

UF09.- POO (II)

- Teoria -

PROGRAMACIÓ
CFGs DAW

Joan V. Cassany

jv.cassanycoscolla@edu.gva.es

2022/2023 1

1. LA CLASSE ARRAYLIST

DECLARACIÓ



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



Array



ArrayList

Un *ArrayList* és una estructura de dades dinàmica del tipus **col·lecció** que implementa una llista de grandària variable.

```
import java.util.ArrayList;
```

Com es tracta d'una classe l'hem d'importar.

```
ArrayList<Element> llista = new ArrayList<Element>();
```

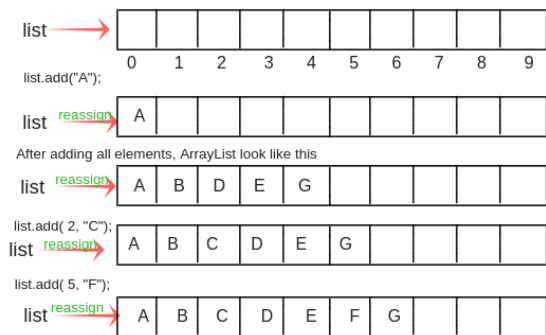
Declaració.

1. LA CLASSE ARRAYLIST

MÈTODES



GENERALITAT
VALENCIANA



- **int size();** nombre d'elements de la llista.
- **E get(int index);** element en la posició index.
- **void clear();** elimina tots els elements de la llista.
- **boolean isEmpty();** true si la llista no conté elements.
- **boolean add(E element);** inserida element al final de la llista.
- **void add(int index, E element);** inserida element en una posició de la llista.
Desplaça una posició tots els altres elements de la llista.
- **void set(int index, E element);** substitueix l'element en la posició index.
- **boolean contains(Object o);** cerca l'objecte en la llista.
- **int indexOf(Object o);** retorna l'índex on se trobe l'objecte.
- **int lastIndexOf(Object o);** com indexOf() però des del final de la llista.
- **E remove(int index);** elimina l'element en la posició.
- **boolean remove(Object obj);** elimina la primera ocurrència de l'objecte.

NOTA: Practica amb l'Exemple01 dels Recursos Complementaris

1. LA CLASSE ARRAYLIST

ARRAYLIST D'OBJECTES: EXEMPLES



GENERALITAT
VALENCIANA



```
gossos.add (new Gos ("Scooby", "marró", "dogo", Sexe.MASCLE, 5, 40));
```

```
System.out.println ("El gos que hi ha en la posició 0 és " + gossos.get(0).diMeNom());
```

```
Gos buscar = new Gos ("Milú", "blanc", "fox terrier", Sexe.MASCLE, 8, 4);
```

```
if (gossos.contains (buscar)){
```

```
    System.out.println ("El gos " + buscar.diMeNom() +
```

```
    " es troba en la posició " + (gossos.indexOf (buscar));
```

```
}
```

```
gossos.remove (buscar);
```

1. LA CLASSE ARRAYLIST

RECORRER UNA LLISTA: ITERATOR

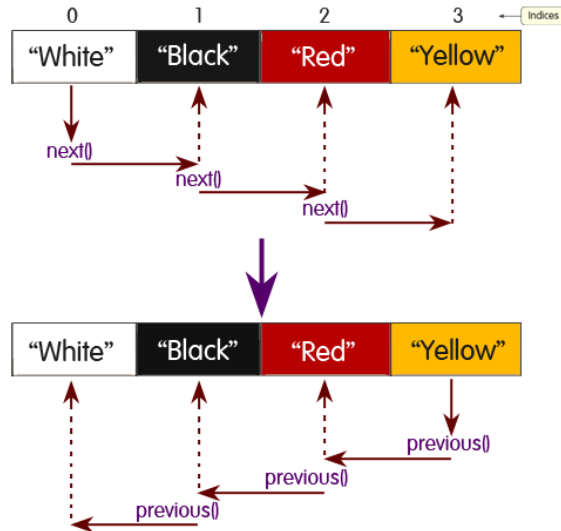


GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

```
import java.util.Iterator;
```

Usant un objecte *Iterator* que permet recórrer llistes com si fora un índex. Es necessita importar la classe.



```
Iterator iter = colors.iterator();
```

```
while (iter.hasNext()) {
```

```
    System.out.print (iter.next());
```

```
}
```

1. LA CLASSE ARRAYLIST

ORDENAR UN ARRAYLIST D'OBJECTES



GENERALITAT
VALENCIANA



EN LA CLASSE

- Hem de dir que és comparable.

```
public class Classe implements Comparable <Classe> { .. }
```

- Inclourem el mètode **compareTo** per a establir el criteri d'ordenació.

```
public int compareTo (Classe objecte) {  
    return (this.tribut).compareTo(objecte.diMeAtribut()); } }
```

EN EL PROGRAMA PRINCIPAL

- Importarem la classe **Collections**.

```
import java.util.Collections;
```

- Utilitzarem el mètode **sort**.

```
Collections.sort(llista);
```

NOTA: Practica amb l'Exemple02 dels Recursos Complementaris

2. COMPOSICIÓ

ESTÀ COMPOST PER



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

La composició és l'agrupament **d'un o diversos objectes i valors dins d'una classe.**

```
public class Punt {  
  
    int x, y;  
  
    public Punt (int x, int y){  
        this.x=x;  
        this.y=y;  
    }  
}
```

```
import java.lang.Math;  
public class Rectangle {  
    Punt p1, p2;  
    String color;  
  
    // Constructor  
    public Rectangle(Punt punt1, Punt punt2, String deColor) {  
        if ((punt1.x < punt2.x) && (punt1.y < punt2.y)) {  
            this.p1 = punt1;  
            this.p2 = punt2;  
            this.color=deColor;  
        } else { System.err.println("Punts del rectangle incorrectes."); }    }  
  
    // Obtindre l'àrea  
    public int getArea()  
        { return Math.abs(this.p1.x - this.p2.x) * Math.abs(this.p1.y - this.p2.y); }  
}
```

NOTA: Practica amb l'Exemple03 dels Recursos Complementaris i l'Exercici01 dels Exercicis

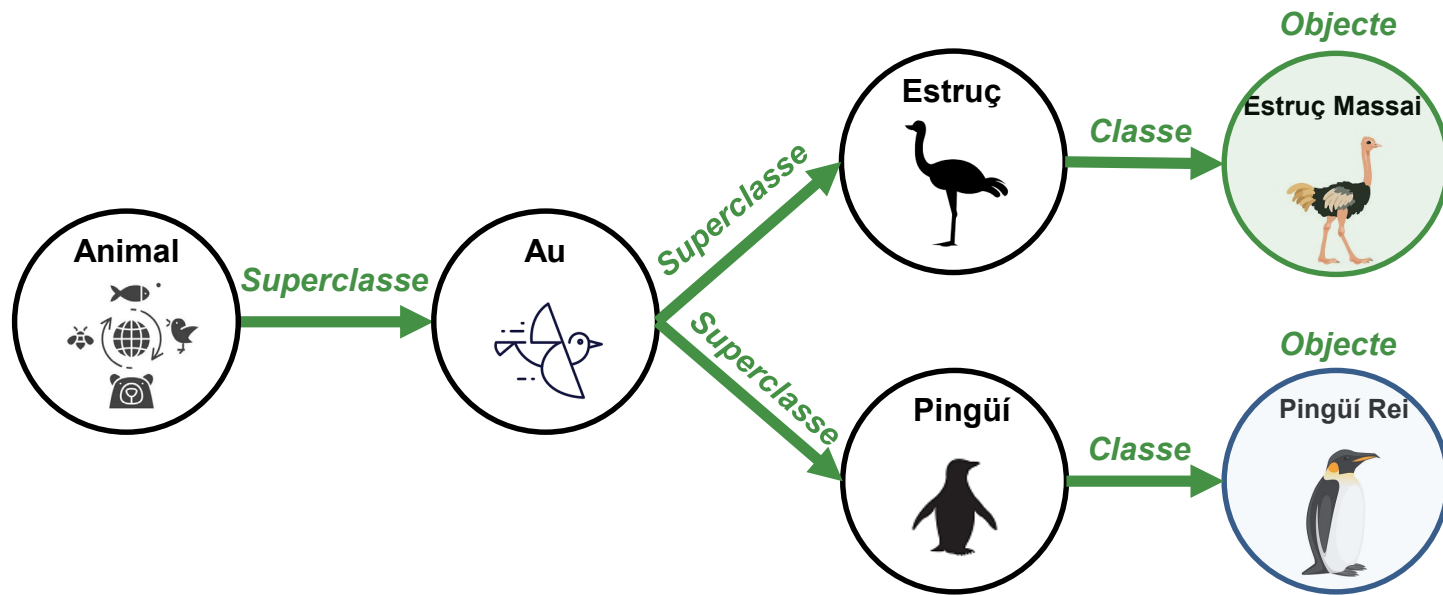
3. HERÈNCIA

INTRODUCCIÓ



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



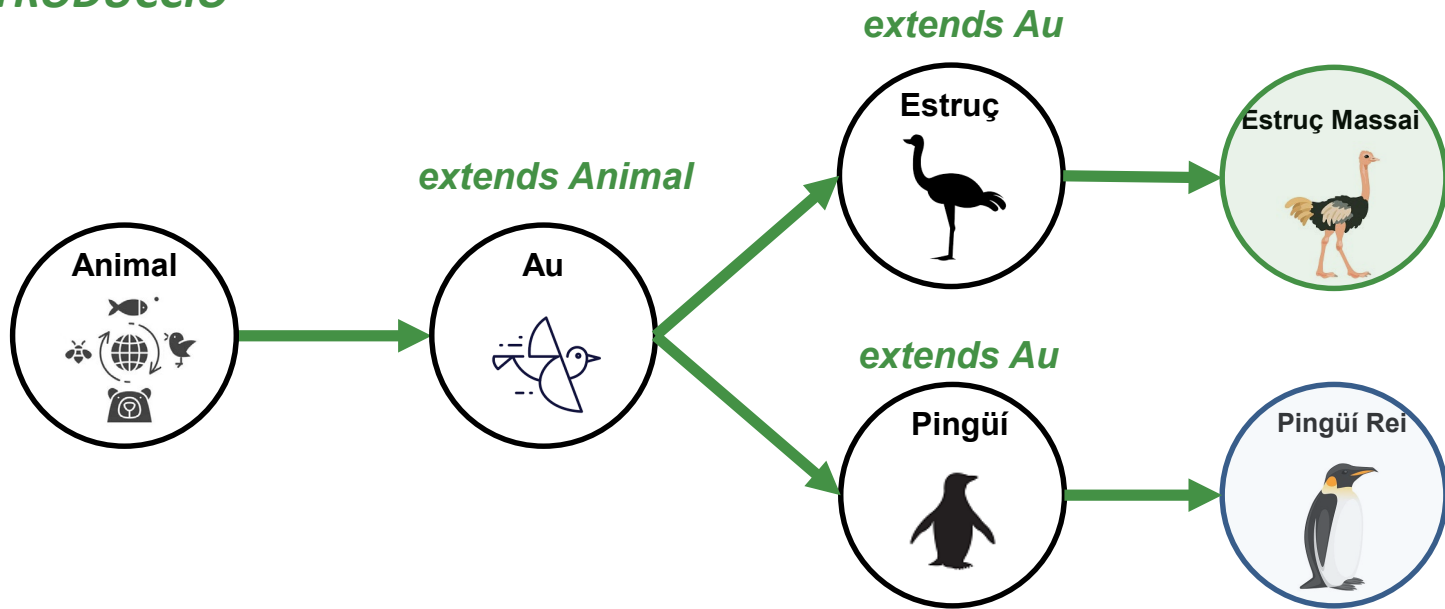
3. HERÈNCIA

INTRODUCCIÓ



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



- En la classe **Animal** tindrem atributs comuns a tots els animal
- La classe **Au** estén la classe **Animal** incorporant el seus atributs i incorporant altres nous.
- Les classes **Estruç** i **Pingüi** estenen la classe **Au** i incorpora els seus atributs, incloent els que li venen heretats des de la classe **Animal**.

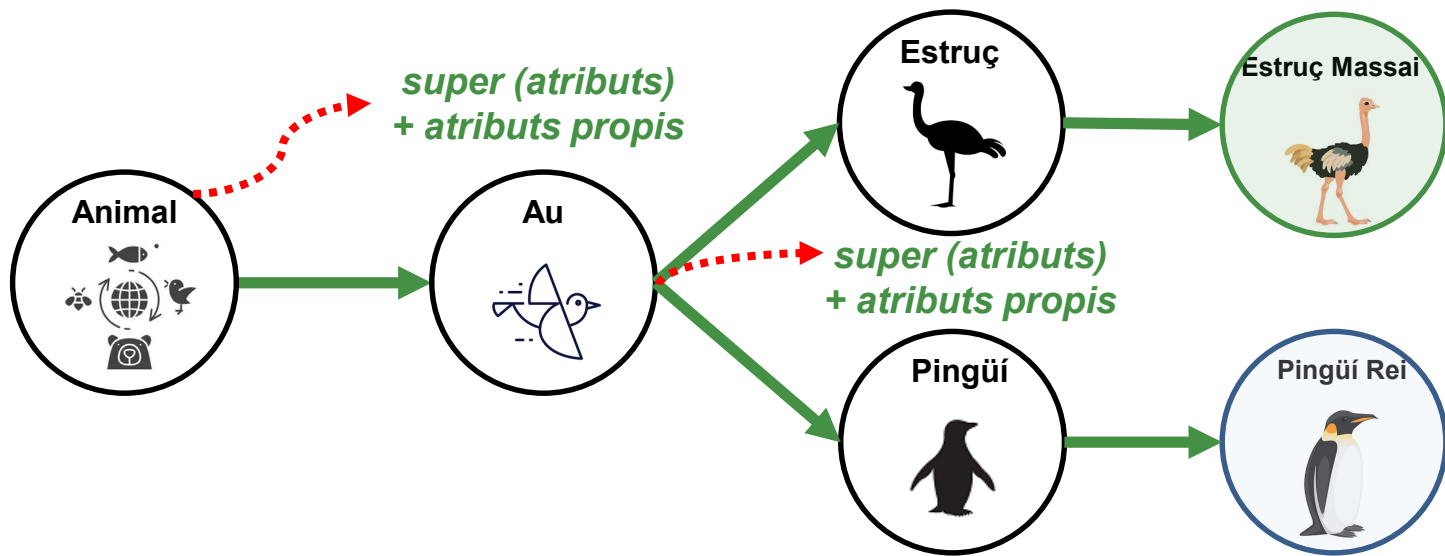
3. HERÈNCIA

CONSTRUCTOR DE LA CLASSE DERIVADA



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



- Dins del constructor de la classe derivada, per a cridar al constructor de la classe base es deu utilitzar el mètode reservat **super()** passant-li com a argument els paràmetres que necessite.
- **COMPTE:** si hi ha més d'un constructor en la superclasse s'ha de tindre molt en compte els paràmetres que s'incorporen al invocar al mètode **super()**.

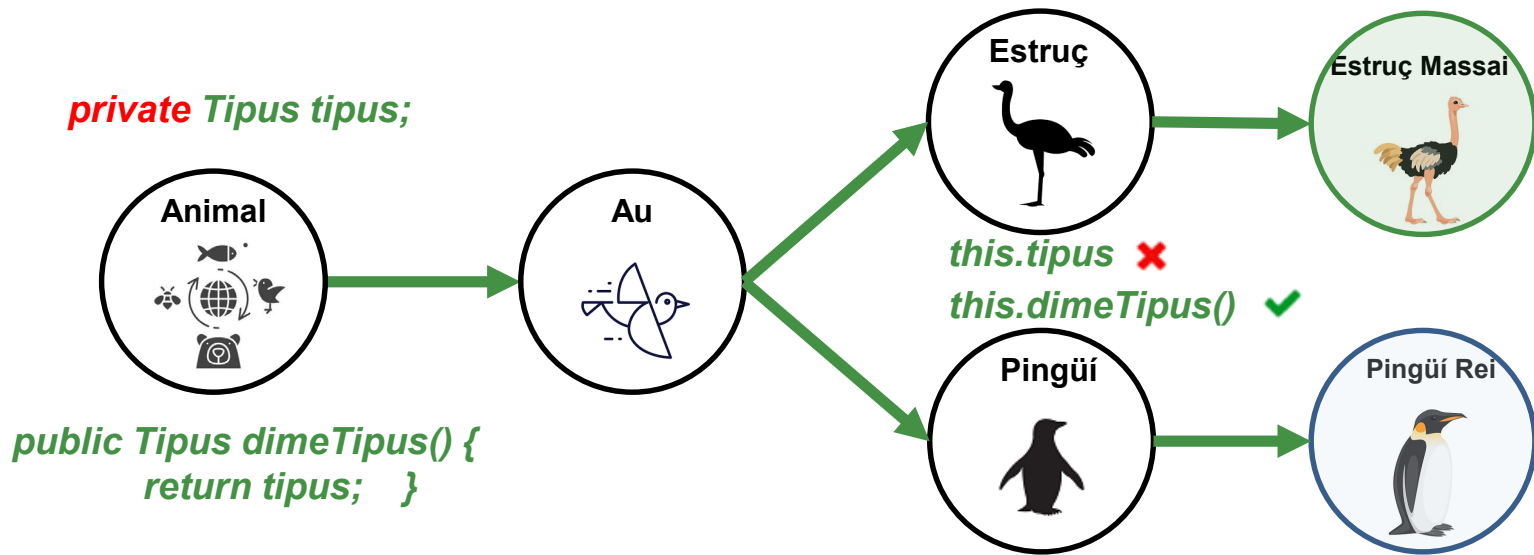
3. HERÈNCIA

ATRIBUTS I MÈTODES PRIVATS



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



- Les subclasses heretaran tots els atributs i mètodes excepte els declarats com **private** en la superclasse. Això no vol dir, però, que les subclasses no tinguin aquest atribut associat, sinó que no és possible accedir directament a aquest atribut. En realitat, els objectes de la subclasse poden establir valors per a aquest atribut a través dels mètodes de la superclasse.

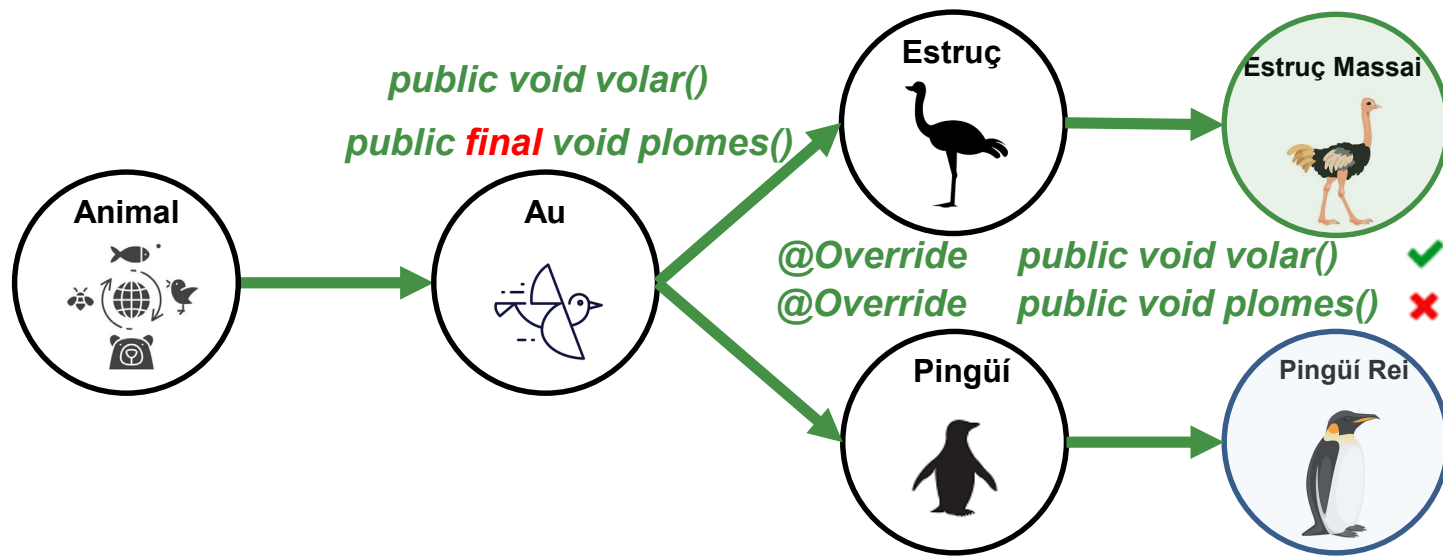
4. POLIMORFISME

SOBREESCRITURA I POLIMORFISME



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



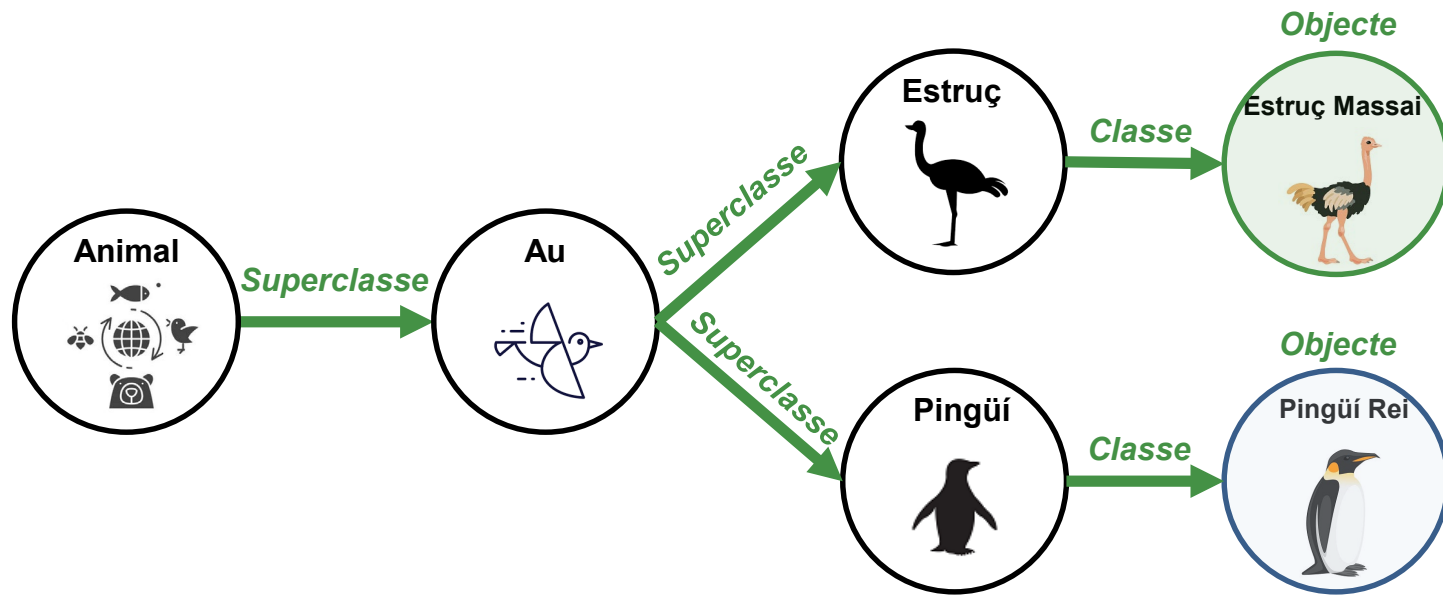
- Quan els mètodes definits en la superclasse no ens valen podem sobreescriure'ls o redefinir-los en la subclasse utilitzant la paraula reservada **@Override**.
- Polimorfisme és la capacitat que tenen els objectes de diferent classes de respondre al mateix mètode.
- Una classe **final** no pot ser heretada. Un mètode **final** no pot ser sobreescrit per les subclasses.
- Els atributs declarats com **protected** són públics per a les classes heretades i privats per a les altres classes.

4. HERÈNCIA I POLIMORFISME



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



REPASSA TOTS ELS CONCEPTES I PRACTICA

Saps com treballar amb un ArrayList amb totes aquestes classes mesclades?
Repasa el segon programa de l'Exemple04: **instanceof**

NOTA: Practica amb l'Exemple4 de la Teoria, l'Exemple04 dels Recursos Complementaris i l'Exercici02.

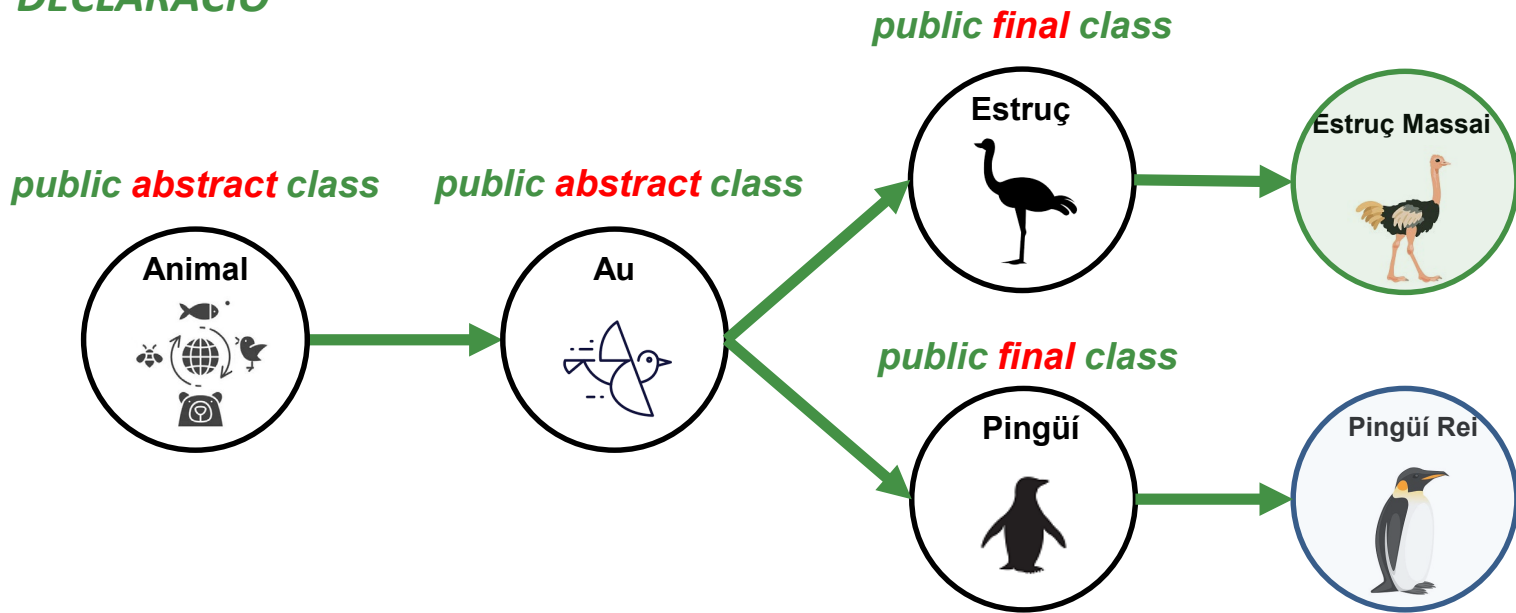
5. CLASSES ABSTRACTES

DECLARACIÓ



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



- Una classe abstracta és aquella que no va a tindre instàncies. Es a dir, no es pot instanciar.
- Les classes abstractes són útils quan volem definir una part general que continguen i siga compartida per totes les subclasses.
- Poden declarar-se **mètodes constructors** en una classe abstracta?
- Té sentit declarar una classe **abstract** com a **final**?

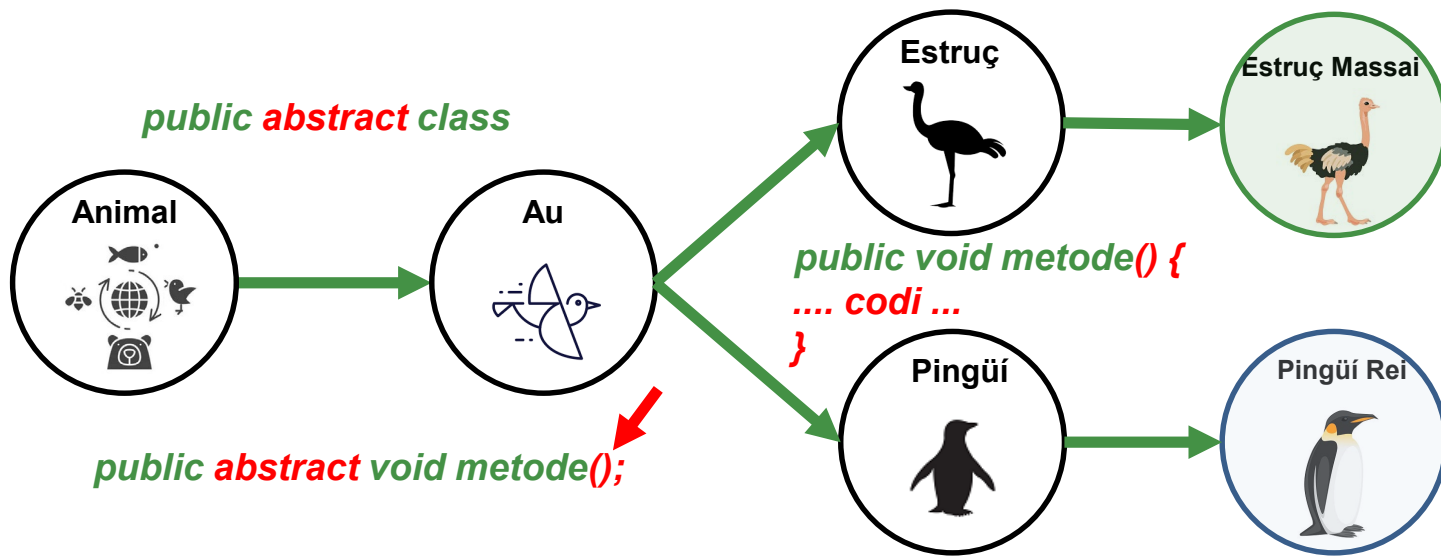
5. CLASSES ABSTRACTES

MÈTODES



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



- Els **mètodes constructors** en una classe abstract no són per a crear instàncies sinó **construir subclasses**.
- Els **mètodes abstractes** s'utilitzen per a fer a que totes les subclasses estiguen obligades a implementar un mètode amb eixe nom. No es tracta d'una sobreescritura (opcional) si no d'una restricció que obliga a la implementació en la subclasse.
- Les classes abstractes no poden tindre mètodes **estàtics abstractes**.

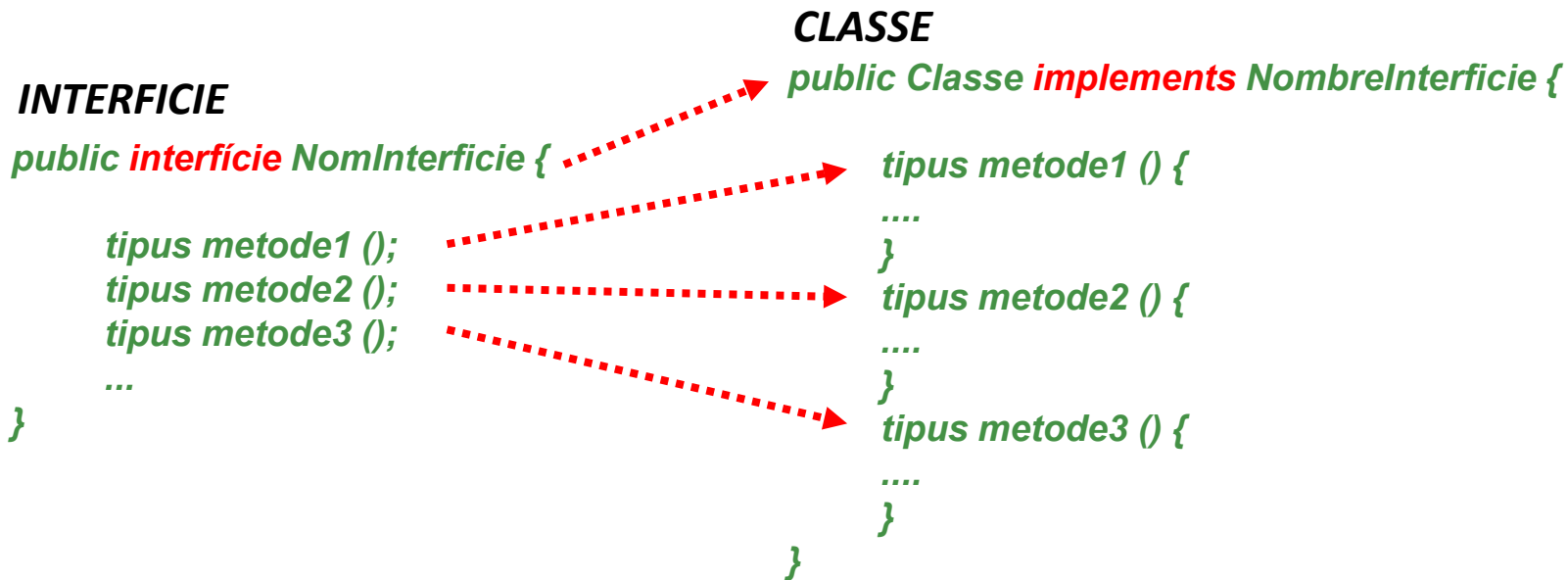
6. INTERFICIES

DEFINICIÓ



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA



- Una interfície s'encarrega d'**especificar** un comportament que després haurà de ser implementat.
- Una interfície serveix per a separar l'especificació del comportament que després serà implementat.
- La interfície determina el **qué** no el **com**.
- No és possible **instanciar** una interfície.
- Una mateixa classe pot **implementar** més d'una interfície.

6. INTERFICIES

ATRIBUTS I MÈTODES



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

INTERFICIE

```
public interfície NomInterficie {  
    double VALORPI=3.1416;  
    tipus metode1 ();  
    tipus metode2 ();  
    tipus metode3 ();  
    ...  
}
```

CLASSE

```
public Classe implements NombreInterficie {  
    public tipus metode1 () {  
        ....  
    }  
    public tipus metode2 () {  
        ....  
    }  
    public tipus metode3 () {  
        ....  
    }  
}
```

- Qualsevol **mètode** definit en una interfície se sobreentén que és **públic**, encara que no es pose explícitament la paraula pública davant seu.
- Qualsevol **atribut** definit en una interfície se sobreentén que és **públic, estàtic i final**, encara que no es posin explícitament les paraules public static final davant seu.

TAULA RESUM



GENERALITAT
VALENCIANA

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

	CLASSE	MÈTODE	ATRIBUT
ABSTRACT	No es pot instanciar. Val per a heretar-se. <i>Exemple: la classe Animal</i>	Indica el què no el com. Obliga a totes les subclasses a definir el mètode. La classe ha de ser abstracta. <i>Exemple: el mètode volar() en la classe Aus.</i>	
STATIC	Serveix per a obligar a que tots els seus mètodes siguen estàtics a més no es pot instanciar.	S'utilitza invocant a la classe sense haver de crear instàncies. Les classes abstractes no tenen mètodes estàtics. <i>Exemple: el mètode random() en la classe Math.</i>	Son atributs de classe, no d'instància. Totes les instàncies el comparteixen. <i>Exemple: el número de dorsal o el total de dorsals en la classe Participant.</i>
FINAL	No permet crear subclasses. <i>Exemple: les classes Estruç i Pingüí.</i>	No es pot sobreescrivre (@Override) en les subclasses. <i>Exemple: plomes() en la classe Aus.</i>	És una constant. <i>Exemple: PI.</i>
PRIVATE		Sols es pot utilitzar dins de la classe en la que es declara. Són procediments o funcions internes de la classe.	Sols els podem accedir des de dins de la classe. Necessitarem setters i getters per a accedir-los. <i>Exemple: en la classe Estruç no podem fer this.tipus sinó this.diMeTipus().</i>

EXERCICIS PROPOSATS

