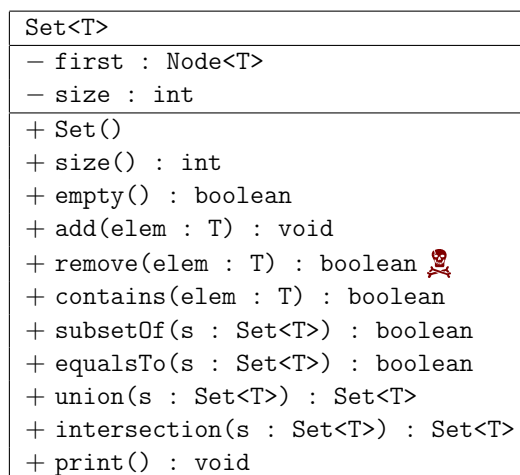


## Esercitazione 6: code e insiemi dinamici generici

**Esercizio 1.** Modificare l'implementazione della coda dinamica di interi (il cui codice è disponibile su Moodle), in modo da renderla generica, cioè in grado di ospitare oggetti di un tipo *T* non noto a priori. Un'applicazione che usa l'implementazione della coda dinamica generica dovrà poter creare code di oggetti di tipi specifici (es. code di *Integer*, di *Double*, ecc.). Nell'implementazione della coda dinamica generica, si assuma che, per il tipo generico *T* degli oggetti ospitati nella coda, sia disponibile il metodo `equals` che ne implementa il criterio di uguaglianza (nel senso che per ogni *x* e *y* di tipo *T*, `x.equals(y)` restituisce `true` se e solo se *x* è uguale a *y*).<sup>1</sup>

Scrivere, inoltre, una classe di test *TestDynamicQueue* sulla base di quella prodotta per la coda dinamica di interi (il cui codice è disponibile su Moodle) che testi il funzionamento della coda dinamica generica, istanziandola sui seguenti tipi: *Integer*, *Double*, *String* e *Person* (la classe *Person* è data e il suo codice è disponibile su Moodle). Tutti i metodi devono essere testati in tutte le istanziazioni.

**Esercizio 2.** Realizzare una classe *Set<T>* per rappresentare insiemi dinamici di oggetti di un tipo generico non noto a priori, con una lista dinamica. Evincere significato e comportamento di campi e metodi della classe *Set<T>* dal seguente diagramma UML:



Si ricorda che un insieme è una collezione in cui l'ordine degli elementi *non è importante* e tale che ogni elemento può comparire *al massimo una volta*.

Nell'implementazione dell'insieme dinamico generico, si assuma che, per il tipo generico *T* degli oggetti dell'insieme, sia disponibile il metodo `equals` che ne implementa il criterio di uguaglianza (nel senso che per ogni *x* e *y* di tipo *T*, `x.equals(y)` restituisce `true` se e solo se *x* è uguale a *y*).

Scrivere, inoltre, una classe di test *TestSet* che testi il funzionamento dell'insieme dinamico generico, istanziandolo sui seguenti tipi: *Integer*, *Double*, *String* e *Person* (la classe *Person* è data e il suo codice è disponibile su Moodle). Tutti i metodi devono essere testati in tutte le istanziazioni.

---

<sup>1</sup>Java fornisce un'implementazione di default per il metodo `equals`, la quale, se necessario, può essere opportunamente ridefinita (sovrascritta) in ogni classe. Il meccanismo di ridefinizione è il cosiddetto *overriding dei metodi* che vedrete in seguito. Qui possiamo accontentarci di assumere che per il tipo *T* sia stata in qualche modo specificata l'opportuna implementazione per il metodo `equals`, il che consente l'uso di tale metodo, quando necessario.