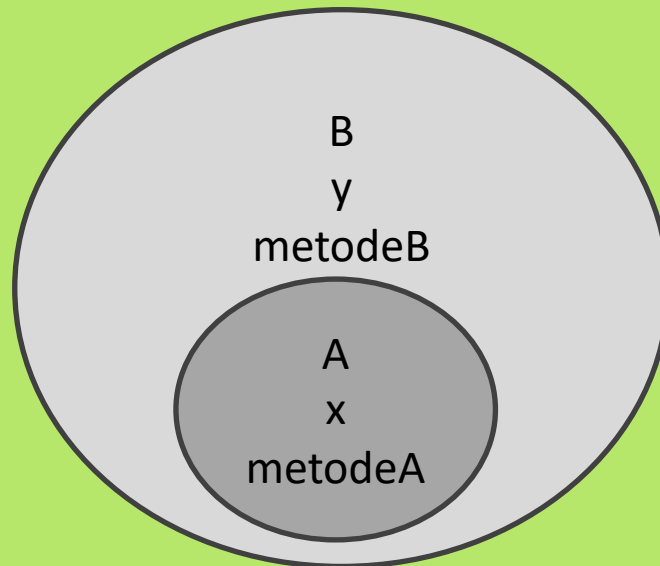
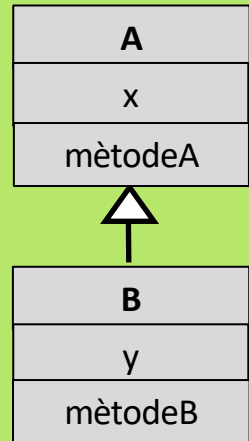
```
public class A{  
    public int x;  
    public void metodeA() {  
        ....  
    }  
}  
  
public class B extends A{  
    public int y;  
    public void metodeB() {  
        ....  
    }  
}
```



Exercici 1: Fes de compilador!

Indica a cada una de las línies del següent codi si haurà errors de compilació o no i explica breument perquè:

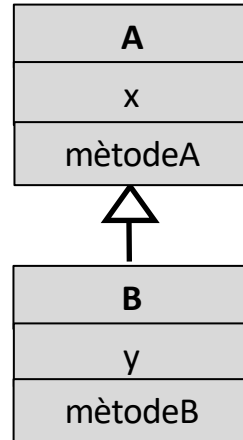
```
0 public static void main(String[] args) { var1 →
1   A var1 = new A();
2   B var2 = new B();
3   int i = var1.x;
4   int j = var2.x;
5   int k = var1.y;
6   int l = var2.y;
7   var1.metodeA();
9   var1.metodeB();
10  var2.metodeA();
11  var2.metodeB();
12 }
```



Exercici 1: Fes de compilador!

Indica a cada una de las línies del següent codi si haurà errors de compilació o no i explica breument perquè:

```
0 public static void main(String[] args) {  
1   A var1 = new A(); OK  
2   B var2 = new B(); OK  
3   int i = var1.x;    OK  
4   int j = var2.x;    OK  
5   int k = var1.y; Error, l'atribut y no està definit per a A.  
6   int l = var2.y;    OK  
7   var1.metodeA();    OK  
9   var1.metodeB(); Error, el metodeB no està definit per a A  
10  var2.metodeA();    OK  
11  var2.metodeB();    OK  
12 }
```



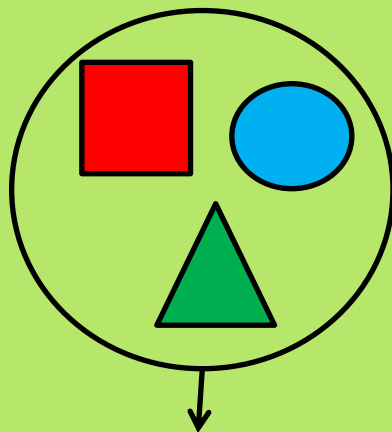
Exercici 2

Implementeu el següent disseny, considerant:

- **Classe abstracta:** Figura geomètrica
- Classe: Triangle, quadrat, cercle, ...

Figura
color posició a pantalla àrea perímetre
calcula àrea calcula perímetre retorna color assigna color retorna posicio assigna posicio

Atributs i mètodes heretats



Figures geomètriques

Quadrat	Circumferència	Triangle
color posició a pantalla àrea perímetre costats	color posició a pantalla àrea perímetre radi	color posició a pantalla àrea perímetre dimensió 3costats
calcula àrea calcula perímetre retorna color assigna color retorna posicio assigna posicio	calcula àrea calcula perímetre retorna color assigna color retorna posicio assigna posicio	calcula àrea calcula perímetre retorna color assigna color retorna posicio assigna posicio

Solució

```
public abstract class Figura {  
    private String color;  
    protected double x, y;  
    protected double area;  
    protected double perimetre;
```

```
// Mètodes abstractes:  
public abstract double calculaArea();  
public abstract double calculaPerimetre();
```

```
//Retorna el Color  
public String getColor(){  
    return color;  
}  
//Assigna el Color  
public void setColor(String color){  
    this.color=color;  
}
```

Figura.java

```
//Retorna la posició de la Figura  
public double [] getPosicion(){  
    double [] posicioxy = {x, y};  
    return posicioxy;  
}  
//Assigna la posició de la Figura  
public void setPosicio(double[] posicioxy){  
    x=posicioxy[1];  
    y=posicioxy[2];  
}  
} // Final de la classe Figura
```

Quadrat.java

```
public class Quadrat extends Figura {  
    private double costat; // longitud dels costats  
    // constructors  
    public Quadrat() {  
        costat=0.0;  
    }  
    public Quadrat(double costat) {  
        this.costat = costat;  
    }  
    // Calcula l'àrea del quadrat:  
    public double calculaArea() {  
        area = costat * costat ;  
        return area;  
    }  
    // Calcula el valor del perímetre:  
    public double calculaPerímetre(){  
        perímetre = 4 * costat;  
        return perímetre;  
    }  
}
```

Cercle.java

```
public class Cercle extends Figura {  
    public static final double PI=3.14159265358979323846;  
    public double radi;  
    // constructors  
    public Cercle(double x, double y, double radi) { crearCercle(x,y,radi); }  
    public Cercle (double radi) { crearCercle(0.0,0.0,radi); }  
    public Cercle (Cercle c){ crearCercle(c.x,c.y,c.radi); }  
    public Cercle() { crearCercle(0.0, 0.0, 1.0); }  
    // Mètode de suport  
    private void crearCercle(double x, double y, double radi) {  
        this.x=x; this.y=y; this.radi =radi;  
    }  
    // calcula l'area del cercle  
    public double calculaArea() {  
        area = PI * radi * radi;  
        return area;  
    }  
    // calcula el valor del perímetre  
    public double calculaPerimetre() {  
        perimetre = 2 * PI * radi;  
        return perimetre;  
    }  
} // fi de la classe Cercle
```

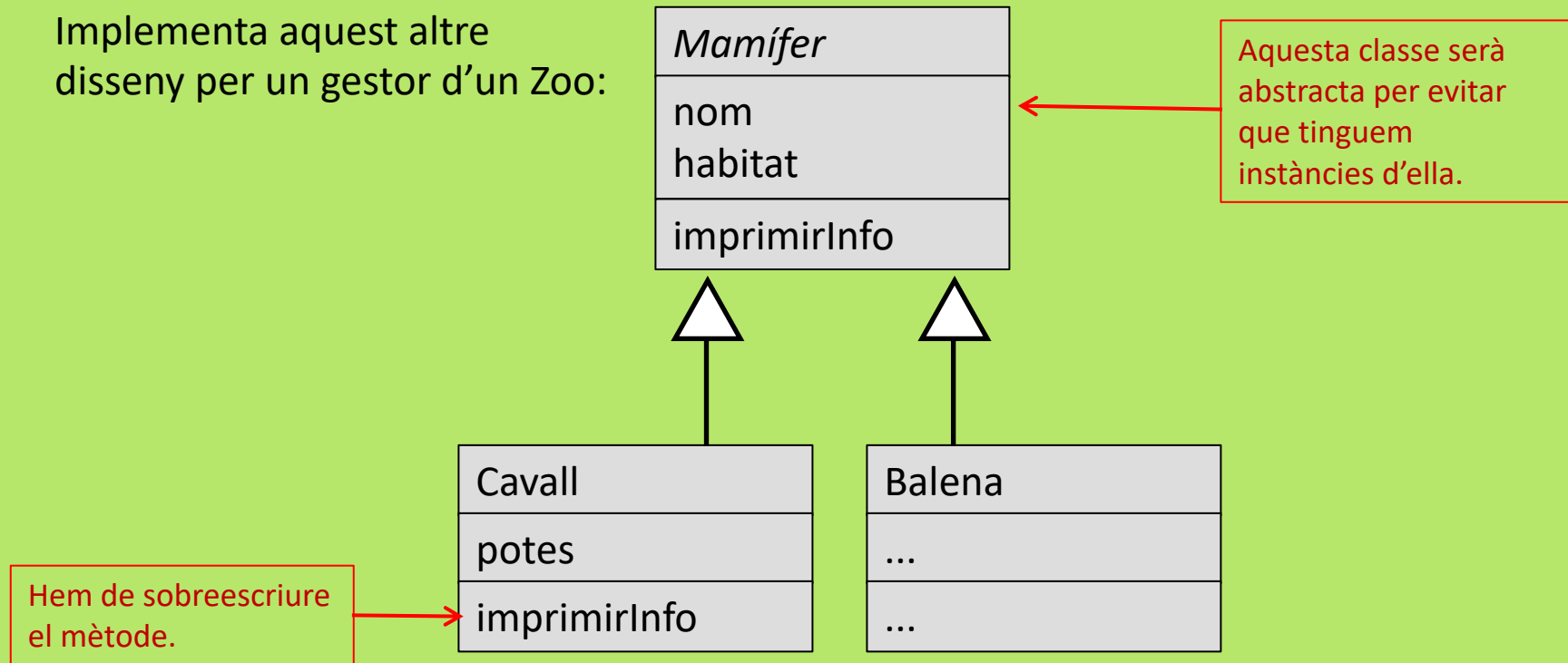

Cercle.java

```
public class Cercle extends Figura {  
    public static final double PI=3.14159265358979323846;  
    public double radi;  
    // constructors  
    public Cercle(double x, double y, double radi) {  
        crearCercle(x,y,radi);  
    }  
    public Cercle (double radi) {  
        crearCercle(0.0,0.0,radi);  
    }  
    public Cercle (Cercle c){  
        double [] pos = c.getPosicio();  
        crearCercle( pos[0],pos[1],c.getRadi());  
    }  
    public Cercle() {  
        crearCercle(0.0, 0.0, 1.0);  
    }  
    // Mètode de suport  
    private void crearCercle(double x, double y, double radi) {  
        setPosico({x,y});  
        setRadi(radi);  
    }  
} // fi de la classe Cercle
```

O millor amb getters i setters...

Exercici 3 (Sobreescritura)

Implementa aquest altre disseny per un gestor d'un Zoo:



Mamífer serà una classe abstracta, ja que hi ha informació d'aquesta classe que no es pot conèixer sense especificar més (saber més sobre l'animal).

Exemple: l'habitat de l'animal: terrestre o marí.

Mamifer.java

```
public abstract class Mamifer {  
    private String nom, habitat;  
    public Mamifer(String nom) {  
        this.nom = nom;  
        this.habitat = "Desconegut"  
    }  
    public void imprimirInfo() {  
        System.out.println("Nom" + getNom()+ ", habitat" + getHabitat()\n");  
    }  
}  
public class Cavall extends Mamifer {
```

Gos.java

```
    private int potes;  
    public Cavall(String nom) {  
        super(nom);  
        setHabitat("Terrestre");  
        potes=4;  
    }  
    public void imprimirInfo() {  
        super.imprimirInfo();  
        System.out.println("Té " + getPotes() + " potes. \n");  
    }  
}  
public class Balena extends Mamifer {
```

Balena.java

```
    public Balena(String nom) {  
        super(nom);  
        setHabitat("Aquàtic");  
    }  
}
```

Exercici

Què surt per pantalla?

```
public class CreaExemples {                                CreaExemple.java
    public static void main(String [] args) {
        Gos bobi = new Gos("Bobi");
        bobi.imprimirInfo();
        Balena bal= new Balena("Nemo");
        bal.imprimirInfo(); /*Està a la classe Mamífer*/
    }
}
```

—————→ Nom Bobi, habitat Terrestre
Té 4 potes.
Nom Nemo, habitat Aquàtic