Metodi Avanzati di Programmazione Corso di Laurea in Informatica Anno Accademico 2015/2016

Prova scritta del 02/09/2016 ore 9:00-12:00

1) Fornire le specifiche algebriche (semantiche e di restrizione) in forma di equazioni per il tipo astratto **AlberoBinario** di cui si forniscono le seguenti specifiche sintattiche:

Tipi: AlberoBinario, Nodo, ListaNodi, Booleano

Operatori:

creaAlbero ()→Albero

insRadice (Albero, Nodo)→ Albero

insFiglioSinistro(Albero, Nodo, Nodo) → Albero // inserisce un nodo (terzo parametro) qual figlio sinistro di un altro nodo (secondo parametro) dell'albero (primo parametro).

insFiglioDestro(Albero, Nodo, Nodo) → Albero // inserisce un nodo (terzo parametro) qual figlio destro di un altro nodo (secondo parametro) dell'albero (primo parametro).

foglia(Albero,Nodo) → Booleano // restituisce vero se il nodo è foglia, falso altrimenti

cancellaNodo(Albero, Nodo) → Albero // cancella sotto-albero radicato nel nodo trasformandolo in foglia.

discendenti(Albero, Nodo)→ListaNodi //Restituisce la lista di tutti i nodi discendenti (diretti e indiretti) del nodo passato come parametro

sottoAlbero(Albero, Nodo)→Albero // restituisce il sotto-albero radicato nel nodo passato come parametro contenuto(Albero,Albero) → Booleano //verifica se il primo albero contiene tutti i nodi contenuti nel secondo albero (7 punti)

2) Descrivere la astrazione generica e le sue applicazioni in ADA. Scrivere la realizzazione del tipo astratto Coppia in ADA, considerando la seguente specifica sintattica di Coppia:

creaCoppia(Elemento, Elemento) → Coppia // crea un coppia di due elementi ; **Elemento può essere un tipo qualunque**

sinistra(Coppia)→Elemento // restituisce l'elemento a sinistra nella coppia

destra(Coppia)→Elemento // restituisce l'elemento a destra nella coppia

Mostrare l'uso di tale tipo astratto nella manipolazione di coppie di reali e coppie di caratteri (7 punti)

- 3) Spiegare le relazioni di **aggregazione** e **composizione** nel paradigma OO, spiegare quando queste vengono usate per simulare ALCUNE forme di ereditarietà. Fornire esempi in UML. (6 punti)
- 4) Descrivere **dettagliatamente** la gestione delle Eccezioni in Java. Fornire esempi. (5 punti)
- 5) Identificare i contenitori più appropriati per la modellazione di **LibrettoEsami** (che modella un insieme di Esami) e **ArchivioStudenti** (che archivia il LibrettoEsami associato a ciascuno Studente). **Motivare i contenitori scelti.**

Scrivere in JAVA le classi **Studente** (con proprietà matricola(unica e distinta per ogni studente), cognome, nome), **Esame** (con proprietà codiceEsame (unica e distinta per ogni esame), nomeEsame), **LibrettoEsami** e **ArchivioStudenti**. Per la classe **LibrettoEsami** considerare la specifica sintattica:

creaLibretto() --> LibrettoEsami,

verbalizza(LibrettoEsami, Esame)--> LibrettoEsami, // inserisce un nuovo Esame nel Libretto esameSostenuto(LibrettoEsami, Esame)-->Booleano // restituisce vero se l'esame è stato sostenuto; falso

*altrimenti*Per la classe **ArchivioStudenti** considerare la specifica sintattica:

creaArchivio()→ArchivioStudenti

inserisci(Studente.LibrettoEsami)→ArchivioStudenti

ricerca(ArchivioStudenti,Studente)→LibrettoEsami // restituisce il libretto Esami archiviato per lo studente specificato, errore se non esiste alcun libretto esami in corrispondenza dello studente

Commentare il codice scritto.

(8 punti)