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

**Prova scritta del 26/09/2016 ore 14:30-17:30**

1) Descrivere la classificazione dei paradigmi di programmazione. (2 punti)
2) Fornire le equazioni delle specifiche algebriche (semantiche e di restrizione) per il tipo astratto ReteScientifica di cui si forniscono le seguenti specifiche sintattiche:

**Tipi:**

ReteScientifica, Autore, Intero, Booleano

**Operatori:**

creaReteScientifica()→ReteScientifica //crea un rete scientifica vuota

registraAutore(ReteScientifica,Autore)→ReteScientifica // aggiunge un nuovo autore a una Rete Scientifica

aggiungiCollaborazione(ReteScientifica, Autore, Autore, Intero)→ ReteScientifica // stabilisce una collaborazione tra due autori con peso pari all'intero specificato

intensificaCollaborazione(ReteScientifica, Autore, Autore)→ ReteScientifica // se esiste una collaborazione tra i due autori ne incrementa il peso di una unità, altrimenti errore

pesoCollaboratori(ReteScientifica, Autore)→ Intero //restituisce la somma dei pesi nelle collaborazioni che coinvolgono l'autore specificato

cancellaAutore(ReteSceintifica,Autore)→ReteScientifica // cancella l'autore e tutte le collaborazioni che lo coinvolgono

maggiore(ReteScientifica, ReteScientifica, intero) → Booleano // restituisce vero se il numero di collaborazioni con peso maggiore dell'intero specificato nella prima rete è maggiore stretto del numero di collaborazioni con peso minore dell'intero specificato nella seconda rete. (7 punti)

3) Illustrare in maniera esaustiva la teoria relativa al concetto di metaclasse nel paradigma OO. Spiegare se, e/o in che forma, esistono le metaclassi in Java. (5 punti)

4) Spiegare il principio di sostituibilità nel paradigma OO. Descrivere le diverse relazioni di ereditarietà nel paradigma OO esemplificando la risposta usando la notazione UML. Indicare il legame tra principio di sostituibilità e relazione di ereditarietà. (5 punti)

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

Object minimo(Object a[])
Object minimo(Object a[], Comparator c)

Fornire e commentare una implementazione JAVA per entrambi i prototipi. Definire in JAVA la classe Esame che includa gli attributi codice, nome, crediti, voto e mostrare come effettuare la ricerca del minimo esame in un array di Esame usando i metodi precedentemente definiti. Il minimo va ricercato una volta rispetto al numero di crediti e una volta rispetto al voto. Commentare le decisioni prese. (7 punti)

6) )Spiegare che cosa si intende per serializzazione in Java. Descrivere le condizioni che devono essere soddisfatte affinché un oggetto possa essere serializzato. Mostrare un esempio di codice Java per salvare e caricare un oggetto istanza della classe Esame definita nel seguito (salvare tutte le informazioni):

```
class Esame{
          transient String aula;
          static Integer codiceGenerator=1;
          private int codiceEsame=codiceGenerator++;
          private String nome;
          private int voto;
          Esame(String aula, String nome, int voto)[
                    this.aula=aula; this.nome=nome;this.voto=voto;
          }
}
```

(7 punti)