

Tema 5. Lenguajes de programación para patrones

Las condiciones de los lenguajes que pueden realizar patrones

- Tiene que apoyar el concepto de polimorfismo
- Poder tener funciones virtuales o métodos
- Poder definir clases abstractas



Tipos de lenguajes que pueden realizar patrones con funciones virtuales

- C++: virtual
- Java: Todos los métodos no estáticos por defecto todos son “funciones virtuales”
Solamente los con la clave “final” o privados no son virtuales ni heredables
- Delphi: Los métodos virtuales se han marcado con la clave “virtual”, las clases derivadas con “override”
- C#: la clase base tiene que disponer la clave “virtual” en sus métodos virtuales y las clases derivadas con “override” en su implementación
- VB.NET: la clase base tiene que disponer la clave “Overridable” en sus métodos virtuales y las clases derivadas con “Overrides” en su implementación de estas funciones virtuales



CLASES ABSTRACTAS Y FUNCIONES VIRTUALES PURAS

- Una **función virtual pura** o **método virtual puro** es una función virtual que necesita ser implementada por una clase derivada que no sea abstracta.
- Clases abstractas : las clases que contienen métodos virtuales puros
 - No pueden ser instanciadas directamente
 - Sus subclase sólo puede instanciar directamente si todos los métodos virtuales puros han sido implementados por esa clase o una clase padre.



EN C++

UNA FUNCIÓN VIRTUAL PURA

- Sintaxis:

virtual tipo retorno nombre función (lista de parámetros) = 0

Ejemplo 1.

virtual void dibujar() = 0;

Ejemplo 2:



```
class figura {
float peso;
public:
...
virtual void rotar(int) = 0;           // función virtual pura
virtual float area() = 0;             // función virtual pura
virtual float volumen ( ) = 0;       // función virtual pura
...
};
```

```
#include "figura.h"
const pi = 3.1416;
class esfera : public figura
{ float r;                             // radio
public:
    esfera (float rad, float p);
    void rotar (int d) {};              // cuerpo nulo
    float area () { return 4 * pi * r * r; }
    float volumen ( ) { return area ( ) * r / 3 ; }
};
```



Funciones virtuales puras en C++ son equivalentes a métodos abstractos en JAVA

```
//Una demostración sencilla de abstract.  
abstract class A {  
    abstract void callme();  
    // concrete methods are still allowed in abstract  
    classes  
    void callmetoo() {  
        System.out.println("This is a concrete  
        method.");  
    }  
}  
class B extends A {  
    void callme() {  
        System.out.println("B's implementation of  
        callme.");  
    }  
}
```



IMPLEMENTACIÓN DE LOS PATRONES

- Definir la aplicación
- Encontrar la clase abstracta en el patrón
- Definir la clase abstracta según la necesidad de la aplicación.

