



Ajuntament  
de Barcelona

# Objectes i classes

Java bàsic

IT Academy





# Una classe és

---

- **Una classe es la definició d'un tipus d'objecte**

**Indica:**

- Atributs que ha de tenir l'objecte**
- Mètodes que ha de tenir**

**(Es com una recepte per construir objectes)**



# Un objecte és

---

- Una encapsulació de:
  - Atributs: variables de l'objecte.
  - Mètodes: procediments i funcions de l'objecte



# Inicialitzar objectes

---

**Una classe defineix objectes però no els crea. Es creen quan s'inicialitzen:**

Exemple:

```
Alumne a = new Alumne("Josep", 9,"1982");
```

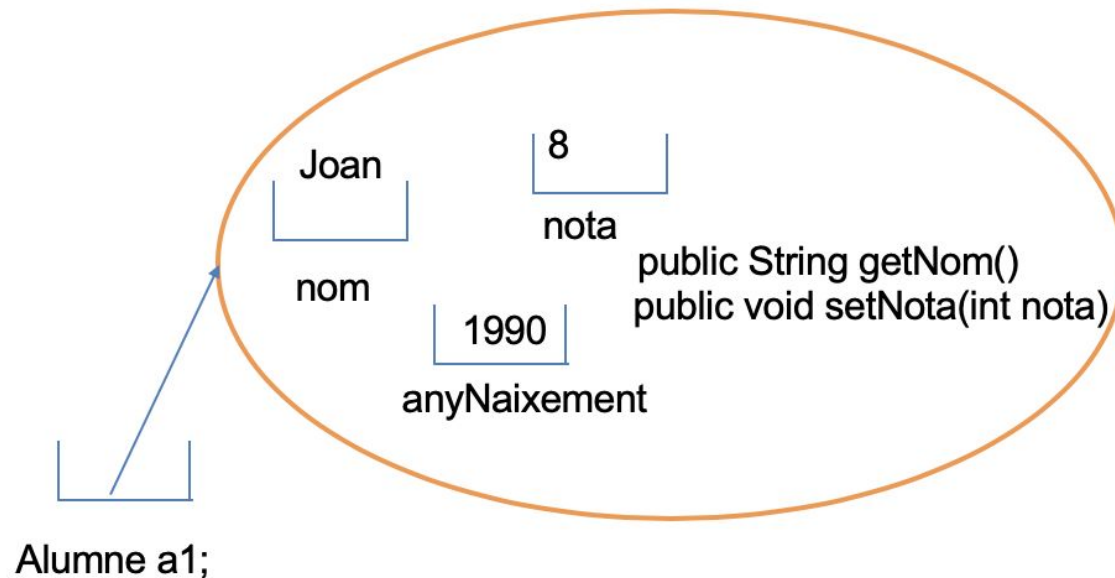
**S'inicialitza a través del constructor:**

```
//Constructor  
public Alumne (String nom, int nota, String anyNaixement){  
    this.nom=nom;  
    this.nota=nota;  
    this.anyNaixement=anyNaixement;  
}
```



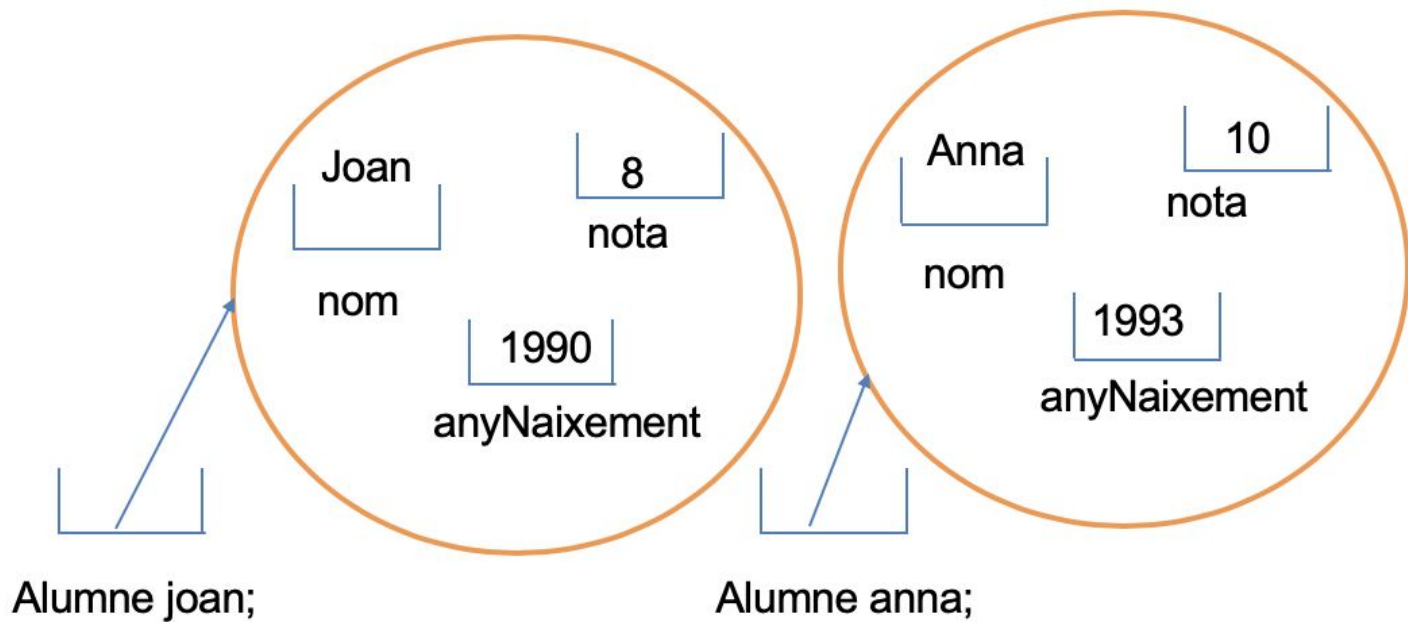
# Exemple

Un objecte Alumne té els següents atributs i mètodes:



# Exemple

Es pot tenir diferents instàncies del objecte, cadascuna amb diferents valors als atributs





# Visibilitat

---

**Un objecte només pot accedir i modificar els seus atributs.**

**Cada objecte té una copia del mètodes i de les variables.**

**Per modificar variables per tota la classe: static**



# Referència

---

Amb les variables tipus **Object** s'emmagatzema una referència, no un valor.

Diferència amb tipus bàsic:

```
int a,b,c;  
a=1;  
b=2;  
c=a;  
a--;  
Quant val c??
```

```
Alumne a1,a2,a3;  
a1 = new Alumne("Pep",9,"1943");  
a2 = new Alumne("Laura",7,"1968");  
a3=a1;  
a1.setNota(8);  
Quina nota té a3?
```





# Referència

---

```
int a,b,c;  
a=1;  
b=2;  
c=a;  
a--;  
Quant val c??
```

**c= 1**

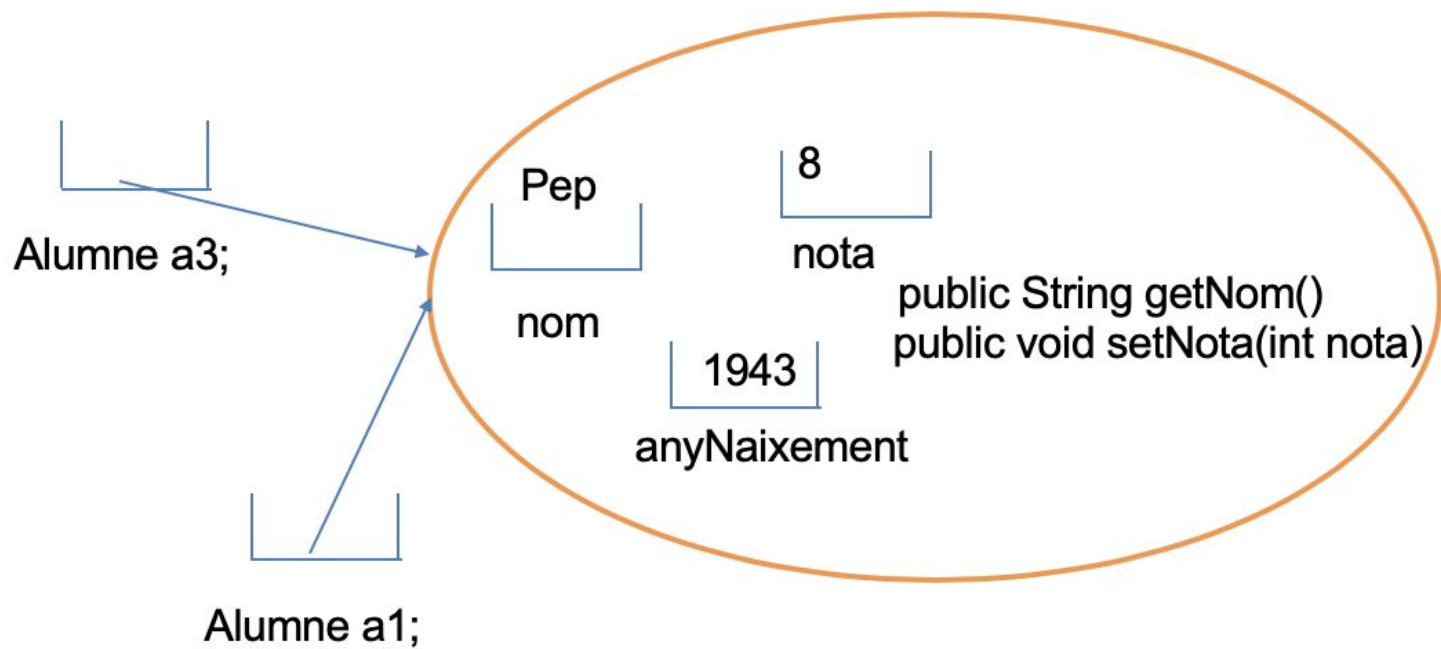
```
Alumne a1,a2,a3;  
a1 = new Alumne("Pep",9,"1943");  
a2 = new Alumne("Laura",7,"1968");  
a3=a1;  
a1.setNota(8);  
Quina nota té a3?
```

**a3.getNota() es 8**



# Referència

---



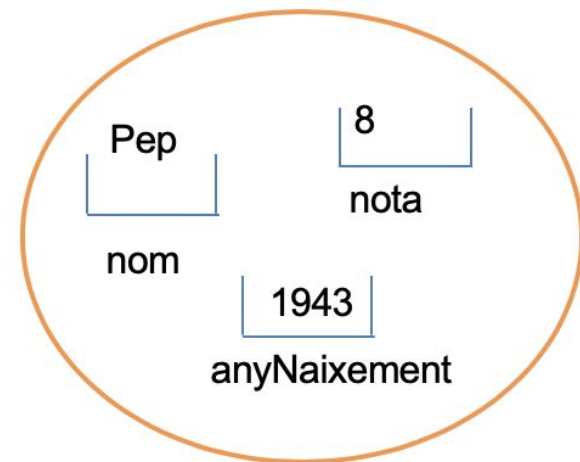


# Referència

**Si cap variable referencia a un objecte, aquest passa a ser inaccessible**

```
Alumne a1,a2,a3;  
a1 = new Alumne("Pep",9,"1943");  
a1 = null;
```

Alumne a1;





# This

---

Quan un paràmetre d'un mètode es diu exactament igual que un atribut, hi ha ocultació. Això vol dir, que l'atribut deixa de ser accessible.

```
private int a;  
public void mètode(int a){  
    a++; //es refereix al paràmetre  
    this.a = 0; // es refereix al atribut  
}
```

Per accedir al atribut s'ha d'usar “this”



# Objectes com a paràmetres

```
int nota=1;
```

```
alumne.ajustarNota(nota);
```



Quan val la variable nota aquí?

```
void ajustarNota(int nota){
```

```
    if(this.nota<5){  
        nota++;
```

```
    }  
    this.nota = nota;
```

```
}
```



I aquí?



# Objectes com a paràmetres

---

```
int nota=1;
```

```
alumne.ajustarNota(nota);
```



Quan val la variable nota aquí?

```
void ajustarNota(int nota){
```

```
    if(this.nota<5){  
        nota++;
```

```
    }  
    this.nota = nota;
```

```
}
```



I aquí?



# Objectes com a paràmetres

```
int nota=1;
```

```
alumne.ajustarNota(nota);
```



Quan val la variable nota aquí?

1

```
void ajustarNota(int nota){
```

```
    if(this.nota<5){  
        nota++;
```

```
    }
```

```
    this.nota = nota;
```

```
}
```



I aquí?

2



# Objectes com a paràmetres

---

**Les variables tipus primitiu passen el valor, no la referència.  
No es modifica a tot arreu. Amb els objectes sí.**

```
alumne2.setNota(0);  
alumne1.copiarNota(alumne2);  
alumne1.getNota();
```

Quan val la variable nota aquí?

```
void copiarNota(Alumne alumne){  
    this.nota = alumne.getNota();  
}
```

I aquí?





# Objectes com a paràmetres

**Les variables tipus primitiu passen el valor, no la referència.  
No es modifica a tot arreu. Amb els objectes sí.**

```
alumne2.setNota(0);  
alumne1.copiarNota(alumne2);  
alumne1.getNota();
```

Quan val la variable nota aquí?

0

```
void copiarNota(Alumne alumne){  
    this.nota = alumne.getNota();  
}
```

I aquí?

0



Barcelona  
**Activa**



[barcelona.cat/barcelonactiva](http://barcelona.cat/barcelonactiva)