

TREE <i>parent</i>	// Riferimento al padre
TREE <i>child</i>	// Riferimento al primo figlio
TREE <i>sibling</i>	// Riferimento al prossimo fratello
ITEM <i>value</i>	// Valore memorizzato nel nodo

```

TREE Tree(ITEM v)
    TREE t = new TREE
    t.value ← v
    t.parent ← t.child ← t.sibling ← nil
    return t

```

```

insertChild(TREE t)
    t.parent ← self
    // inserisci t prima dell'attuale primo figlio
    t.sibling ← child
    child ← t

```

```

insertSibling(TREE t)
    t.parent ← parent
    // inserisci t prima dell'attuale prossimo
    // fratello
    t.sibling ← sibling
    sibling ← t

```

```

deleteChild
    TREE newChild ← child.rightSibling
    delete(child)
    child ← newChild

```

```

deleteSibling
    TREE newBrother ← sibling.rightSibling
    delete(sibling)
    sibling ← newBrother

```

// metodo ausiliare

```

delete(TREE t)
    TREE u ← t.leftmostChild
    while u ≠ nil do
        TREE next ← u.rightSibling
        delete(u)
        u ← next

```