

## TP3- Implémentation

Le but de ce Tp est d'implémenter une liste chaînée représentant un ensemble sans verrou (avec des CAS). Pour simplifier on supposera que les éléments sont des entiers (Integer). L'interface Ensemble est alors la suivante:

```
public interface Set{
    boolean add( Integer x);
    boolean remove (Integer x);
    boolean contains( Integer x);
}
```

**Exercice 1.**— Dans cette implémentation on utilise la classe Node ci dessous. Pour simplifier on supposera que la clef (key) est l'entier (int) correspondant à l'Integer item.

```
public class Node {
    public Integer item;
    public int key;
    public AtomicMarkableReference<Node> next;
}
```

1. Réaliser une implémentation atomique d'une liste pour cette implémentation. Préciser les points de linéarisation.
2. Quelle propriété de progression ont add? remove? contains?
3. Réutiliser le programme du TP précédent qui lancent 3 threads et les threads ajoutent et enlèvent des éléments avec cette implémentation.

**Exercice 2.**— Tester et comparer les 3 implémentations (gros grain, grain fin et CAS) en supposant différents pourcentages entre les appels de contains, add et remove.