

## Programmazione ad Oggetti - parte B: Esercitazione di laboratorio 7

Esercizio 1: Scrivere una classe Item da utilizzare come elemento di un `BinaryTree` che contenga un campo "nome" (stringa) e l'operatore "<". Creare un `BinaryTree` nel programma principale inserendo alcuni elementi e verificare il corretto attraversamento dell'albero secondo una delle modalità viste a lezione.

Esercizio 2: Scrivere una funzione membro "equal\_tree\_structure" della classe `BinaryTree` per determinare se un albero ha la stessa struttura di un albero il cui puntatore al nodo radice viene passato come argomento (stesso numero nodi, stesso numero di livelli, stesso numero di figli in ogni sotto-albero – ma non necessariamente gli stessi valori nei nodi). E' consentito rendere "root" membro pubblico della classe `BinaryTree` per avere un accesso diretto.

```
int equal_tree_structure(Node<T>* theRootB)
```

Esercizio 3: Scrivere una funzione membro "equal\_tree" della classe `BinaryTree` per determinare se un albero è uguale ad un albero il cui puntatore viene passato come argomento (stesso numero nodi, stesso numero di livelli, stesso numero di figli in ogni sotto-albero, stessi valori nei nodi).

```
int equal_tree(BinaryTree<T>* treeB)
```

Esercizio 4: Scrivere il distruttore della classe `BinaryTree`