

**Metodi Avanzati di Programmazione**  
**Corso di Laurea in Informatica**  
**Anno Accademico 2016/2017**

**II Prova in Itinere del 07/06/2017 ore 15:00-18:00, Aula Magna**

1) Definire il ciclo di vita di un Thread. Spiegare i due meccanismi che Java mette a disposizione per scrivere una applicazione multi-threading. Mostrare l'uso di entrambi i meccanismi per scrivere un programma che crea 10 thread. Ogni thread stampa a video i numeri della successione infinita:

$X^Y, X^{2Y}, X^{3Y}, X^{4Y}, \dots$

Per il primo thread  $X=1$  e  $Y=-1$ , per il secondo  $X=2$  e  $Y=-2$ , per il terzo  $X=3$ ,  $Y=-3$ , ..., per il decimo thread  $X=10$  e  $Y=-10$ .

Fare in modo che i thread costruiti siano daemon e spiegarne il significato.

**Commentare il codice scritto.**

(12 punti)

2) Identificare il contenitore più appropriato per la modellazione del Treno. Scrivere le classi Posto (con proprietà codicePosto, categoria) e Prenotazione (con proprietà codice fiscale, nome, cognome, prezzo) e la classe Treno.

Per la classe Treno considerare la specifica sintattica

**creaTreno() --> Treno,**

**prenota(Treno, Posto, Prenotazione)-->Treno** , // i posti sono unici nel treno rispetto

a CodicePosto, la prenotazione in un posto occupato genera errore

**leggiPrenotazione(Treno, Posto)-->Prenotazione** ,

**uguale(Treno, Treno)--> Booleano**

Descrivere il meccanismo di serializzazione nel dettaglio. Scrivere il codice necessario per serializzare e deserializzare una istanza di Treno in Java. **Commentare il codice scritto.**

(15 punti)

3) Descrivere in maniera esaustiva le forme di RTTI tradizionale in Java fornendo appropriati esempi. **Commentare il codice scritto.**

(6 punti)