

6 - Riferimenti vs Puntatori

Un puntatore è un oggetto il cui valore (un indirizzo) si può riferire ad un altro oggetto, mentre un riferimento non è un oggetto vero e proprio, ma è una sorta di "nome alternativo" che consente di accedere ad un oggetto esistente.

Inizializzazione:

- quando viene creato un riferimento, questo deve essere inizializzato, perché si deve sempre riferire ad un oggetto esistente, non esiste il concetto di riferimento nullo;
- un puntatore può non essere inizializzato (non consigliato), oppure essere inizializzato con `nullptr`, in questo caso non punterà ad alcun oggetto.

Modificabilità:

- una volta creato un riferimento, questo si riferirà sempre allo stesso oggetto; non c'è modo di "riassegnare" un riferimento ad un oggetto differente;
- un puntatore (non qualificato `const`) può essere modificato per puntare ad oggetti diversi o a nessun oggetto.

Accesso a oggetto riferito/puntato:

- ogni volta che si effettua una operazione su un riferimento, si sta implicitamente operando sull'oggetto riferito;
- nel caso dei puntatori abbiamo a che fare con due oggetti diversi: l'oggetto puntatore e l'oggetto puntato:
 - operazioni in lettura e scrittura applicate direttamente al puntatore accedono e, potenzialmente, modificano l'oggetto puntatore;
 - per accedere all'oggetto puntato occorre utilizzare la dereferenziazione.

Qualificazione `const`:

- nel caso dei riferimenti si applica all'oggetto riferito;
- nel caso dei puntatori è possibile specificare il qualificatore `const` per ognuno dei due oggetti:
 - nella dichiarazione di un puntatore è possibile scrivere `const` due volte:
 - `const` a sinistra di `*`, si applica all'oggetto puntato;
 - `const` a destra di `*`, si applica all'oggetto puntatore.

Lifetime:

Dopo che il puntatore termina il suo ciclo di vita, solo il puntatore viene distrutto, non l'oggetto puntato. Analogamente per i riferimenti.

Nel caso in cui un riferimento viene inizializzato con un temporaneo, l'oggetto temporaneo finisce il ciclo di vita insieme al riferimento.

Riferimenti "dangling"

Come è possibile creare un dangling pointer (puntatore non nullo che contiene l'indirizzo di un oggetto non più esistente), è anche possibile creare una *dangling reference*.

I casi sopra considerati sono esclusivamente i riferimenti a lvalue (`T&`), la maggior parte delle osservazioni sono valide anche nel caso di riferimenti a rvalue (`T&&`).