

Tempo a disposizione: **2 ore**

Risultati: **12 Luglio 2012, dopo le 13, su CampusNet**

Discussione dell'elaborato: **13 Luglio 2012, ore 9**

```
package esame;

import java.util.Iterator;

public interface Set {
    public boolean add(Object o);
    public boolean contains(Object o);
    public boolean isEmpty();
    public Iterator iterator();
}
```

Esercizio 1. Data l'interfaccia `Set` riportata in figura che descrive l'astrazione di *insieme di oggetti*, realizzare nello stesso package:

1. `Array100Set`, che implementa l'interfaccia `Set` utilizzando un array di 100 oggetti.
2. Eventuali altre classi necessarie per realizzare la classe precedente.

Nota: il metodo `add(Object)` aggiunge un oggetto all'insieme solo se l'oggetto non è già presente e utilizza il valore di ritorno per indicare se l'oggetto è stato effettivamente aggiunto.

Disegnare un class diagram UML che riporti le classi realizzate e le relazioni tra le stesse.

Esercizio 2. Si supponga di realizzare un sistema di gestione del seguente dominio applicativo:

La banca dati di una Università raccoglie le informazioni su tutti gli studenti che sono o sono stati iscritti. Ogni studente è caratterizzato da un identificativo numerico univoco (matricola), da un nome e da un cognome. Per ogni studente viene mantenuto lo storico degli esami sostenuti ed ogni esame è relativo ad un insegnamento ed è caratterizzato da un voto ed una data in cui l'esame è stato sostenuto. Infine, gli insegnamenti sono caratterizzati da un nome e da un numero di crediti formativi (CFU). I docenti dell'università possono accedere allo storico degli esami sostenuti dagli studenti, mentre gli studenti possono accedere unicamente alle informazioni che li riguardano.

Disegnare il **modello del dominio** e realizzare un **budget a tempo di consegna minimo** indicando anche opportuni diagrammi di Gantt e PERT e riportando il costo ed il tempo di consegna.

Per la redazione del budget si tenga presente che:

1. viene adottato un processo di sviluppo a cascata
2. il sistema è composto da un server e da un client le cui fasi di analisi e di codifica vengono svolte separatamente, mentre la fase di progettazione è unica
3. il team a disposizione comprende: 1 analista, 1 progettista senior (che può essere impiegato anche come analista), 1 progettista junior (che può essere impiegato anche come programmatore), 1 programmatore
4. i tempi sono stimati utilizzando i seguenti criteri basati sul modello del dominio
 - a. l'analisi del server richiede 1PD (PD = Person Day) per ogni classe
 - b. l'analisi del client richiede 1PD per ogni classe e per ogni associazione
 - c. la progettazione del server richiede 1PD per ogni classe
 - d. la progettazione del client richiede 1PD per ogni classe e per ogni associazione
 - e. la codifica del server richiede 2PD per ogni classe
 - f. la codifica del client richiede 5PD per ogni classe e per ogni associazione
5. gli stipendi netti dei collaboratori sono i seguenti: analista €2500, progettista senior €2000, progettista junior €1500, programmatore €1000