

# Programació

## UT6.2. Classes abstractes i mètodes abstractes



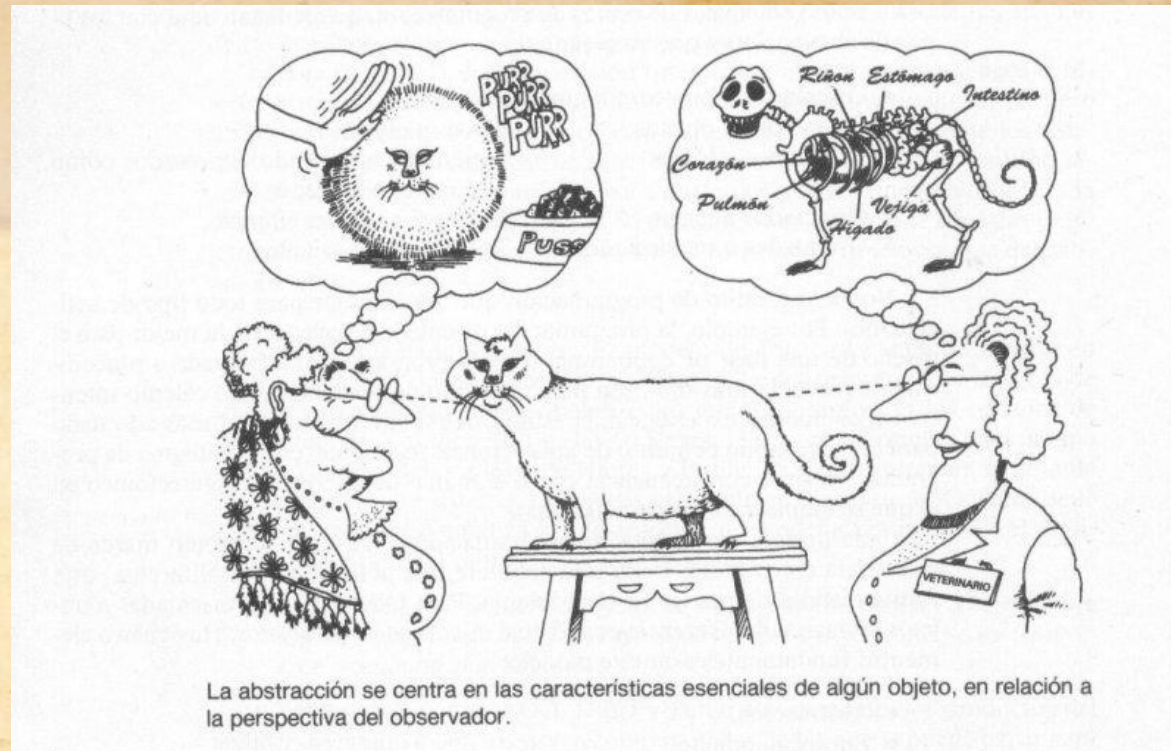
# Classes abstractes

- Fins ara hem treballat amb **classes concretes**.  
Estes apareixen de la descripció dels atributs i els mètodes que defineixen el comportament d'un cert conjunt d'objectes homogenis.
- Les **classes abstractes** són classes NO INSTÀNCIABLES que apareixen del factor comú del codi d'altres classes amb atributs i/o mètodes comuns, i/o capçaleres de mètodes comuns sense definició.



# Classes abstractes

- **ATENCIÓ:** No hem de pensar que sempre les classes superiors de la jerarquia seran sempre abstractes i les inferiors no. Dependrà de cada problema.



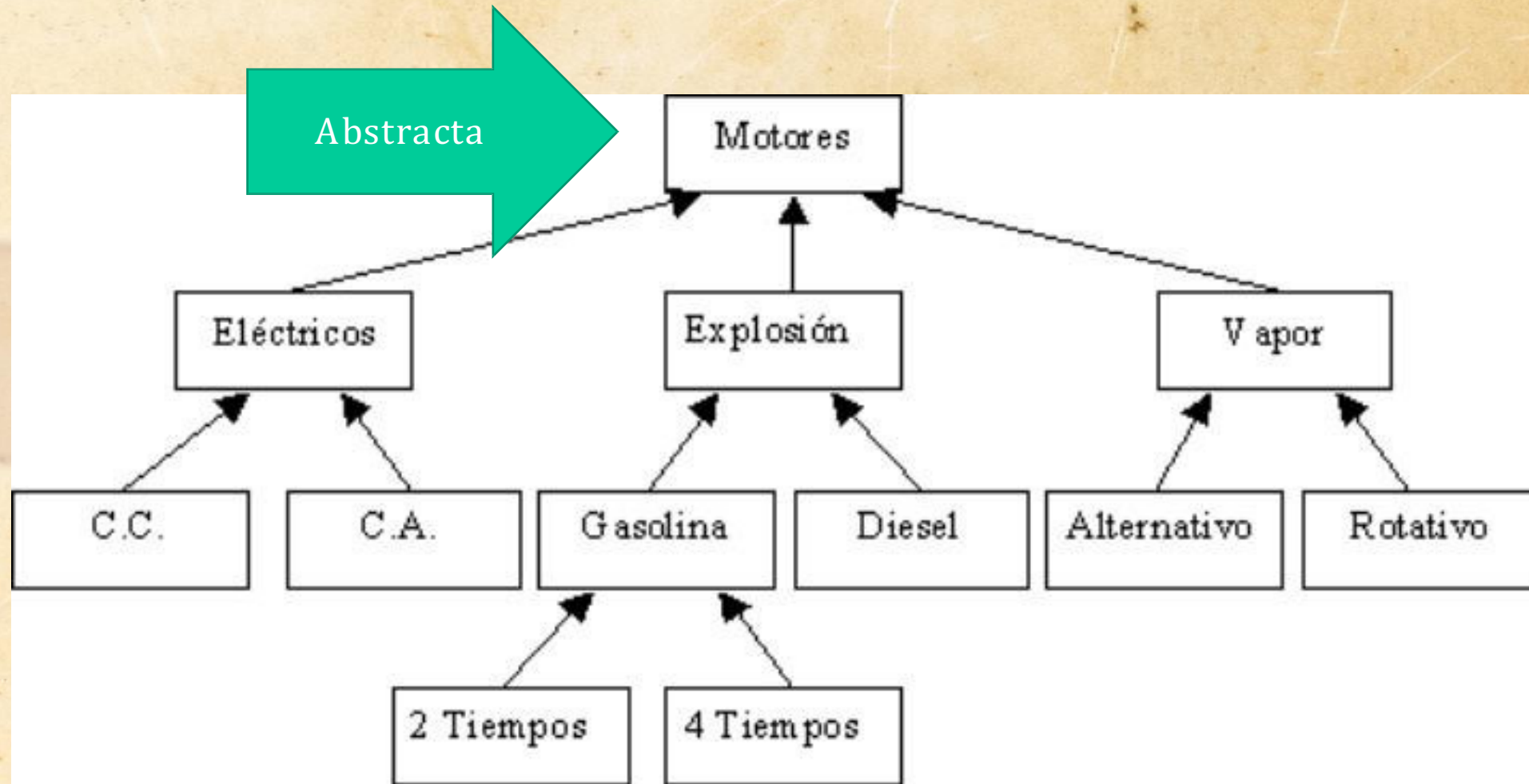


# Definició de classes abstractes en Java

```
public abstract class <NomClasse>{  
    <CosDeLaClase>  
}
```

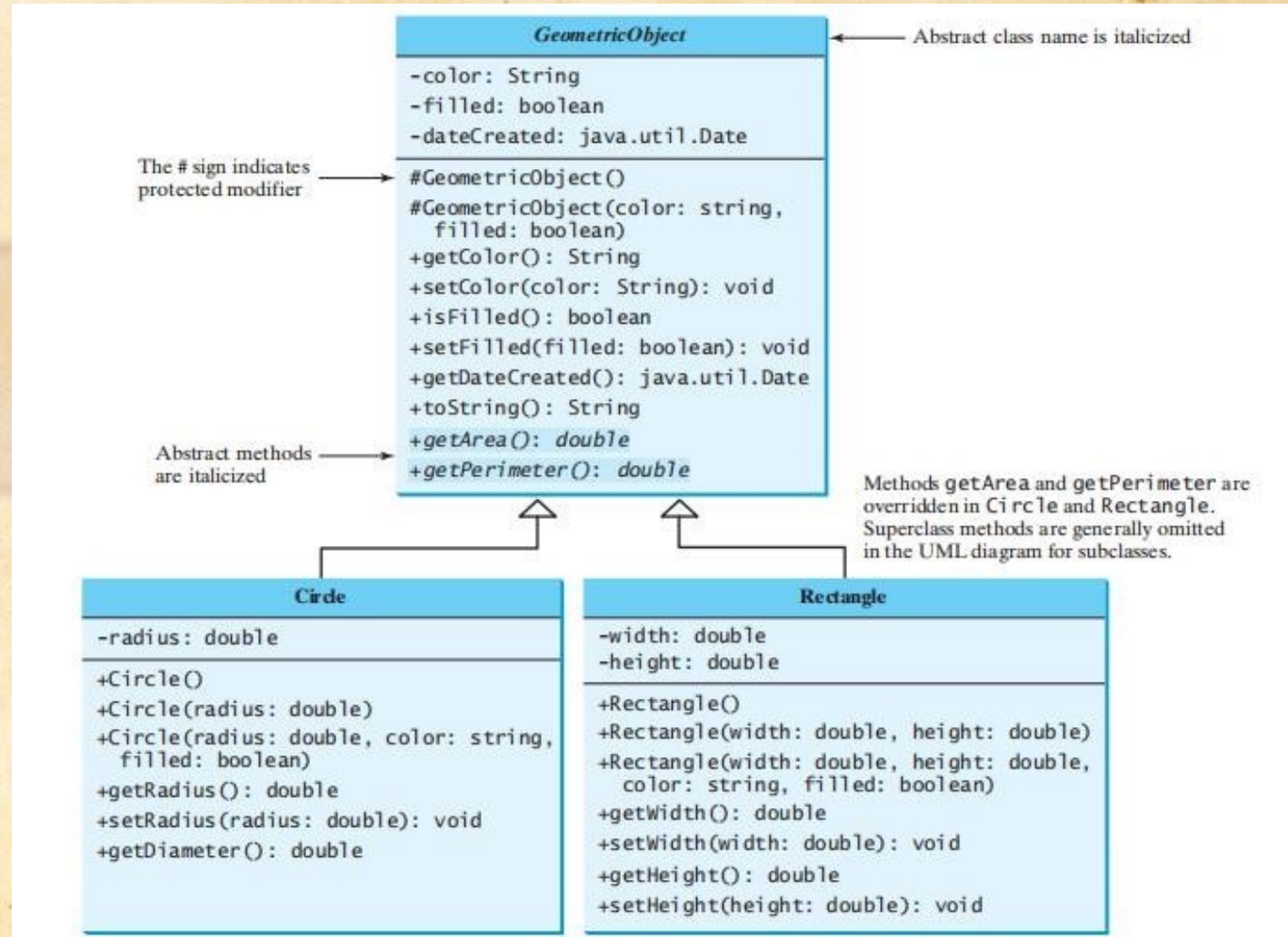
- La classe podrà alhora heretar d'una altra, sempre i quan l'altra també siga abstracta.

# Exemple





# Exemple FiguraGeometrica





# Mètodes abstractes

- A l'exemple de la diapositiva anterior, es defineixen dos mètodes que abans estaven a les classes `Circle` i `Rectangle` respectivament.
- Els mètodes `getArea()` i `getPerimeter()` són mètodes abstractes.
- Un mètode abstracte és un mètode declarat però no implementat, és dir, és un mètode del que només escrivim el seu nom, paràmetres i tipus retornat però NO el seu codi. Els mètodes abstractes s'escriuen sense claus `{}` i amb `;` al final de la declaració.



# Mètodes abstractes

El codi a executar en el mètode abstracte s'obtindrà a partir de la sobreescritura efectuada per alguna de les seues classes derivades.

## EJEMPLO

```
public abstract class Forma {  
    private int xpos, ypos;  
    private Color color;  
    //..  
    public abstract void dibuja();  
    public void setColor(Color c){ /*...*/ };  
}
```

Los métodos abstractos  
no tienen cuerpo

```
public class Circle extends Forma{  
    private int radio;  
    ... //  
    public void dibuja(){ /*...*/ };  
    public void setRadio(int){ /*...*/ };  
}
```

Dibuja un círculo

```
public class Rectangle extends Forma{  
    private int altura, anchura;  
    ... //  
    public void dibuja(){ /*...*/ };  
    public void setAltura(int){ /*...*/ };  
}
```

Dibuja un rectángulo



# Implicacions de els mètodes abstractes

- Qualsevol classe que continga alguna operació (mètode) abstracta **s'ha d'especificar com a classe abstracta.**
- En el moment d'especificar una classe derivada d'una classe amb operacions abstractes apareixen dos opcions:
  - Sobreescrivre en la classe derivada les operacions abstractes.
  - Sobreescrivre en la classe derivada part o cap de les operacions, fent la classe derivada també abstracta.



# Definició de mètodes abstractes en Java

```
visibilitat abstract tipusDeDada nomMetode(llistaParametres) ;
```

- **NO IMPLEMENTAREM** el mètode per la seua condició d'abstracte.
- Seran les classes derivades que hereten de la classe que continga eixe mètode les responsables d'implementar-lo.



# Quina seria una definició correcta d'una classe abstracta?

```
class A {  
    abstract void unfinished() {  
    }  
}
```

(a)

```
public class abstract A {  
    abstract void unfinished();  
}
```

(b)

```
class A {  
    abstract void unfinished();  
}
```

(c)

```
abstract class A {  
    protected void unfinished();  
}
```

(d)

```
abstract class A {  
    abstract void unfinished();  
}
```

(e)

```
abstract class A {  
    abstract int unfinished();  
}
```

(f)