

**Metodi Avanzati di Programmazione**  
**Corso di Laurea in Informatica (Bari)**  
**AA 2011-2012**  
**Prova Scritta del 9 luglio 2012**

1) Descrivere le forme di astrazione in programmazione a voi note per la realizzazione della astrazione funzionale. Esempificare la risposta.

(4 punti)

2) Dare le specifiche algebriche (semantiche e di restrizione) per il tipo astratto *TabellaTris* (intesa come una griglia di 3x3 celle i cui segnare "x" o "o") di cui si forniscono le seguenti specifiche sintattiche:

**Tipi:** TabellaTris, Segno Booleano, Intero;

**Operatori:**

creaTabella() → TabellaTris // crea una tavola 3x3 vuota.

segna (TabellaTris, Intero, Intero, Segno) → TabellaTris // assegna il segno specificato alla cella vuota identificata dagli interi passati come indice di riga e di colonna

contaSegni(TabellaTris) → Intero // conta i segni nella tavola

fineGioco(TabellaTris) → Booleano // restituisce vero se tutte le celle della tavola ospitano un segno

leggiSegno(TabellaTris, Intero, Intero) → Segno // restituisce il segno nella cella identificata da dagli interi passati come indice di riga e di colonna , restituisce errore se la cella è vuota

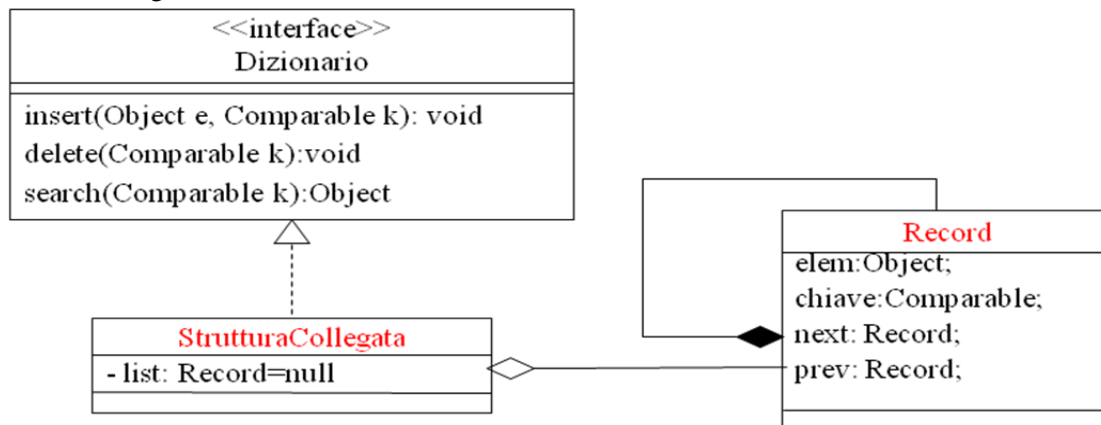
cancellaSegno(TabellaTris, Intero, Intero) → TabellaTris // cancella il segno nella cella identificata da dagli interi passati come indice di riga e di colonna

vittoriaInRiga(TabellaTris, Intero, Intero) → Booleano // restituisce vero se le celle allineate nella riga che ospita la cella identificata dagli interi passati come indice di riga e colonna contengono lo stesso segno

uguale(TabellaTris, TabellaTris) → Booleano // restituisce vero se le due tabelle hanno gli stessi segni nelle stesse posizioni

(7 punti)

3) Descrivere e commentare i principali concetti del paradigma OO che si evincono dal diagramma UML che segue.



(5 punti)

4) Con riferimento al problema di ricercare il valore massimo contenuto in un array di oggetti, considerare i prototipi:

*Object massimo(Object a[])*

*Object massimo(Object a[], Comparator c)*

Fornire e **commentare** una implementazione JAVA per entrambi i prototipi. Definire in JAVA la classe *Studiante* che includa gli attributi *matricola*, *cognome*, *nome* e mostrare come effettuare la ricerca del massimo studente in un array di *Studiante* usando i metodi precedentemente definiti. Il massimo va ricercato una volta rispetto alla *matricola* e una volta rispetto al *cognome*. Commentare le decisioni prese.

(6 punti)

5) Con riferimento alla progettazione UML del dato astratto *StrutturaCollegato*, descrivere i passi da eseguire per rendere possibile in JAVA la serializzazione su file di un oggetto istanza della classe *StrutturaCollegata*. Estendere la definizione della classe *StrutturaCollegata* con due metodi preposti al salvataggio e caricamento di oggetto serializzato in /da file. Fornire e **commentare** l'implementazione JAVA di tali metodi in *StrutturaCollegata*.

(6 punti)

6) Descrivere e commentare il ciclo di vita di un thread. Commentare l'uso di *synchronized* in JAVA esemplificando la risposta .

(5 punti)