

Metodi Avanzati di Programmazione

Corso di Laurea in Informatica (Bari)

AA 2011-2012

Prova Scritta del 20 giugno 2012

1) Fornire una definizione di classe in ADA e mostrarne l'uso nella definizione del tipo astratto COPPIA di elementi di tipo TipoElemento1 e TipoElemento2.

```
crea(TipoElem1, TipoElem2) --> Coppia
```

```
leggi1(Coppia) --> TipoElem1
```

```
leggi2(Coppia) --> TipoElem2
```

Mostrare l'uso di tale classe nella manipolazione di un numero razionale (numeratore e denominatore di tipo intero) e un numero complesso (parte reale e parte immaginaria di tipo real). Commentare il codice scritto in ADA. (5 punti)

2) Dare le specifiche algebriche (semantiche e di restrizione) per il tipo astratto *MappaNavale* (intesa come una griglia di celle che possono ospitare o meno una nave) di cui si forniscono le seguenti specifiche sintattiche:

Tipi: MappaNavale, Riga, Colonna, Nave Booleano, Intero;

Operatori:

```
creaMappa(Intero, Intero) → MappaNavale // crea una mappa vuota specificando numero di righe e numero di colonne.
```

```
vuoto (MappaNavale) → Booleano
```

```
contaNavi(MappaNavale) → Intero // conta il numero di navi nella mappa
```

```
assegnaNave(MappaNavale, Intero, Intero, Nave) → MappaNavale // posiziona la nave nella mappa in base a indice di riga e indice di colonna passati come parametri
```

```
su(MappaNavale, Nave) → MappaNavale // sposta la nave nella riga sopra rispetto a quella corrente; genera errore se lo spostamento non è possibile
```

```
uguale(MappaNavale, MappaNavale) → Booleano // restituisce vero se le due mappe hanno uguali dimensioni e navi piazzate nelle medesime posizioni. (7 punti)
```

3) Descrivere il meccanismo di aggregazione/composizione nel paradigma OO. Esemplicare in UML. (5 punti)

4) Descrivere la classe TreeSet in Java. Definire la classe Nave che contiene i campi matricola, nome, proprietario. Mostrare un esempio di codice che permetta di creare sia un TreeSet di Navi ordinati in base a matricola sia un TreeSet di Navi ordinato in base al nome. Commentare il codice scritto.

(5 punti)

5) Spiegare il significato delle parole chiave *try*, *catch*, *finally*, *throw* e *throws* nella gestione delle eccezioni in Java. Spiegare la differenza tra Eccezioni Controllare e Non. Esemplicare la risposta.

(5 punti)

6) Descrivere il meccanismo di RMI in Java. Usare lo stesso per scrivere

a. un server che metta a disposizione i metodi *contaCaratteri(String) → Intero* e *maiuscolo(String) → String*

b. un client che invochi i metodi remoto *contaCaratteri* e *maiuscolo* per la stringa "prova di Map del 20 giugno 2012" e stampi a video i risultati .

(6 punti)