Esame di Ingegneria del software

Appello del 24 settembre 2019

Nome e cognome:
Matricola:

Il punteggio relativo a ciascuna domanda, indicato fra parentesi, è in trentesimi. I candidati devono consegnare entro un'ora dall'inizio della prova.

1 **La modularità del sistema influisce sulla gestione del processo di sviluppo?** (1)

- sí, perché richiede l'uso di strumenti CASE ☐
- sí, perché facilita la ripartizione del lavoro ☐
- no, perché è solo un aspetto tecnico del progetto ☐

2 **Nel modello OO, un metodo è** (1)

- un'operazione privata. ☐
- un'operazione pubblica. ☐
- l'implementazione di un'operazione. ☐

3 **Un guasto è** (1)

- un comportamento scorretto rispetto alle specifiche. ☐
- un difetto del codice sorgente. ☐
- un errore di progetto o di programmazione. ☐

4 **I linguaggi formali** (1)

- hanno una sintassi grafica. ☐
- sono standardizzati. ☐
- hanno una semantica di tipo matematico. ☐

5 **Negli Automi a Stati Finiti le uscite** (1)

- dipendono dalla marcatura ☐
- dipendono dallo stato e dall'ingresso ☐
- dipendono dalle condizioni di guardia ☐

6 **Con riferimento alla Fig. 1**, (5)

	V	**F**
il **Produttore** inizia l'interazione	☐	☐
il **Produttore** fa terminare l'interazione	☐	☐
il **Produttore** invia il segnale consuma	☐	☐
il **Consumatore** entra in **Elaborazione** prima del **Produttore**	☐	☐
il **Consumatore** entra in **Attesa** quando riceve produci	☐	☐

7 **Con riferimento alla Fig. 2**, (5)

	V	**F**
un **ConcreteScrollWdw** contiene un **Window**	☐	☐
un **ConcreteScrollWdw** è un **PlainWdw**	☐	☐
ConcreteScrollWdw implementa **PlainWdw**	☐	☐
ConcreteScrollWdw estende il comportamento di **PlainWdw**	☐	☐
un **Window** contiene un **PlainWdw**	☐	☐

8 Con riferimento alla Fig. 3, (5)

	V	F
Un **Network** è composto da istanze di **Node**	☐	☐
Un **Network** è composto da istanze di **Vector**	☐	☐
createIterator() è implementato da **Node**	☐	☐
Database deriva da **Vector**	☐	☐
Si può accedere ai nodi di un **Network** senza conoscerne l'implementazione	☐	☐

9 Disegnare un diagramma di classi che rappresenti la seguente applicazione: (5)
un'agenda elettronica permette di (i) inserire coppie di stringhe (*nome*, *numero*) in un elenco, (ii) cancellare coppie fornendo il nome, e (iii) cercare i numeri corrispondenti ai nomi. L'elenco viene memorizzato in un file. Strutturare l'applicazione separando le funzioni della gestione dell'elenco e della gestione del file contenente l'elenco. L'elenco, oltre alle operazioni di inserimento, cancellazione e ricerca, ha un'operazione `enumera()` per accedere in sequenza alle coppie, e un'operazione `azzera()` per riposizionare tale sequenza sulla prima coppia. Queste due operazioni vengono usate dal gestore del file. Il programma principale (non rappresentato) accede all'elenco ed al file attraverso un'unica classe.

10 Riprogettare la soluzione dell'esercizio precedente (5)
prevedendo che l'elenco possa venire implementato con una lista oppure con una tabella, e che si applichi il pattern *Iterator* (Fig. 4) al posto delle operazioni `enumera()` e `azzera()`.

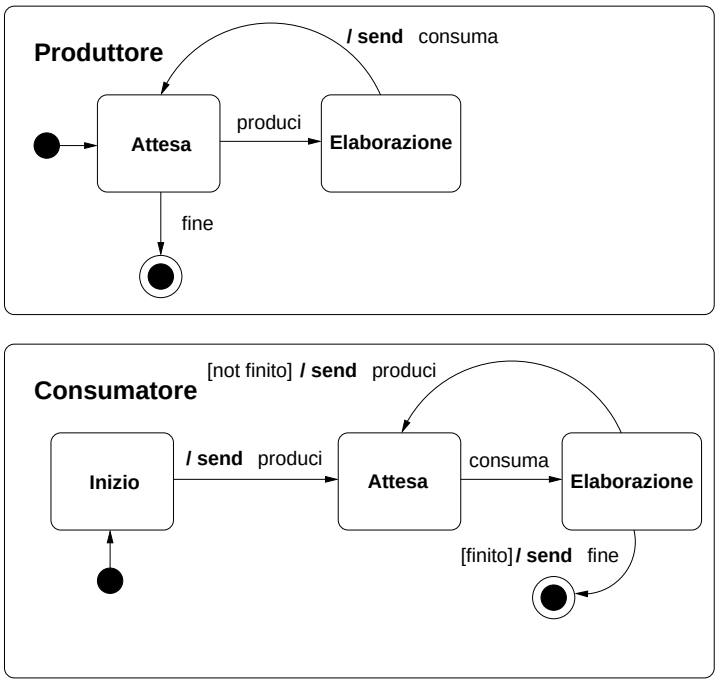

Figura 1: Domanda 6.

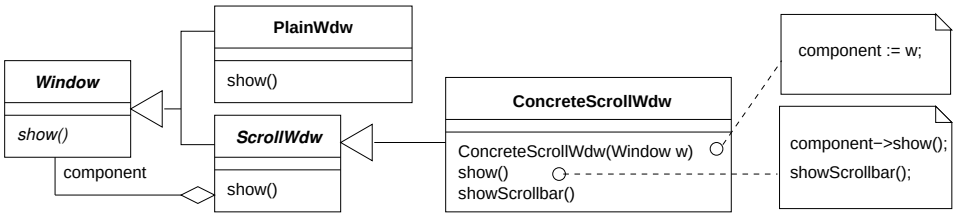

Figura 2: Domanda 7.

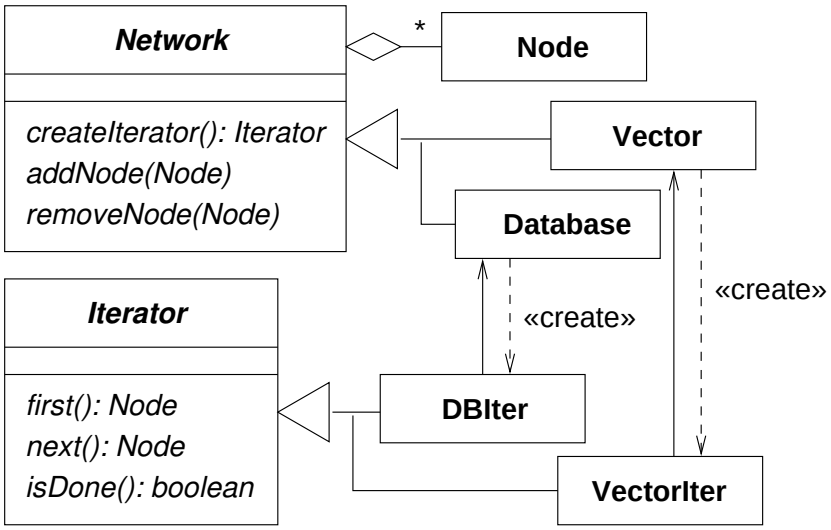

Figura 3: Domanda 8.

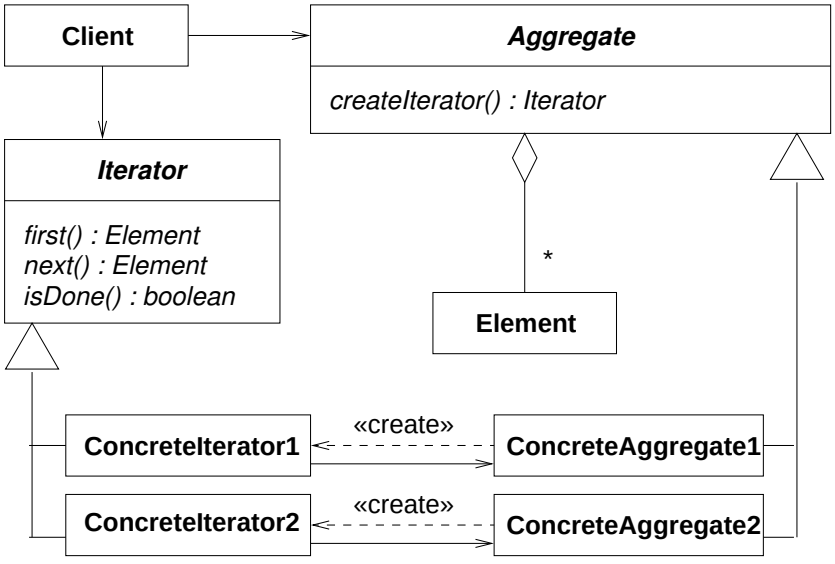

Figura 4: Domanda 10.