

Tecnología de la programación

Sesión 13

Objetivos de la sesión

1. Repasar la introducción de punteros, especialmente las transparencias 17 y 18.
2. Estudiar hasta la transparencia 40 (incluida).

Guion

Transparencia 20:

- Es un ejercicio para decir qué pasa. Lo que sucede es que NO es correcto, porque estamos asignando a `dest(q)` (en C++ con `*q`) el valor 3. Sin embargo, `q` NO apuntaba a nada.

Transparencias 23, 24 y 25:

- Se recomienda dibujar lo que va pasando en cada línea de código.
- La forma de hacer la posibilidad 4 es lo que viene a continuación en todo el tema, con el operador `reservar`.

Transparencia 36:

- El código siguiente NO es correcto (se recomienda ir dibujando línea a línea lo que va pasando):

```
puntero a entero p, q
p = reservar(entero)
q = reservar(entero)
dest(p) = 5
dest(q) = 6
p = q
liberar(p)
liberar(q)
```

Está mal porque al hacer `p=q` has perdido la referencia a lo que apuntaba `p` (la parte de memoria donde estaba el 5). Es decir, esa memoria se ha quedado perdida.

Transparencia 37:

- El siguiente código es correcto. Se reserva usando un puntero y se libera usando otro. Sin la instrucción **q = p** no sería correcto, ya que se pierde la referencia a una zona reservada.

```
int x;  
int *p,*q;  
p = new(int);  
*p = 5;  
q = p;  
p = &x;  
delete(q);
```