

Junio 2017

Ejercicio 1:

A) Si el programa compila y su salida por pantalla será:

3
16
8

NOTA: Aunque en la última declaración del main se declara un new bar sobre una clase Foo se toman todas las funciones de la clase Foo excepto todas aquellas que pongan override, como es el caso de la función output() de la clase bar, de ahí que el valor sea |8| entre barras.

B) No el programa no compila, debido a que en la penúltima línea declaramos una clase Bar de tipo doublé y luego inicializamos una clase foo de tipo int por lo que es un error. Si en lugar de doublé fuese int si compilaría.

*Ejercicio 2:

A) Define el concepto de programación genérica. Haz que tu definición sea independiente de un lenguaje concreto.

La Programación Genérica es un estilo de programación que permite escribir estructuras de datos y algoritmos utilizando tipos de datos que están todavía sin definir, pero manteniendo la seguridad del control de tipos de un lenguaje.

B) Siendo coherente con tu definición el (a) y dentro del paradigma de orientación a objetos, define los conceptos de clase genérica y método genérico.

- Una clase genérica te permite establecer una clase orientada a una serie de tipos de datos desconocidos, que tú debes de concretar a la hora de utilizar una clase de este tipo.
- Del mismo modo un método genérico es aquel que realiza una serie de operaciones, comparaciones o salidas por pantalla con tipos de datos aún sin definir. En ocasiones debemos de establecer que el dato desconocido que utilizaremos es ordenado o numérico para realizar operaciones o comparaciones en el método.

C) Implementa un lenguaje orientado a objetos a tu elección una clase genérica que tenga (entre otros elementos que consideres necesarios) un método genérico. El ejemplo debe de ser completo, es decir: la clase debe incluir todos los atributos y métodos para poder, potencialmente, compilar.

```
Java
Class Identificador<T> {
    T Id;
    Identificador (T _id) { id=_id }

    T consultar_id(){
        Return this.Id;
    }
}
```

```
C++
Template<typename T>
Class Identificador{
    T Id;
    Identificador (T _id) id(_id) { }
    T consultar_id(){
        Return this->Id;
    }
};
```

D) Desarrolla un programa principal que utilice la clase anterior, instanciando tanto la clase como el método correspondiente.

Java

```
Public class Main{  
    Public static void main (String[] args) {  
        Identificador<int> i= new  
        Identificador<Int>(3245);  
  
        System.out.print(i.consultar_id());  
    }  
}
```

C++

```
Int main(){  
    Identificador<Int>* i=new  
    Identificador<Int>(3245);  
  
    Std::cout<< i->consultar_id() << std::endl;
```

E) Explica sobre tu código en (c) y (d) tanto tu definición de la programación genérica (a) como tus definiciones concretas para clase y método genérico (b)