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

Prova scritta del 16/11/2016 ore 9:00-12:00

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

Tipi:

GrafoPesatoOrientato, Nodo, Reale, Lista

Operatori:

creaGrafoOrientatoPesato ()→GrafoOrientatoPesato

insNodo (GrafoOrientatoPesato, Nodo)→ GrafoOrientatoPesato

insArco(GrafoOrientatoPesato, Nodo, Nodo, Reale)→ GrafoOrientatoPesato // inserisce un arco dal primo nodo (secondo parametro) al secondo nodo (TerzoPrametro) e assegna un peso all'arco (quarto parametro)

pesoOut(GrafoOrientatoPesato,Nodo) → Reale // somma di pesi associati ad archi in uscita al nodo specificato

pesoIn(GrafoOrientatoPesato,Nodo) → Reale // somma di pesi associati ad archi in entrata al nodo specificato

cancellaNodo1(GrafoOrientatoPesato, Nodo) →GrafoOrientatoPesato // cancella il nodo solo se non ci sono archi che lo coinvolgono

cancellaNodo2(GrafoOrientatoPesato, Nodo)→GrafoOrientatoPesato // cancella un nodo e tutti gli archi che lo coinvolgono

connessioni(GrafoOrientatoPesato, Nodo, Reale)→Lista *// restituisce la lista dei nodi a cui arriva un arco in partenza dal nodo specificato (secondo parametro) con peso superiore a quello indicato (terzo parametro)*

unione(GrafoOrientatoPesato,GrafoOrientatoPesato, Reale) → GrafoOrientatoPesato *//restituisce un grafo che contiene tutti i nodi dei due grafi (primo e secondo paragrafo) per i quali la somma dei pesi in uscita sia maggiore del reale specificato come terzo paragrafo, senza costruire gli archi.*

(7 punti)

2) Descrivere in maniera esaustiva la astrazioni di funzione, la astrazione di procedura e la astrazione funzionale. Spiegare il rapporto tra loro esistente

(5 punti)

3) Fornire le definizioni di classi astratte, classi finali, metaclassi e interfacce nel paradigma OO. Fornire esempi delle stesse in UML.

(5 punti)

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

Object quartile(Object a[], Float)
Object quartile(Object a[], Comparator c, Float)
Il parametro Float è un valore compreso tra 0 e 100 e denota una percentuale
Il quartile restituisce l'oggetto al di sotto del quale ricade una certa percentuale di oggetti nello array. Per esempio il quartile al 25% è l'oggetto in modo che dopo aver ordinato lo array il 25% risultano alla sua sinistra e i rimanenti alla sua destra.

Fornire e **commentare** una implementazione JAVA per entrambi i prototipi. Definire in JAVA la classe **Studente** che includa gli attributi matricola di tipo intero e cognome di tipo Stringa mostrare come effettuare la ricerca del punto quartile in un array di Studente usando i metodi precedentemente definiti. Il quartile va cercato una volta rispetto a matricola e una volta rispetto a nome. **Commentare le decisioni prese e il codice scritto.** **(5 punti)**

5) Descrivere **dettagliatamente** la gestione delle eccezioni in Java. Fornire esempi dei costrutti descritti

(6 punti)

6) Descrivere l'uso di **syncronized** in Java e fornire esempi del suo uso. **(5 punti)**