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

Prova scritta del 22/02/2017 ore 9:00-12:00

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

Tipi:

Cassa, Cliente, Intero, Booleano

Operatori:

creaCassa()→Cassa //istanzia una cassa inizialmente vuota

aggiungiCliente(Cassa, Cliente) →Cassa // aggiunge un cliente in coda alla cassa

serviCliente(Cassa)→ Cassa //rimuove il cliente nella testa della cassa

leggiCliente (Cassa) → Cliente // restituisce il cliente nella testa della cassa

contaClienti(Cassa) →Intero //restituisce il numero dei clienti in coda alla cassa

leggiCliente(Cassa, Intero)→ Cliente //restituisce il cliente nella posizione specificata, errore se la posizione non è valida

restituisciClienteAnziano(Cassa)→Cliente // restituisce il cliente più anziano in coda alla cassa

incoda(Cassa, Cassa, Cliente)→Cassa //aggiunge il cliente alla cassa meno affollata e restituisce la cassa modificata

(7 punti)

2) Fornire la definizione di astrazione generica e descrivere il ruolo della astrazione generica nella definizione di tipi astratti e classi nel linguaggio ADA

(5 punti)

3) Descrivere le relazioni di aggregazione e composizione nel paradigma OO, mostrando gli ambiti di applicazione. Formulare esempi delle stesse usando UML.

(5 punti)

4) Scrivere un esempio di programma Java, che usi le Java Generics, per gestire una lista di "Studente" (modellato da cognome, nome, matricola). La classe deve erogare i seguenti servizi: creare la lista di studenti, permettere di aggiungere uno studente alla lista, ricercare il numero di occorrenze di studenti uguali a quello passato. Due studenti sono uguali se condividono la stessa matricola. Commentare vantaggi dell'uso delle Java Generics.

(5 punti)

5) Descrivere **dettagliatamente** le forme di RTTI tradizionale in Java. Fornire esempi dei costrutti descritti

(5 punti)

6) Descrivere il **ciclo di vita di un Thread**. Descrivere la classe **Thread** e l'interfaccia **Runnable**. Scrivere **un programma che** avvia 10 thread che stampano a video la sequenza definita come segue.

X X+1 X+2 X+3 ...

dove X è un parametro reale. Nel programma mostrare la definizione dei thread a mezzo della interfaccia Runnable. **Commentare il codice scritto.**

(6 punti)