Septiembre 2017

Ejercicio 1:

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

```
3
|16|
24
```

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 sil() de la clase bar, de ahí que el valor sea 24 y que no este entre ||, debido a que la función de sil() se coge de la clase bar y la función output() de la clase Foo.

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

```
9
lalala
```

NOTA: En este caso como en el anterior el override influye para que en ningún momento se ejecute la función transform() de la clase foo.

*Ejercicio 2:

A) Define función genérica o método genérico

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) De los tres ejemplos cuales compilan y cuales no

Java

```
class Min {
public static <T>
   T min(T t1, T t2)
   {
      if (t1<t2) return t1;
      else       return t2;
   }
}
```

En este caso **no compilaría** debido a que para definir una función genérica en java en la cabecera debemos de declarar el dato genérico que se utiliza en este caso 'T', la cabecera debería ser:

Class Min <T> {

C++

```
template<typename T>
T min(T t1, T t2)
{
   if (t1<t2) return t1;
   else       return t2;
}
```

Si compilara esta clase genérica en c++ debido a que se ha declarado el template y no existe ningún tipo de error en la nomenclatura.

Haskell

```
min :: t -> t -> t
min a b
   | a < b     = a
   | otherwise = b
```

No compila para que una función genérica función en haskell debemos de poner que lavariable genérica en este caso t, se puede comparar, es decir se pueden utilizar comandos como < > == … para ello al declarar la función deberíamos poner:

min :: Ord t => t -> t -> t

de esta forma la función si compilaría.

C) Esta hecho en el mismo B creo preguntar profesor

D)

Haskell

```
min :: (Ord t) => t -> t -> t
min a b
   | a < b = a
   | otherwise = b
```

Java

Preguntar como seria siempre me da error en t1<t2

E) Los cambios que hemos realizado en D como bien he explicado en el apartado C permiten que el código en haskell refleje que el dato genérico T se vea como un dato que se puede comparar.