

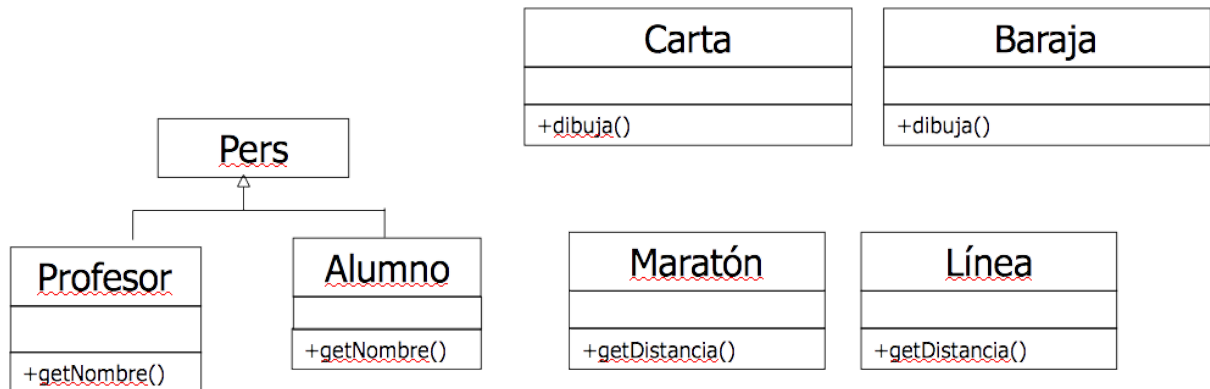
# U.D 6 - POLIMORFISMO I.

## Espacios de nombres

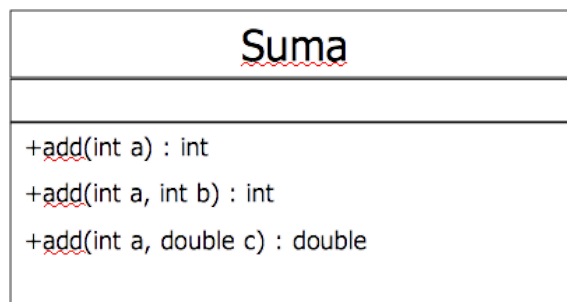
JAVA	C++
<pre><b>package graficos;</b> class Circulo {...}</pre>	<pre><b>namespace graficos</b> {     class Circulo {...} }</pre>
<pre>// sin import  class Main {     public static void main(...) {         graficos.Circulo c = new...;         c.pintar(System.out);     } }</pre>	<pre>// sin using  #include "graficos.h" int main() {     graficos::Circulo c;     c.setRadio(4); }</pre>
<pre>// import  import graficos.*; class Main {     public static void main(...) {         Circulo c=new...;         c.pintar(System.out);     } }</pre> <hr/> <pre>import graficos.Circulo;</pre>	<pre>// using namespace  #include "graficos.h" using namespace graficos; int main() {     Circulo c;     c.setRadio(4); }</pre> <hr/> <pre>using graficos::Circulo;</pre>

## Sobrecarga

Basada en ámbito:



Basada en signatura de tipo



## Downcasting (dinámico) seguro

```
class Animal {}  
class Perro extends Animal {  
    public void ladrar() {}  
}  
class PastorBelga extends Perro {}
```

Downcasting seguro usando Class directamente (pintar objetos en memoria):

```
Animal a = new PastorBelga();  
if (a.getClass() == PastorBelga.class) // cierto  
{ PastorBelga pb = (PastorBelga)a; }  
if (a.getClass() == Perro.class) // falso  
{ Perro p = (Perro)a; }
```

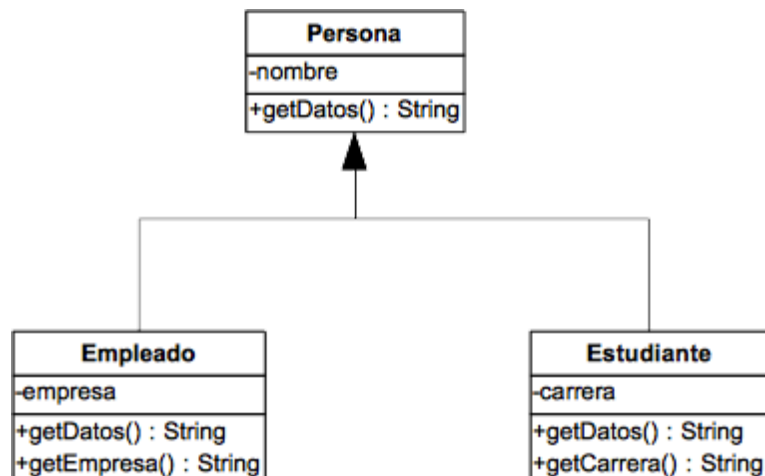
Downcasting seguro usando 'instanceof':

```
Animal a = new PastorBelga();  
if (a instanceof PastorBelga) // cierto  
{ PastorBelga pb = (PastorBelga)a; }  
if (a instanceof Perro) // cierto  
{ Perro p = (Perro)a; }
```

Downcasting seguro usando Class.isInstance():

```
Animal a = new Perro();  
Animal b = new PastorBelga();  
// En otro lugar...  
Class clasePerro = a.getClass();  
if (clasePerro.isInstance(b)) { //cierto  
    Perro p = (Perro)b; // seguro  
    p.ladrar();  
}
```

## Ejemplo de polimorfismo en herencia



```
class Persona {
    public Persona(String n)    {nombre=n;}
    public String getDatos() {return nombre;}
    ...
    private String nombre;
}

class Empleado extends Persona {
    public Empleado(String n,String e)
    { super(n); empresa=e; }
    public String getDatos()
    { return super.getDatos()+"trabaja en " + empresa; }
    ...
    private String empresa;
}

class Estudiante extends Persona {
    public Estudiante(String n,String c)
    { super(n); carrera=c; }
    public String getDatos()
    { return super.getDatos() + " estudia " + carrera; }
    ...
    private String carrera;
}
```

## Código cliente 1

```
Empleado empleado = new Empleado("Carlos","Lavandería");
Persona pers = new Persona("Juan");

empleado = pers;
```

```
System.out.println( empleado.getDatos() );
```

#### Código cliente 2

```
Empleado empleado = null;
Estudiante estudiante = null;
Persona pers = new Persona("José");
empleado = pers;
estudiante = pers;
System.out.println( empleado.getDatos() );
System.out.println( estudiante.getDatos() );
```

#### Código cliente 3

```
Empleado emp = new Empleado("Carlos", "lavanderia");
Estudiante est = new Estudiante("Juan", "Derecho");
Persona pers;
pers = emp;
System.out.println( emp.getDatos() );
System.out.println( est.getDatos() );
System.out.println( pers.getDatos() );
```

#### Código cliente 4

```
Empleado uno= new Empleado("Carlos", "lavanderia");
Persona desc = uno;
System.out.println( desc.getEmpresa() );
```

#### Código cliente 5

```
Persona desc = new Persona("Carlos");
Empleado emp = (Empleado)desc;
System.out.println( emp.getEmpresa() );
```

#### Código cliente 6

```
Persona desc = new Persona("Carlos");
if (desc instanceof Empleado) {
    Empleado emp = (Empleado)desc;
    System.out.println( emp.getEmpresa() );
}
```

#### Persona.getDatos() con enlace estático

```
class Persona {
    public Persona(String n) {nombre=n;}
    public final String getDatos() { return (nombre); }
```

```
...  
private String nombre;  
}
```

#### Código cliente 7

```
Empleado emp = new Empleado("Carlos", "lavanderia");  
Estudiante est = new Estudiante("Juan", "Derecho");  
Persona pers;  
pers = emp;  
System.out.println( emp.getDatos() );  
System.out.println( est.getDatos() );  
System.out.println( pers.getDatos() );
```